Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Generating evidential BEV maps in continuous driving space

Yunshuang Yuan [a], Hao Cheng [b,*], Michael Ying Yang [b], Monika Sester [a]

[a] *Institute of Cartography and Geoinformatics, Leibniz University Hannover, Germany*

[b] *Scene Understanding Group, ITC Faculty, University of Twente, The Netherlands*

ARTICLE INFO

Keywords:
Evidential deep learning
Semantic segmentation
Cooperative perception
Bird's eye view

ABSTRACT

Safety is critical for autonomous driving, and one aspect of improving safety is to accurately capture the uncertainties of the perception system, especially knowing the unknown. Different from only providing deterministic or probabilistic results, *e.g.,* probabilistic object detection, that only provide partial information for the perception scenario, we propose a complete probabilistic model named GevBEV. It interprets the 2D driving space as a probabilistic Bird's Eye View (BEV) map with point-based spatial Gaussian distributions, from which one can draw evidence as the parameters for the categorical Dirichlet distribution of any new sample point in the continuous driving space. The experimental results show that GevBEV not only provides more reliable uncertainty quantification but also outperforms the previous works on the benchmarks OPV2V and V2V4Real of BEV map interpretation for cooperative perception in simulated and real-world driving scenarios, respectively. A critical factor in cooperative perception is the data transmission size through the communication channels. GevBEV helps reduce communication overhead by selecting only the most important information to share from the learned uncertainty, reducing the average information communicated by 87% with only a slight performance drop. Our code is published at https://github.com/YuanYunshuang/GevBEV.

1. Introduction

In recent decades, a plethora of algorithms, *e.g.,* Lin et al. (2021), Feng et al. (2022), Zhang et al. (2023), Fang et al. (2022) and Zang et al. (2017), have been developed for the perception systems of autonomous vehicles (AV) and many photogrammetry and remote sensing tasks. Thanks to the open-sourced datasets, *e.g.,* Geiger et al. (2012), Caesar et al. (2020), Sun et al. (2020), and the corresponding benchmarks for different standardized perception tasks for interpreting the collected data, *e.g.,* object detection and semantic segmentation, we are able to evaluate the performance of these algorithms by comparing their predicted results with human-annotated ground truth. However, is an algorithm's better performance on these benchmarks the only goal we should chase? Obviously not; this goal is not enough for the system to be deployed reliably in the real world.

An AV system must accurately evaluate the trustworthiness of its interpretation of the driving environment, not just focus on accuracy compared to ground truth. This is because real-world driving is complex, with inevitable noise, limitations of sensors and algorithms, and occlusions that make it impossible for an AV system to be perfect. For example, from an AV's perception range, occlusions are inevitable. In this case, predicting an occluded area as a drivable road surface is likely to improve the overall accuracy because the road surface is more frequently seen in the data collected in the past, whereas it does not necessarily reflect the real-world situation. The unaccountable guess over an unobserved target based on the prior distribution drawn from historical data ignores observation uncertainties and may even lead to serious accidents. Consequently, the perception algorithms may evolve to reach higher scores on these benchmarks at the sacrifice of being overconfident and predicting dangerous false positives. Therefore, safer and more trustworthy interpretations of the driving scenarios are merited.

Achieving safe driving while considering uncertainties in the perception system is a challenging task. In the real world, uncertainties come from different sources (Gawlikowski et al., 2021), such as sensor noise, imperfections, and perception model failures. When measurements are insufficient due to limited field-of-view (FoV), occlusions, or low sensor accuracy and resolution, high uncertainty is common. One common circumstance is that an ego AV with limited FoV cannot reliably perceive the driving area, leading to a critical situation. In such a case, the ego AV should slow down and wait for the clearance of this uncertain area. An alternative is to exploit cooperative perception, *i.e.,* the information seen by other road users with a different part of the view.

This paper proposes a cooperative perception method with the consideration of uncertainty to address the limited FoV and safety problem. The concept of cooperative perception (co-perception) is to share and fuse the information from the so-called Collective Perception Messages (CPMs) among AVs to enable seeing the areas beyond the ego

[*] Corresponding author.
E-mail address: h.cheng-2@utwente.nl (H. Cheng).

https://doi.org/10.1016/j.isprsjprs.2023.08.013
Received 6 February 2023; Received in revised form 30 July 2023; Accepted 28 August 2023
Available online 8 September 2023

vehicles' own views via Vehicle-to-Vehicle (V2V) communication. The AVs with communication abilities are called Connected Autonomous Vehicles (CAVs) (see an example in Fig. 5). One bottleneck of the co-perception technique to be realized in the real world is the communication overhead and time delay for real-time communication. Hence, sharing a large amount of data, *e.g.,* raw sensory data, among the CAVs is not an optimal solution. Although recent works (Yuan et al., 2022; Xu et al., 2022a; Cui et al., 2022; Xu et al., 2022d) based on sharing deep features learned by neural networks have proven that co-perception can significantly improve the performance of the perception system, the CAVs should only share the most important information needed by the ego CAV to reduce the communication workload. The reason is that the congested network drops messages and then leads to a significant performance drop in the co-perception system.

This paper proposes to interpret the driving space for co-perception scenarios by BEV maps with uncertainty quantification. Each 2D point on the driving surface in a BEV map is classified into one of the predefined categories. Compared to object detection and semantic segmentation, this BEV interpretation gives a more comprehensive and denser overview of the driving surroundings to assist the AVs in safer driving plans. In this paper, we use Evidential Deep Learning (EDL) (Sensoy et al., 2018) to quantify the uncertainty of the classification. Essentially, our proposed BEV map also provides quantified uncertainty values for the point-wise classification results.

More specifically, we interpret the driving scenarios with learnable point-based spatial Gaussian distributions in a continuous driving space instead of discrete grids so that any new sampled points in the observed area can draw densities from these Gaussian distributions. Each Gaussian describes the likelihood of the neighboring points belonging to the same class as the distribution center point. The classification distribution of these newly sampled points is assumed to be a Dirichlet distribution to describe the probability and the uncertainty that the point belongs to one specific class. The sampled densities from the Gaussian distributions are then regarded as evidence of the Dirichlet distribution. Hence, the parameters of these Gaussian distributions can be jointly learned by controlling the Dirichlet distributions of the new samples. Our method provides a reliable uncertainty that is back-traceable and explainable for each of its prediction results. For simplicity, we name our Gaussian Evidential BEV approach **GevBEV**.

Furthermore, the learned evidential BEV maps provide a holistic interpretation of the driving environment. Namely, apart from the confident detection of the drivable surface and other vehicles from the point-wise classification results, the self-driving system knows what it is not sure about (detections with high uncertainty) or does not know at all (unobserved areas with no measurement). In the co-perception network, the evidential BEV maps serve as a critical criterion to identify the exact areas where extra information is needed from other CAVs via more efficient communication for the co-perception. Based on this criterion, the most important information shared among the CAVs is distilled by intersecting the evidential maps associated with each CAV's detection in the local frame of the ego CAV. In this way, the redundant data in CPMs is avoided to prevent the communication network from saturation and package dropping.

In summary, the *key contributions* of our proposed evidential GevBEV are:

- We propose a Gaussian-based framework to learn holistic BEV maps in a continuous space of any resolution, which is in contrast to previous works that are limited to the map resolution provided by the training data.
- The evidential deep learning with Dirichlet evidence is utilized to quantify the classification uncertainty and generates better-calibrated uncertainties than conventional deterministic models.
- Our model GevBEV achieves a new state-of-the-art performance on the co-perception benchmarks OPV2V (Xu et al., 2022d) with simulated driving scenes as well as V2V4Real (Xu et al., 2023a) in real-world driving, outperforming the runner up model CoBEVT (Xu et al., 2022b) with a big margin.
- To our best knowledge, we are the first to apply evidential BEV maps for a co-perception task. Classification uncertainties serve as a critical criterion to effectively select and share CPM among CAVs and significantly reduce communication overhead.

2. Related work

In this section, we discuss the related work in three aspects: interpretation of driving spaces, the state-of-the-art of co-perception, and uncertainty estimation of BEV maps.

2.1. Interpretation of driving spaces

Object detection (Jiao et al., 2019; Li et al., 2022b) is a typical way to interpret the driving space of an AV. A detected object is commonly characterized by a 2D/3D bounding box from camera/LiDAR data. However, this interpretation of the driving scenario may not be complete because the space with no detection or occlusion is not interpreted. This space can be some drivable areas, non-drivable areas, or be occupied by objects that are not detected or observed. Consequently, the AV may not be able to make reliable driving decisions depending on the output of object detection.

Semantic segmentation is another common method for the interpretation of the driving space. It classifies each measurement point – pixels in images or points of LiDAR reflections – into a specific semantic class. To further differentiate the points from the same semantic class but belonging to different object instances, it is extended to panoptic segmentation (Kirillov et al., 2018); besides the semantic label, the measurement point is also assigned with an instance identity. Although semantic and panoptic segmentations are holistic and dense in the ego AV's perspective – range view, they are partial and sparse in the BEV, from which the AV usually makes driving plans (Qiu et al., 2022).

Typically, the interpretation of the driving space is further extended to BEV map segmentation to mitigate the aforementioned limitations. The driving environment is represented as a BEV 2D image (Zhou and Krähenbühl, 2022; Xu et al., 2022b) and each pixel in the BEV map is marked with a semantic label, which gives a holistic overview of the driving surface for vehicle mapping and planning (Loukkal et al., 2021). In previous image-based works, BEV interpretations are also carried out as occupancy grid mapping (Lu et al., 2019), cross-view semantic segmentation (Pan et al., 2020a), or map-view semantic segmentation (Zhou and Krähenbühl, 2022). They transform image features from the image coordinates to an orthographic coordinate of the BEV map via either explicit geometric or implicit learned transformations (Zhou and Krähenbühl, 2022; Xu et al., 2022b; Li et al., 2022a). Compared to images, point cloud data with 3D information are more straightforward to generate such BEV maps by compressing information in the orthogonal direction in approaches such as PIXOR (Yang et al., 2018), Pointpillars (Lang et al., 2019), and Voxelnet (Zhou and Tuzel, 2018). We also resort to BEV maps for a holistic view of the 2D driving space. Nevertheless, because of the sparsity of distant measurements and occlusions, generating dense BEV maps from images or point clouds is unreliable without considering observability. For example, the occupied area of some occluded vehicles might be classified as a drivable area just because the categorical distribution learned from the historical data implies that the invisible points in the BEV map are more likely to be a drivable area than a vehicle. Therefore, in order to avoid unaccountable predictions on unobserved areas, we propose to only draw results from observed areas based on the geometric location of the measured points.

2.2. State-of-the-art of co-perception

With the development of Vehicle-to-Vehicle communication and the availability of simulation tools to generate high-fidelity collaborative detection data (Dosovitskiy et al., 2017; Xu et al., 2021, 2022d), co-perception extends the perception system from a single ego vehicle's perspective to including the perceptions from neighboring vehicles. Modern deep learning architectures, such as graph neural networks and Transformer (Vaswani et al., 2017), are utilized to fuse the perception information, mainly deep features of the backbone detection networks, from the CAVs. For example, V2Vnet (Wang et al., 2020b) and DiscoNet (Li et al., 2021) use graph models to aggregate the detection information from nearby vehicles. AttFuse (Xu et al., 2022d), V2XViT (Xu et al., 2022c), and CoBEVT (Xu et al., 2022b) propose to use the Transformer network with the self-attention mechanism to facilitate the collaboration and information fusion in a BEV setting among CAVs. FCooper (Chen, 2019) keeps the sparsity of feature maps learned from point clouds to reduce communication overhead of co-perception, and uses Maxout to fuse these features. However, none of these methods have explored uncertainty estimation to filter out non-informative data shared among the CAVs and further increase the communication efficiency of the co-perception system.

2.3. Uncertainty estimation of BEV maps

Uncertainties are not avoidable in DNNs, making it necessary to estimate them, especially for safety-critical applications. The uncertainty of a DNN's output is called predictive uncertainty (Gawlikowski et al., 2021). This uncertainty is mostly qualified either by modeling the epistemic uncertainty that captures the systematic uncertainty in the model or the aleatoric uncertainty that captures the random noise of observations (Kendall and Gal, 2017). There are also other approaches, such as Prior Network (Malinin and Gales, 2018) that quantifies the predictive uncertainty by modeling the distributional uncertainty caused by the distribution mismatch between the training data and the new inference data.

To estimate the epistemic uncertainty, Bayesian Neural Networks (BNNs) (Mackay, 1991; Neal, 1995) provide a natural interpretation of the uncertainty by directly inferring distributions over the network parameters. Notwithstanding, they are hard to use for DNNs because calculating the posterior over millions of parameters is intractable. Therefore, approximation methods such as Monte-Carlo (MC) Dropout (Gal, 2016) and Deep Ensemble (Lakshminarayanan et al., 2017) have been developed. MC Dropout shows that training a dropout-based neural network is equivalent to optimizing the posterior distribution of the network output. However, several forward runs have to be conducted with the dropout enabled to infer the uncertainty, which is inefficient and time-consuming. Therefore, it is not considered in this work. Deep Ensemble trains several models to approximate the distribution of the network parameters. It also needs several forward runs over each trained model and thus is also not adopted in this work for the same reason.

To capture the aleatoric uncertainty, Direct Modeling is widely used (Feng et al., 2019; Meyer et al., 2019; Miller et al., 2019; Pan et al., 2020b; Feng et al., 2020). Compared to MC Dropout and Deep Ensemble, Direct Modeling assumes a probability distribution over the network outputs and directly predicts the parameters for the assumed distribution. Therefore, uncertainty is obtained over a single forward run and is more efficient. For classification problems, the conventional deterministic DNNs apply the Softmax function over the output logits to model the categorical distribution as multi-nominal distribution. However, the Softmax outputs are often overconfident and poorly calibrated (Sensoy et al., 2018; Vasudevan et al., 2019).

Instead, converting the output logits into positive numbers via, *e.g.,* ReLU activation to parameterize a Dirichlet distribution quantifies class probabilities and uncertainties better. For example, the Prior Network (Malinin and Gales, 2018) captures the predictive uncertainty by explicitly modeling the distributional uncertainty and minimizing the expected Kullback–Leibler (KL) divergence between the predictions over certain (in-distribution) data and a sharp Dirichlet and between the predictions over uncertain (out-of-distribution) data and a flat Dirichlet. However, additional out-of-distribution samples are needed to train such a network to differentiate in- and out-of-distribution samples. In complex visual problems like object detection and semantic segmentation, obtaining enough samples to cover the infinite out-of-distribution space is prohibitive.

Differently, the Evidential Neural Network (Sensoy et al., 2018) treats the network output as beliefs following the Evidence and Dempster–Shafer theory (Dempster, 2008) and then derives the parameters for the Dirichlet distribution to model the epistemic uncertainty. Compared to BNNs, this method quantifies the uncertainty of a classification by the collection of evidence leading to the prediction result, meaning that the epistemic uncertainty of the classification can be easily quantified by the amount of evidence. Instead of minimizing the discrepancy of the predictive distributions with pre-defined ground truth distributions, Evidential Neural Network formulates the loss as the expected value of the basic loss, *e.g.,* cross-entropy, for the Dirichlet distribution. Therefore, no additional data or ground truth distributions are needed. Hence, we propose to apply this method for modeling the categorical distributions of the points in a 2D driving space.

Moreover, instead of modeling the uncertainty with Dirichlet distribution directly, we introduce a spatial Gaussian distribution for the measurement points and draw Dirichlet distributions for a BEV map based on the predicted Gaussian parameters. This configuration mimics the conditional random field algorithm (Lafferty et al., 2001), which can be used to smooth the segmentation results by considering the neighboring results. It should be noted that in this paper, we are more focused on epistemic uncertainty (a lack of knowledge in the neural network-based model) to help us understand the output of the perception system in CAVs, and to use this uncertainty quantification in the co-perception step for distilling the most important information shared among the CAVs.

3. Method

3.1. Problem formulation

We formulate the task of generating the Gaussian evidence BEV (GevBEV) map with object detection and semantic segmentation under the setting of co-perception between the ego CAV and cooperative CAVs. We follow the OPV2V benchmark (Xu et al., 2022d) setting for co-perception. Given the ego vehicle, there are $N_{\text{coop}} < 6$ cooperative CAVs in the communication range of the ego vehicle and some other vehicular participants. All CAVs can send CPMs to each other in a Request-Respond manner – the ego CAV first sends a CPM request that specifies the information over the area it needs, then a cooperative CAV that receives this request will respond with the corresponding message only if the request information is available.

In this paper, we choose to use point cloud data to demonstrate our proposed interpretation of driving spaces. It should be noted that this interpretation can be applied to any modality or multi-modalities as far as the measurement points can be projected to the BEV map, not only for autonomous driving but also other mapping tasks with multiple sources of measurements. We leave this for our future work. The input feature vector of each point in the point cloud is denoted as $f^{\text{in}} = [x, y, z, d, \cos\theta, \sin\theta, \mathrm{i}]$, where x, y, z are the local coordinates in the ego LiDAR frame, d is the distance of the point to the LiDAR origin, θ is the angle of the point relative to the x-axis of the LiDAR frame, and i is the intensity of the LiDAR reflection. This input feature vector is leveraged to train a U-Net-based (Ronneberger et al., 2015) end-to-end multi-task network. Namely, the main outputs of the proposed GevBEV are object detection results, the BEV maps for both the driving

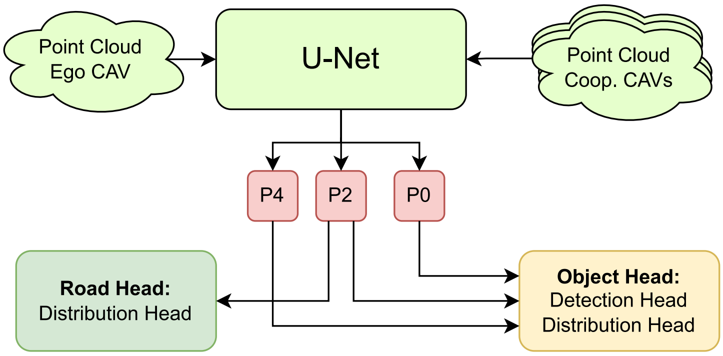

Fig. 1. Overview of the GevBEV map framework. It takes as input the point cloud data. The road head learns the distribution for the road surface (green), and the objects head (orange) detects the objects and learns the corresponding distribution of the bounding boxes. P_s: intermediate learned features of voxels downsampled with strides $s = 0, 2, 4$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

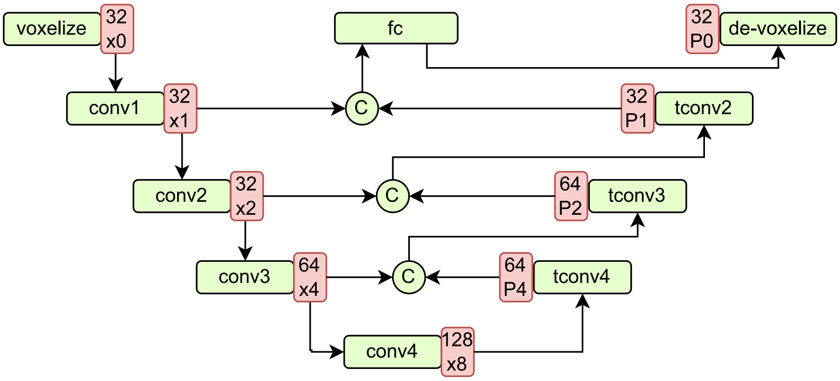

Fig. 2. The U-Net-based backbone. The light green boxes indicate the network components, the orange boxes show the components' output channels and voxel strides. The green ellipses are the concatenation operation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

surface (*i.e.,* roads) and dynamic road objects (*i.e.,* vehicles), and the uncertainty of the predicted BEV maps is quantified by a categorical Dirichlet distribution.

In the following subsections, we first introduce the detailed framework for generating GevBEV maps by sharing all the perception information among CAVs without any filtering strategy to distill the most important information. Then, we present the method for an uncertainty-based CPM selection by manipulating our generated evidence BEV maps to improve the efficiency of the communication among CAVs.

3.2. Framework

The overview of the GevBEV map framework is shown in Fig. 1. It takes as input the point cloud data from ego and cooperative vehicles and trains multiple heads for multi-task learning. Concretely, a U-Net-based backbone network first learns and aggregates deep features of different resolutions from the input point clouds. The aggregated features for voxels of different sizes are notated as P_s, where s is the downsampling ratio relative to the predefined input voxel size. P_0 are the learned features for each point in the point cloud. Those features of different voxel sizes are separately input to a road head and an objects head. The road head, specified as *Distribution Head*, generates the GevBEV map layer for the static road surface; The objects head, specified as *Detection Head* and *Distribution Head*, detects dynamic objects and also generates the GevBEV map for the object layer. We separate the GevBEV map into two layers because objects tend to have smaller sizes compared to the road surface. Otherwise, small objects could be smoothed out by the points of surfaces because they are dominating in quantity. We explain each module of the framework, *i.e., U-Net-based backbone*, *Detection Head*, and *Distribution Head*, in the following in detail.

U-Net. Given its high performance on point-wise feature extraction and representation, U-Net (Ronneberger et al., 2015) is utilized as the backbone of our model. Fig. 2 depicts the general structure of the U-Net with our customized encoding. First, the input features are voxelized with a Multi-Layer Perceptron (mlp) as described by Eq. (1),

$$\bar{f}_v = \frac{\sum_{i\in\mathbf{v}} f_i^{\text{in}}}{|\mathbf{v}|}, \quad \tilde{f}_{vi} = \text{mlp}([f_{vi}, (f_{vi} - \bar{f}_v)]), \tag{1}$$

where $\mathbf{v}$ is the set of points belonging to a voxel, f_i^{in} denotes the input feature of point i, and $[\cdot, \cdot]$ represents the concatenation operation. Then, the voxel features $\tilde{f}_{\mathbf{v}}$ are calculated by averaging the encoded point features $\tilde{f}_{vi}$ that belong to the voxel, as denoted in Eq. (2).

$$\tilde{f}_{\mathbf{v}} = \frac{\sum_{i\in\mathbf{v}} \tilde{f}_{vi}}{|\mathbf{v}|}. \tag{2}$$

Afterwards, the encoded voxels are fed to the four convolutional blocks. Namely, conv1 contains only one layer to digest the input. conv2 to conv4 consist of three convolutional layers, in which a previous layer down-samples the sparse tensors. Each sparse convolutional layer is followed by batch normalization and Leaky ReLU activation. In the upsampling layers, the transposed convolutional layers have a similar structure as the counterparts in the downsampling layers. The features from the shortcuts of the convolutional layers are all concatenated with the features from the transposed convolutional layers. In the end, we concatenate the voxel features with the encoded point features $\tilde{f}_{vi}$ to de-voxelize the stride-one voxels and obtain features P_0 for each point. It is worth mentioning that all the convolutional layers in this network are implemented with Minkowski Engine (Choy et al., 2019) over sparse voxels to decrease the computational load.

Detection head. We use this head to demonstrate the alignment of the predicted bounding boxes of object detection with the GevBEV maps. As

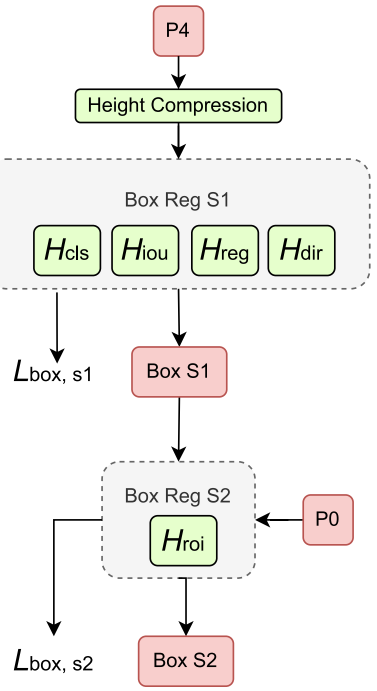

Fig. 3. Detection head.

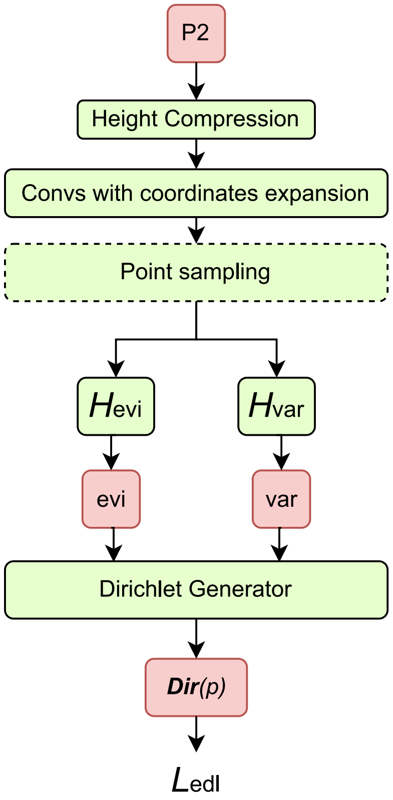

Fig. 4. Distribution head.

shown in Fig. 3, in this head, we use voxels of stride-four P_4 to generate pre-defined reference bounding boxes, the so-called anchors, and to do further regression based on them. First, we use strided convolutional layers over the vertical dimension to compress the height of the 3D sparse tensor in order to obtain a BEV feature map. Then, this map is used for the bounding box classification and regression. We follow Yuan et al. (2022) that uses a two-stage strategy for co-perception: Stage-one (BoxReg S1) generates proposal boxes based on the local information of the ego CAVs, and Stage-two (BoxReg S2) refines the boxes by fusing the information from the neighboring CAVs.

We use the following Eqs. (3)–(5) to encode bounding boxes:

$$\mathcal{B}_{loc} = [\frac{x_g - x_a}{d_{xy}}, \frac{y_g - y_a}{d_{xy}}, \frac{z_g - z_a}{h_a}], \tag{3}$$

$$\mathcal{B}_{dim} = [\log(\frac{l_g}{l_a}), \log(\frac{w_g}{w_a}), \log(\frac{h_g}{h_a})], \tag{4}$$

$$\begin{aligned} \mathcal{B}_{dir} = [&\cos\theta_g - \cos\theta_a, \sin\theta_g - \sin\theta_a, \\ &\cos\theta_g - \cos(\theta_a + \pi), \sin\theta_g - \sin(\theta_a + \pi)]. \end{aligned} \tag{5}$$

$\mathcal{B}_{loc}$ and $\mathcal{B}_{dim}$ are the commonly used encodings for the location and dimension of the bounding boxes. The subscripts g and a represent ground truth and anchor, respectively. x, y, z are the coordinates of the bounding box centers, and d_{xy} is the diagonal length of the bounding box in the XY-plane. l, w, h indicate the bounding box length, width, and height. We use $\sin$ and $\cos$ encodings, denoted by Eq. (5), for the bounding box direction θ to avoid the direction flipping problem. The first two elements of $\mathcal{B}_{dir}$ are the encodings of the original direction angle, while the last two elements are for the reversed direction. The additional encodings for the reversed direction reduce the distance of sin-cos encodings between the regression source–anchors and the target–ground truth boxes whenever they have opposite directions, therefore, facilitating the direction regression.

The overall loss for all the heads in BoxReg S1 is notated as $\mathcal{L}_{box,s1}$, as shown in Fig. 3. To be more specific, BoxReg S1 follows CIASSD (Zheng et al., 2021) to use four heads $\mathcal{H}_{cls}, \mathcal{H}_{iou}, \mathcal{H}_{reg}, \mathcal{H}_{dir}$ for generating bounding boxes. $\mathcal{H}_{cls}$ uses binary cross-entropy, other heads in this stage use smooth-L1 loss. $\mathcal{H}_{iou}$ regresses the Intersection over Unions (IoU) between the detected and ground truth bounding boxes. It is used to rectify the classification score from $\mathcal{H}_{cls}$ so that better-bounding boxes can be kept during Non-Maximum Suppression (NMS). The direction head $\mathcal{H}_{dir}$ regresses the encoded angle offsets between the ground truth bounding boxes and the corresponding anchors. During the box decoding phase, the predicted angle offsets from $\mathcal{H}_{dir}$ with smaller values are selected as the correct direction of the bounding boxes.

In BoxReg S2, the proposal bounding boxes detected by different CAVs are first shared with the ego CAV and fused by NMS, and then they are used as reference anchors for further refinement. Concretely, following FPV-RCNN (Yuan et al., 2022), the CAVs only share the keypoints belonging to the proposals and then use these keypoints to refine the fused bounding boxes. Moreover, in this paper, we use Minkowski Engine (Choy et al., 2019) to simplify the structure of Region-of-Interest (RoI) head $\mathcal{H}_{roi}$. Namely, the fused boxes are first transformed to a canonical local box coordinate system, where the keypoints coordinates are noted as kpt_i^{in}. We then voxelize the keypoints to a 6 × 6 × 6 grid with a similar voxelization operation as used in the U-Net, where f_i^{in} becomes the canonical coordinates kpt_i^{in} of keypoints $[x, y, z]$. The first element f_{vi} of the concatenation in Eq. (1) becomes the learned feature of keypoints and the second element becomes the learned positional encoding of the keypoint coordinates. To make it straightforward for understanding, Eq. (1) is reformulated into Eq. (6).

$$\begin{aligned} \text{k}\bar{\text{p}}\text{t}_v &= \frac{\sum_{i \in \mathbf{v}} \text{kpt}_i^{in}}{|\mathbf{v}|}, \\ \tilde{f}_{vi} &= \text{mlp}([f_{vi}, \text{mlp}(\text{kpt}_{vi} - \text{k}\bar{\text{p}}\text{t}_v)]). \end{aligned} \tag{6}$$

After summarizing the features for each voxel by Eq. (2), all voxels in the grids are then aggregated by the weighted average of the voxel features $\tilde{f}_{\mathbf{v}}$. The weights are learned from $\tilde{f}_{\mathbf{v}}$ by mlp. The aggregated features of each box grid are then fed to one IoU head and one box regression head to generate the refinement parameters for the fused proposal boxes. The decoded bounding boxes are notated as Box S2, as shown in Fig. 3. Smooth-L1 loss is used for both heads. The summation of the losses in this stage is notated as $L_{box,s2}$.

Distribution head. As illustrated in Fig. 1, we design a distribution head to generate point-based distributions for both road surface and objects. To balance the trade-off between computational load and the point resolution, we use voxels of stride-two P_2 to generate distributions for this purpose. The overall structure of the distribution head is described in Fig. 4. The input 3D voxels are first compressed along the vertical direction to obtain the 2D point-wise deep features. This height compression is composed of two sparse convolution layers with the same kernel size and stride size so that all voxels in the vertical direction can merge into one. In this module, the weights are shared for both the road and object distribution heads.

Furthermore, we dilate the voxel coordinates during the sparse convolutions to close the gaps between discrete measurement points. We call this as coordinate expansion. Even though we propose to only predict the distributions on observable areas, all sensors measure the continuous space in a discrete way, leading to unavoidable gaps at a large measure distance. As described above, on the one hand, the basic convolutional layers used in the U-Net maintain the sparsity of the voxels to reduce the computational load. On the other hand, they cannot infer information from the gaps between two neighboring laser measurements caused by the range view of the LiDAR at distant observable areas. However, as long as the measurement density is enough, the model should also be able to infer the information between the two discrete measurements in a controlled manner. Therefore, we use several coordinate-expandable sparse convolutions to carefully close the gaps by controlling the expansion range with predefined kernel sizes. The detailed setting is given in Section 4.5.

To further accelerate the training process, we use the point sampling module, as illustrated with the dashed line box in Fig. 4, to downsample the total amount of points. We call all these selected points center points for the simplicity of explanation. This module is optional, and the later empirical results show that it does not have an obvious negative influence on the training result.

We assume each center point c_i has a Dirichlet distribution to model the point classification distribution and a spatial isotropic Gaussian distribution to model the neighborhood consistency of the point. The parameters for these two distributions are then regressed by the head $\mathcal{H}_{\text{cls}}$ and $\mathcal{H}_{\text{var}}$, respectively. Both heads are composed of two fully connected layers activated by ReLU to constrain the parameters for both distributions to be positive. Their outputs are noted as

$$\mathbf{o}_{\text{cls}} = [o^{\text{fg}}_{\text{cls}}, o^{\text{bg}}_{\text{cls}}], \tag{7}$$

$$\mathbf{o}_{\text{var}} = [o^{\text{fg}}_{\sigma x}, o^{\text{fg}}_{\sigma y}, o^{\text{bg}}_{\sigma x}, o^{\text{bg}}_{\sigma y}], \tag{8}$$

where fg indicates foreground and bg background. $\mathbf{o}_{\text{cls}}$ is regarded as the evidence of the point to be foreground or background. $\mathbf{o}_{\text{var}}$ is the regressed variances of the point in x- and y-axis. To ensure that each point is contributing, we add a small initial variance to the predictions. Hence, the resulting variances $\sigma^2_{x,y} = \mathbf{o}_{\text{var}} + \sigma^2_0$. For any new given target point $\mathbf{x}_j$ in the neighborhood of the center point $\mathbf{c}_i$ in the BEV space, we can then draw the probability density $\phi(\mathbf{x}_j)_i$ of this new point belonging to a specific class by Eq. (11),

$$\Sigma_i = \begin{bmatrix} \sigma^2_x, 0 \\ 0, \sigma^2_y \end{bmatrix}, \tag{9}$$

$$m_{ji} = (\mathbf{x}_j - \mathbf{c}_i)^T \Sigma_i^{-1} (\mathbf{x}_j - \mathbf{c}_i), \tag{10}$$

$$\phi(\mathbf{x}_j)_i = \frac{\exp(-0.5 \cdot m)}{\sqrt{2\pi^d |\Sigma_i|}}, \tag{11}$$

where Σ_i is the covariance of the center point $\mathbf{c}_i$ for foreground or background distribution, m is the squared Mahalanobis distance of point $\mathbf{x}_j$ to the center point $\mathbf{c}_i$, and d is the dimension of the distribution. In our case, $d = 2$.

To obtain the overall Dirichlet evidence $e(\mathbf{x}_j)$ for point $\mathbf{x}_j$, we summarize the normalized and weighted probability mass drawn from all the neighboring center points $\text{nbr}(j)$ that is in the maximum distribution range v as

$$\begin{aligned} e_k(\mathbf{x}_j) &= \sum_{i \in \text{nbr}(j)} \frac{\phi(\mathbf{x}_j)_i}{\phi(\mathbf{c}_i)} \cdot o^k_{cls}, \\ &= -\frac{1}{2} \sum_{i \in \text{nbr}(j)} m_{ji} \cdot o^k_{\text{cls}}, \end{aligned} \tag{12}$$

where $\phi(\mathbf{c}_i)$ is the probability density at the center point, and $k \in \{\text{fg}, \text{bg}\}$. Hereafter, Eq. (12) is derived as following

$$\begin{aligned} \log(\phi(\mathbf{x}_j)_i) &= -\frac{1}{2} m_{ji} - \frac{1}{2} \log(2\pi^d |\Sigma|), \\ &= -\frac{1}{2}(d \log(2\pi) - \log |\Sigma|) - \frac{1}{2} m_{ji}, \\ \log(\phi(\mathbf{c}_i)) &= -\frac{1}{2} m_i - \frac{1}{2} \log(2\pi^d |\Sigma|), \\ &= -\frac{1}{2}(d \log(2\pi) - \log |\Sigma|), \\ \frac{\phi(\mathbf{x}_j)_i}{\phi(\mathbf{c}_i)} &= \exp(\log \frac{\phi(\mathbf{x}_j)_i}{\phi(\mathbf{c}_i)}), \\ &= \exp(\log \phi(\mathbf{x}_j)_i - \log \phi(\mathbf{c}_i)), \\ &= -\frac{1}{2} m_{ji}, \end{aligned} \tag{13}$$

where the squared Mahalanobis distance of the center point $\mathbf{c}_i$ to itself is $m_i = 0$. Following Sensoy et al. (2018), the expected probability $p_{j,k}$ – point $\mathbf{x}_j$ belonging to class k – and the uncertainty u_j of this classification result are

$$\hat{p}_{j,k} = \frac{\alpha_{j,k}}{S_j} = \frac{e_k(\mathbf{x}_j) + 1}{\sum_{k \in \{\text{fg},\text{bg}\}} (e_k(\mathbf{x}_j) + 1)}, \tag{14}$$

$$u_j = \frac{K}{S_j}, \tag{15}$$

where $\alpha_{j,k}$ is the concentration parameter of class k for $k = 1, \ldots, K$, and S_j the strength of the Dirichlet distribution of point $\mathbf{x}_j$. For the loss $\mathcal{L}_{edl}$ of the distribution head, we use the recommended loss function in Sensoy et al. (2018), which is formulated as the expectation of the sum of the squared loss and a Kullback–Leibler (KL) divergence regularization that prevents the network from generating excessively high evidences. The final expression of this loss is described by

$$\begin{aligned} \mathcal{L}_{edl} = &\sum_{j=1}^{N} \sum_{k \in \{\text{fg},\text{bg}\}} [(y_{j,k} - \hat{p}_{j,k})^2 + \frac{\hat{p}_{j,k}(1 - \hat{p}_{j,k})}{S_j + 1}] \\ &+ \lambda_t \sum_{j=1}^{N} \text{KL}[\text{Dir}(\mathbf{p}_j | \tilde{\alpha}_j) \parallel \text{Dir}(\mathbf{p}_j | \mathbf{1})], \end{aligned} \tag{16}$$

where

$$\tilde{\alpha}_j = \alpha_j \odot (1 - \mathbf{y}_j) + \mathbf{y}_j,$$
$$\lambda_t = \min(1, A_{\text{epoch}} / A_{\text{max}}).$$

The first term in Eq. (16) is the expected sum of squared loss between the target label $y_{j,k}$ and the prediction $p_{j,k}$. N denotes the total number of samples, and $j \in \{1, \ldots, N\}$. The second term is the KL-divergence weighted by an annealing coefficient λ_t that changes with the ratio between the epoch number A_{epoch} and the maximum annealing step A_{max}. $\mathbf{y}_j = \{y_{j,k}\}_{k \in \{\text{fg},\text{bg}\}}$ is the categorical ground truth label of point $\mathbf{x}_j$. $\tilde{\alpha}_j$ is the filtered version of α_j to ensure that the KL-divergence only punishes the misleading predictions of α_j.

We propose a Gaussian-based method to train the Distribution Head in a continuous space for the evidential BEV map. To achieve this goal, the target points for the supervised learning are not limited to the original observation points of the point clouds or the center points of a specific resolution with discrete points; They can be any points in the continuous BEV plane. More specifically, the original observed target points are treated as seeds based on which we generate continuous target points by randomly shifting them from their original observation in a controlled range, *e.g.,* by a normally distributed distance $\mathcal{N}(0, 3)$

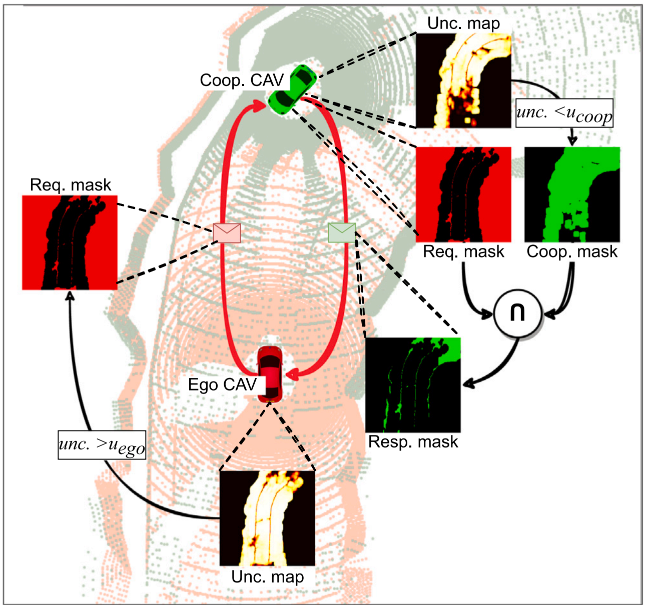

Fig. 5. An example of the information shared between CAVs. The ego CAV and its request mask with high uncertainty areas indicated in red color, and the cooperative CAV and its response mask with low uncertainty indicated in green color. In the uncertainty maps, light color indicates low uncertainty and black color indicates no measurements. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

m. In this way, a pre-defined number n_{tgt} of target points are generated from the center points to control the density, and only the generated target points of the observed areas are leveraged to train the model. To avoid memory overflow and long-computational time during training, the randomly sampled target points are further down-sampled. Via voxel down-sampling, the road head is supervised only with a limited number of target points – N_{tgt} for both foreground and background samples. However, the importance and the amount of target points for objects head are more biased between the foreground and background. Only a small amount of background target points are included for training as they are the majority but less important. In contrast, all samples that are extended to a specific range of the ground truth bounding boxes' edges are adopted for training the foreground class and better describing the details around the bounding boxes.

3.3. Co-perception

This section introduces the co-perception method for the uncertainty-based CPM selection by manipulating our generated evidence BEV maps. Instead of simply sharing all the information among CAVs that may congest the communication network, the most important information is distilled by the generated BEV maps that quantify the perception uncertainty of the CAVs. As shown in Fig. 5, the red ego CAV first generates a request binary mask by thresholding its uncertainty map generated by Eq. (12) with the threshold u_{ego} (bottom Unc. mask) and only sends the request for the perception information in the areas it has high uncertainty (left red Req. maks). Then, the green cooperative CAV responds with its own masked evidence map (green Resp. mask). To be more specific, the binary mask generated by thresholding the uncertainty map of the cooperative CAV with the threshold u_{coop} is used to intersect with the received request mask from the ego CAV, resulting in a response mask with the low-uncertainty areas from the cooperative CAV. Afterwards, this resulting response mask is used to distill the CPM communicated from the cooperative CAV to the ego CAV by only selecting the evidences over the response-masked areas. We denote this sharing strategy as CPM_{all} because it considers all areas in the pre-defined FoV of the ego CAV for information sharing. However, in the driving space, the CAVs pay more attention to the situations on the road surface. To this end, the information to be shared can be further constrained by the road surface geometry of the current scenario. In real applications, this geometry can be retrieved from some prior information, such as maps. We notice that the co-perception benchmarks also provide an HD map acquired beforehand. As a proof-of-concept study, we register the current scenario to the HD map to further rule out non-surface areas in the masked areas. We denote this sharing strategy as CPM_{road} when the extra HD map is already provided to the CAVs. For simplicity, in this paper, we fix the uncertainty threshold $u_{\mathrm{coop}} = 1.0$ for the cooperative CAVs and only vary the threshold u_{ego} for the ego CAV in our experiments to evaluate its effectiveness.

4. Experiments

In the following, we introduce the dataset, data augmentations, evaluation metrics, and detailed experiments to evaluate our proposed model.

4.1. Dataset

In this work, we conduct the experiments on two multi-agent co-perception benchmarks, OPV2V (Xu et al., 2022d), a simulated dataset generated by CARLA (Dosovitskiy et al., 2017) and OpenCDA (Xu et al., 2021, 2023b), and V2V4Real (Xu et al., 2023a), a real dataset captured with two vehicles driving in real-world scenarios.

The OPV2V dataset has 73 scenes, including six road types from nine cities. It contained 12k frames of LiDAR point clouds and the annotated 3D bounding boxes for each frame. The detection range of this dataset is set to $[-50, 50]$ m for x- and y-coordinate and $[-3, 3]$ m for z coordinate, same as the baseline work CoBEVT (Xu et al., 2022b). The V2V4Real dataset covers a driving area of 410 km. It contains about 10k annotated LiDAR frames. We set the detection range of this dataset to $[-102.4, 102.4]$ m for x-coordinate, $[-38.4, 38.4]$ m for y-coordinate and $[-5, 3]$ m for z-coordinate as most annotated vehicles are in this range. We follow the official partitioning of both datasets for the training and test. Namely, 44 training scenarios and 16 test scenarios for OPV2V, 33 training scenarios and 9 test scenarios for V2V4Real.

4.2. Data augmentation

A point cloud is one of the most common ways to represent the geometric information sensed by LiDAR sensors. It is a collection of reflected points when the LiDAR rays hit the surface of objects. Considering that the points are very sparse if a ray hits distant objects in a point cloud, we propose the following augmentations of the LiDAR data to help the perception tasks.

Free space augmentation. In a point cloud, the free space – traversed space by the ray that is not occupied by any obstacles – is often neglected. However, this information is also a result of the measurement and is critical for identifying the occupancy and visibility of the driving space. Therefore, we augment the point cloud data by sampling points from the LiDAR ray paths and call these points free space points f_{si}, where $i \in \mathbb{N}$. As exemplified in Fig. 6, a LiDAR (orange cylinder) casts a ray (red line) that hits the surface of the ground at point fs_0 and only records this reflected point into the point cloud. Over the ray path, we sample free space points f_{si} in a limited distance d_{fs} from the hit point f_{s0} with a large step s_{fs}. In order to constrain the computational overhead, we only sample points in the region of a limited height, *i.e.*, $z \leq h_{fs}$, over the ground (blue area) where it is critical for driving. Finally, these points are down-sampled again by voxel down-sampling with a given voxel size of v_{fs} to obtain evenly and sparsely distributed free space points. These augmented free

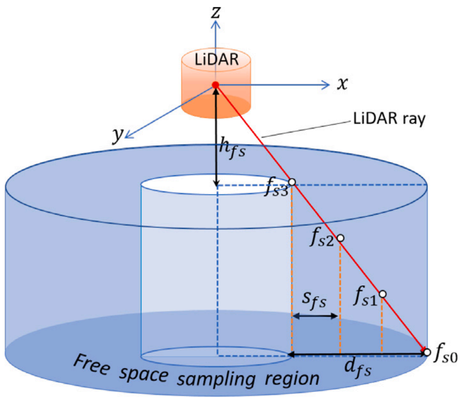

Fig. 6. Sampling free space points. The red point is the origin of the LiDAR coordinate system, z-axis indicates the vertical direction in the driving space, x-axis indicates the horizontal direction. f_{s0} is the intersection point of the LiDAR ray (red dashed line) and the ground. f_{s1}, f_{s2}, and f_{s3} are the sampled free space points belonging to the ray path. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

space points are then added to the original point cloud by setting their intensity value $\mathrm{i} = -1$ as the indicator.

Geometric augmentation. We augment the point cloud data by randomly rotating, flipping along x- and y-axis and then scaling the geometric size of the point clouds. Also, we add a small Gaussian noise $[\delta x, \delta y, \delta z]$ to all the points. This augmentation helps increase the robustness of the model against domain shift during testing.

4.3. Evaluation metrics

We use Intersection over Union (IoU) as a metric to evaluate the overall prediction performance and the calibration plot to analyze the quality of the predictive uncertainty. Although our proposed model GevBEV can generate BEV maps of any resolution, we only evaluate the predicted BEV maps of resolution of 0.4 m so that it can be compared with other methods only generating the BEV maps of this predefined prediction resolution.

IoU. We report both the result overall perception areas in the predefined prediction range (IoU_{all}) and the result over the observable areas (IoU_{obs}), which is in line with the concept that the prediction is reliable only when it is conducted over the observed areas. Since our model is designed not to conduct predictions over non-observable areas, all the points in non-observable areas are regarded as false when calculating IoU_{all}. Mathematically, a point $\mathbf{x}_j$ is observable if $\exists i \in \{i| \parallel \mathbf{x}_i - \mathbf{x}_j \parallel_2 < v\}$, meaning a point is observable if it is in the range $v = 2\,\text{m}$ of any center points. The IoUs over these observable areas are calculated by Eqs. (18)–(21).

$$\mathbf{X} = \{\mathbf{x}_j | u_j < u_{\text{thr}}\}, \tag{17}$$

$$\mathbf{X}^{\text{fg}} = \{\mathbf{x}_j | \hat{p}_j^{\text{fg}} > \hat{p}_j^{\text{bg}}, \mathbf{x}_j \in \mathbf{X}\}, \tag{18}$$

$$\mathbf{Y} = \{\mathbf{x}_j | \text{lbl}(\mathbf{x}_j) = \text{fg}\}, \tag{19}$$

$$\mathbf{Y}^{\text{fg}} = \{\mathbf{x}_j | \mathbf{x}_j \in \mathbf{Y}, \mathbf{x}_j \in \mathbf{X}\}, \tag{20}$$

$$IoU = \frac{|\mathbf{X}^{\text{fg}} \cap \mathbf{Y}^{\text{fg}}|}{|\mathbf{X}^{\text{fg}} \cup \mathbf{Y}^{\text{fg}}|}. \tag{21}$$

We only evaluate the results of points $\mathbf{X}$ that have uncertainties under the uncertainty threshold u_{thr}. $\mathbf{X}^{\text{fg}}$ is the set of positive predictions – points classified as foreground by its predictive classification probabilities. $\text{lbl}(\mathbf{x}_j)$ is the class label of point $\mathbf{x}_j$, hence $\mathbf{Y}$ is the set of true positive points. $\mathbf{Y}^{\text{fg}}$ is the set of all positive points in the ground truth. Then the IoU is calculated between the true positive and ground truth positive points.

Calibration plot. We plot classification accuracy versus uncertainty to show their desired correlation. Our model generates uncertainty for the classification of each point, which enables an overall evaluation of all classes. However, the unbalanced number of samples for different classes will lead to a biased evaluation. For example, the classification of a point has a high uncertainty due to the lack of evidence from its neighbors, while it may end up with high classification accuracy because its neighbors belong to a dominant class. To avoid this biased evaluation, each sample is weighted by the ratio of the total number of samples in that particular class. Then, we divide the uncertainty $u \in [0, 1]$ into ten bins, and each bin has an interval of 0.1. Subsequently, we calculate the weighted average classification accuracy of all the samples in that uncertainty interval. A perfect calibration plot is shown by a diagonal line indicating the highest negative correlation between classification accuracy and uncertainty, *i.e.,* high accuracy is associated with low uncertainty.

4.4. Baseline and comparative models

We evaluate the effectiveness of our proposed model GevBEV compared to both state-of-the-art co-perception camera- and lidar-based models on the co-perception benchmarks OPV2V and V2V4Real in both simulated and real driving scenarios, namely *FCooper* (Chen, 2019), *AttFuse* (Xu et al., 2022d), *V2VNet* (Wang et al., 2020b), *DiscoNet* (Li et al., 2021), *V2XViT* (Xu et al., 2022c) and *i.e., CoBEVT* (Xu et al., 2022b). Unlike our GevBEV, none of these models listed above provide uncertainty estimation for perception and distilling the essential information communicated to the ego CAV. Hence, we only compare GevBEV with those models on the perception performance. They are not further compared with GevBEV in terms of CPM size for the V2V communication in the co-perception application.

Moreover, we conducted a series of ablation studies to analyze the efficacy of the proposed modules of GevBEV.

- *BEV* is the proposed model with the point-based spatial Gaussian and the evidential loss $\mathcal{L}_{edl}$ removed, turning our model from a probabilistic model into a deterministic model. It uses cross-entropy to train the corresponding heads to classify points of the BEV maps. We treat this model as our baseline model.
- *EviBEV* only has the point-based spatial Gaussian removed. It still uses $\mathcal{L}_{edl}$ to train the distribution head.
- *GevBEV⁻* is our proposed model but trained without free space augmentation.

4.5. Implementation details

In all our experiments, we set the input voxel size to $0.2\,\text{m}$ to balance between computational overhead and performance. The free space points are sampled with the configuration $h_{fs} = -1.5\,\text{m}, d_{fs} = 1.5\,\text{m}, s_{fs} = 7.5\,\text{m}$ on OPV2V. However, we increase s_{fs} to 9 m for V2V4Real dataset because it has a longer detection range in x-direction. During the geometric augmentation, the point cloud coordinates are scaled randomly in the range of $[0.95, 1.05]$ and then added with normally distributed $\mathcal{N}(0, 0.2)\,\text{m}$ noise. For the detection head, we generate two reference anchors at each observation point on the BEV map of stride four. These two anchors have the same size, $[l, w, h] = [4.41, 1.98, 1.64]\,\text{m}$ for OPV2V and $[l, w, h] = [3.90, 1.60, 1.56]\,\text{m}$ for V2V4Real. However, they have different angles, $0°$ and $90°$, respectively. The anchors that have an IoU with the ground truth bounding boxes over 0.4 are regarded as positive, and those under 0.2 as negative; other anchors are neglected for calculating the loss during training. The negative samples are many more than the positive ones, leading to difficulties in the training process. Hence, we only randomly sample 512 negative samples for training.

We have trained the whole network from scratch for 50 epochs with weight decay of 0.01 using the Adam optimizer (Kingma and Ba, 2014)

Table 1
Comparison with the state-of-the-art models on OPV2V and V2V4real dataset. Best values are highlighted in boldface and the second best values are underlined.

Model	Modality		OPV2V		V2V4Real
	Camera	LiDAR	Road	Object	Object
AttFuse (Xu et al., 2022d)	✓		60.5	51.9	–
V2VNet (Wang et al., 2020b)	✓		60.2	53.5	–
DiscoNet (Li et al., 2021)	✓		60.7	52.9	–
CoBEVT (Xu et al., 2022b)	✓		63.0	60.4	–
Fcooper (Chen, 2019)		✓	70.3	52.1	25.9
AttFuse (Xu et al., 2022d)		✓	75.3	52.0	25.5
V2XViT (Xu et al., 2022c)		✓	75.0	50.4	$\underline{29.9}$
CoBEVT (Xu et al., 2022b)		✓	$\underline{75.9}$	$\underline{52.3}$	29.6
GevBEV (ours)		✓	**79.5**	**74.7**	**46.3**

parameterized with $\beta = [0.95, 0.999]$ and $\gamma = 0.1$. We use multi-step learning-rate scheduler with a starting learning rate of $lr = 10^{-3}$ and two milestones at epoch 20 and 45. The learning rate decreases at each milestone by a factor of 0.1.

For the sampling generators, we use $n_{\mathrm{tgt}} = 10$ for the road head and $n_{\mathrm{tgt}} = 1$ for the objects head. The resolution of voxel down-sampling of the road head is set to 0.4 m, and N_{tgt} is set to 3000. For the objects head, all the generated target points with a minimum distance of less than 4 m to any ground truth boxes' edges are kept. From the background, we sample $N_{\mathrm{tgt}} = 50 \cdot n_{\mathrm{gt}}$ points as negative samples, where n_{gt} is the number of the ground truth bounding boxes. For these sampled target points, the evidences are drawn by Eq. (12) in a maximum distribution range of $v = 2$ m. The parameters mentioned above are all set empirically, and the code and the trained model is released at https://github.com/YuanYunshuang/GevBEV for reproduction.

4.6. Results

4.6.1. Quantitative analysis

Compared to state-of-the-art models. Our proposed GevBEV model is benchmarked with the state-of-the-art models for co-perception on the simulated OPV2V dataset (Xu et al., 2022d) and the real dataset V2V4Real (Xu et al., 2023a). Table 1 lists the results for road and dynamic object segmentation. It can be seen that the models conducted on camera data is inferior to LiDAR data. This is because LiDAR data provides more accurate 3D information, which is essential for the projection to a BEV map for the perception task. GevBEV outperforms all the other models, including the models that are conducted on the same LiDAR data as GevBEV for a fair comparison. Compared to the runner up model CoBEVT on the OPV2V benchmark, our model with the distribution heads improves the IoU by 23.7% for segmenting dynamic objects and 4.7% for segmenting road surfaces. On the V2V4Real Benchmark, surprisingly, our model improves the IoU by 54.8% compared to V2XViT. These improvements indicate that the point-based spatial Gaussian effectively provides smoother information about each surface point's neighborhood, leading to more accurate results on both benchmarks. Besides, our proposed sampling method for training is more robust against the errors in ground truth. This leads to a remarkable improvement on the real dataset V2V4Real that contains inaccurate labels in real-world driving.

However, the real-world driving scenarios post more challenges for the co-perception task by inevitably introducing localization errors, as indicated by the large performance gap on object segmentation between OPV2V (74.7%) and V2V4Real (46.3%), Also, the V2V4Real dataset only provides the communication between two connected vehicles, fewer than that of the simulated OPV2V dataset. Hence, the perception performance on V2V4Real is much worse than that tested on OPV2V.

Sensitivity to localization noise. In the previous experiments of the simulated data, we assumed perfect localization information. In order to evaluate the simulation of the results to localization noise, we introduce localization errors generated by normal distributions with a standard deviation ranging from 0 to 0.5 m for position and from 0 to 1 degrees for orientation. Fig. 7 demonstrates that all models experience slight performance declines in road surface segmentation as the location and rotation errors increase. Still, our proposed GevBEV model outperforms CoBEVT, maintaining the best performance with a margin of approximately 4% across all error configurations. In contrast to the segmentation performance, objects exhibit higher sensitivity to localization errors due to their smaller size. In this setting, our model still outperforms the runner up model CoBEVT on both the OPV2V and V2V4Real datasets, as depicted in Figs. 8 and 9. This indicates that our proposed approach is more robust than CoBEVT to cope with localization noise.

Ablation study. Table 2 shows the performances of the ablative models. In general, the baseline model BEV without the point-based spatial Gaussian (G.s.) and the evidential loss $\mathcal{L}_{edl}$ is inferior to the other models. This indicates that this conventional deterministic model trained by optimizing the cross-entropy is not as good as the probabilistic models. In contrast, by modeling each point with a spatial continuous Gaussian distribution, we are able to close the gaps caused by the sparsity of point clouds and generate smoother BEV maps. EviBEV with the point-based spatial Gaussian performs better than the baseline for the surface measured by the IoUs for *all* and the *observed* areas. However, its performances for objects are slightly degraded.

Moreover, BEV and EviBEV perform worse than CoBEVT (Xu et al., 2022b) on IoU_{all} of the road head, as shown in Table 1. This is because our frameworks are based on fully sparse convolution networks, which do not operate on unobserved areas. In order to facilitate comparison with dense convolution models, certain road surfaces in our frameworks are considered inadequately observed and thus treated as false predictions when calculating IoU_{all}. Unlike GevBEV, both our BEV and EviBEV models have additional areas designated as unobserved due to the absence of Gaussian tails. Consequently, this leads to lower IoU values. However, it is worth mentioning that the object classification of the ablative models significantly outperforms the runner up model, CoBEVT. This can be attributed to two factors. Firstly, we carefully control the network in our model to expand coordinates in a specific range and only make predictions over the observable areas. The coordinate expansion module can cover most of the object areas so that these areas will be given a prediction rather than being treated as unobserved. Secondly, thanks to the benefits of dynamic sampling from continuous driving space during training, our model shows a tendency to be cautious when making positive predictions for vehicle points. This cautious approach allows us to capture edge details of the vehicles more effectively, enhancing the overall object classification performance.

From the comparison between GevBEV^{-} and GevBEV, the improved IoUs, especially for roads, indicate that the free space augmentation (Ag.) provides an explicit cue to the unoccupied space along the ray paths and improves the detection performance. Also, this module plays an important role in mitigating the problem of sparsity for point clouds and largely improves the prediction performance. Overall, GevBEV outperforms the ablative models in all the measurements, which validates the efficacy of each proposed module.

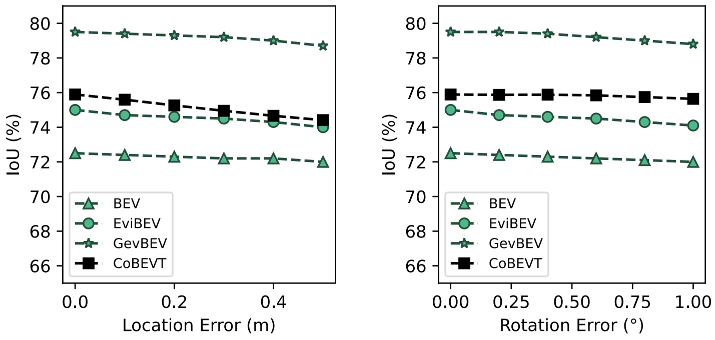

Fig. 7. OPV2V road segmentation result with localization noise.

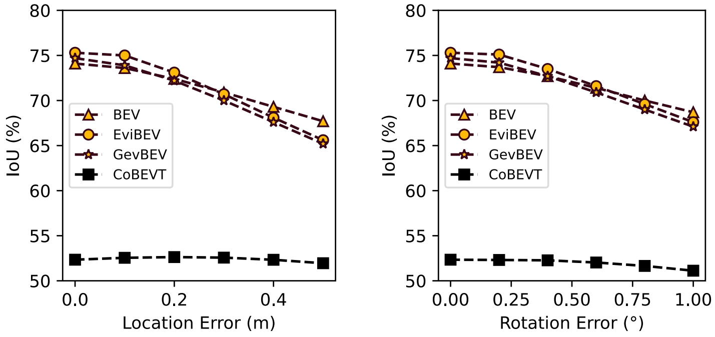

Fig. 8. OPV2V object segmentation result with localization noise.

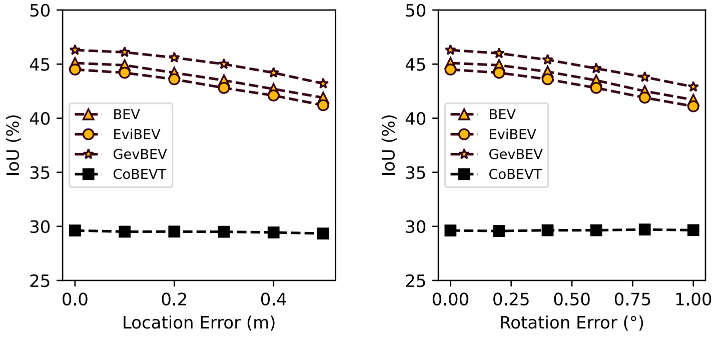

Fig. 9. V2V4Real object segmentation result with localization noise.

Table 2
Ablation study of the proposed modules. All IoU results are measured in percentage. Best values are highlighted in boldface. G.s.: point-based spatial Gaussian, $\mathcal{L}_{edl}$: evidential loss, Ag.: free space augmentation; Rd: road, Obj: object.

Model	Modules			OPV2V (all)		V2V4Real (all)	OPV2V (obs)		V2V4Real (obs)
	G.s.	$\mathcal{L}_{edl}$	Ag.	Rd	Obj	Obj	Rd	Obj	Obj
BEV			✓	72.5	74.1	45.1	76.1	75.8	46.1
EviBEV		✓	✓	75.0	**75.3**	44.5	78.3	**76.3**	45.3
GevBEV$^-$	✓	✓		59.7	73.1	46.0	62.5	73.2	46.7
GevBEV	✓	✓	✓	**79.5**	74.7	**46.3**	**83.1**	76.1	**46.9**

4.6.2. Visual analysis

Holistic BEV maps for autonomous driving. With our proposed probabilistic model, we generate the GevBEV maps and visualize the results in a complex driving scenario in Fig. 10. From left to right, the three sub-figures in the first row show the results of uncertainty, classification confidence, and the ground truth of the road surface. In the uncertainty map, a lighter color indicates lower uncertainty, whereas black areas are regarded as non-observable. Correspondingly, the confidence map gives the confidence score for both foreground (road surface) and background. The bottom sub-figure is the detailed detection results of both road surface and objects overlaid in one figure. Only the points that are classified as roads with uncertainty under the threshold of 0.7 are highlighted in the light color in the bottom layer to show the situation of the drivable area. The predicted and corresponding ground truth bounding boxes are plotted in red and green colors, respectively. Moreover, the bounding boxes are filled with the point confidences drawn from the objects head, where magenta points are associated with high confidence, while cyan the opposite.

As shown in Fig. 10, the GevBEV maps are regarded as holistic BEV maps providing a reliable information source to support AVs for decision-making. First, this information can be sourced back to the original measurement points. For example, due to occlusions and long

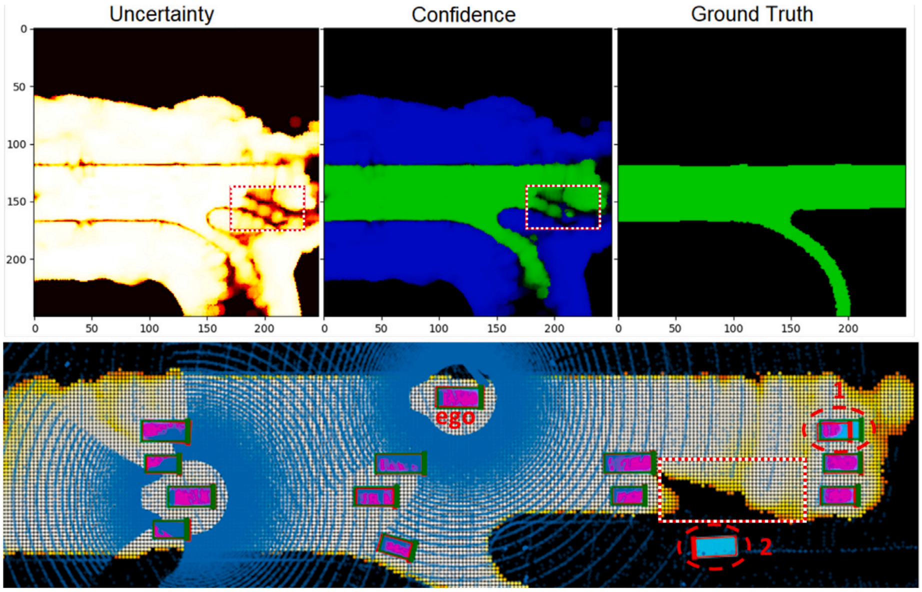

Fig. 10. Results of the GevBEV maps. In the *Uncertainty map*, lower uncertainty is presented in a lighter color. In the *Confidence map*, road confidence is indicated in green and background confidence in blue with different intensities. The bottom sub-figure shows the more detailed detection results in a larger extent. Specifically, the original input point cloud is denoted by blue points, road points are highlighted by a light color if their uncertainties are under the threshold of 0.7, and the objects are shown in bounding boxes with red color indicating the detection, and with green color the corresponding ground truth, respectively. The thick bar in the front of the bounding boxes denotes the driving direction. Moreover, those bounding boxes are filled with the point confidences drawn from the objects head, where magenta points are associated with high confidence, while cyan indicates the opposite. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

detection range, the area marked with a dashed line red rectangle is an area where the ego CAV is not certain about. If the ego CAV needs to drive into this uncertain area, it should either request the missing information from other CAVs or slow down waiting for the clearance of the uncertain area. Second, the object's head generates evidence of the areas that might be occupied by vehicles. Therefore, they are a reliable and explainable clue to validate the bounding box detection. For example, as shown in the bottom sub-figure, most predicted boxes are well aligned with the ground truth except those two marked with red ellipses. The vehicle in ellipse 1 is only partially observed; there is not enough evidence to support this detection. Similarly, there is nearly no evidence for false positive detection in ellipse 2. Therefore, this detection can be simply removed.

Comparison to baseline. Figs. 11 and 12 illustrate the classification confidences, with the color scale transitioning from dark to light, representing confidence values from 0 to 1. In the right column of Fig. 11 and the bottom row of Fig. 12, the red color indicates that our model retains information about unobserved areas. In the absence of observations, the model remains unbiased towards predicting either the foreground or the background class. Additionally, our model exhibits a tendency to produce more refined details for both road and vehicle edges. Notably in Fig. 12, CoBEVT tends to generate more false positive predictions, and in some cases, vehicles even appear merged together. In contrast, our model accurately separates all vehicles, thanks to its precise edge description.

Evidence of object point distribution. Furthermore, we use the average evidence score to better show the relations between the quality of detection and the corresponding object point distributions. Inspired by the work from Wang et al. (2020a), in addition to the normal IoU over the detection areas, we also leverage the generated uncertainty to calculate the JIoU. JIoU is a probability version of IoU that better evaluates the probabilistic features of object detection. Slightly different to Wang et al. (2020a) that defines JIoU as the IoU between the probability mass covered by the detection and the ground truth bounding boxes, we define it as the IoU between the sum of the evidences in the detected bounding boxes and the sum of all the evidence masses describing this object. Then, we calibrate the average evidence score for a single detected bounding box, which is the mean of the drawn confidences of points inside the bounding boxes. This JIoU ensures low evidence score when the predicted box is fully filled with strong evidences but does not cover all evidences that describe this object.

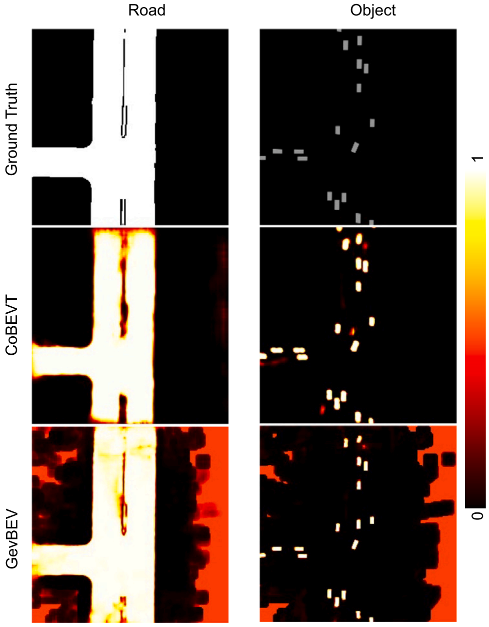

Fig. 11. The comparison of classification confidences between GevBEV and CoBEVT on the OPV2V dataset. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The results in Fig. 13 demonstrate that the average evidence score is very well related to the quality of the detection. As shown in the last row, worse detection tends to have less evidence inside the predicted bounding boxes. Moreover, JIoU reveals the alignment of the prediction and ground truth bounding boxes over the evidence masses. Objects without enough clues from the measurements are hard to define a

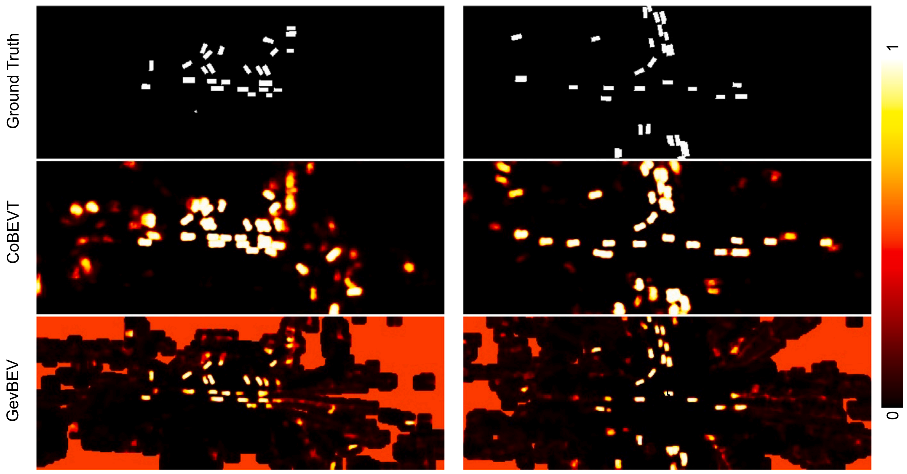

Fig. 12. The comparison of classification confidences between GevBEV and CoBEVT on the V2VReal dataset. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

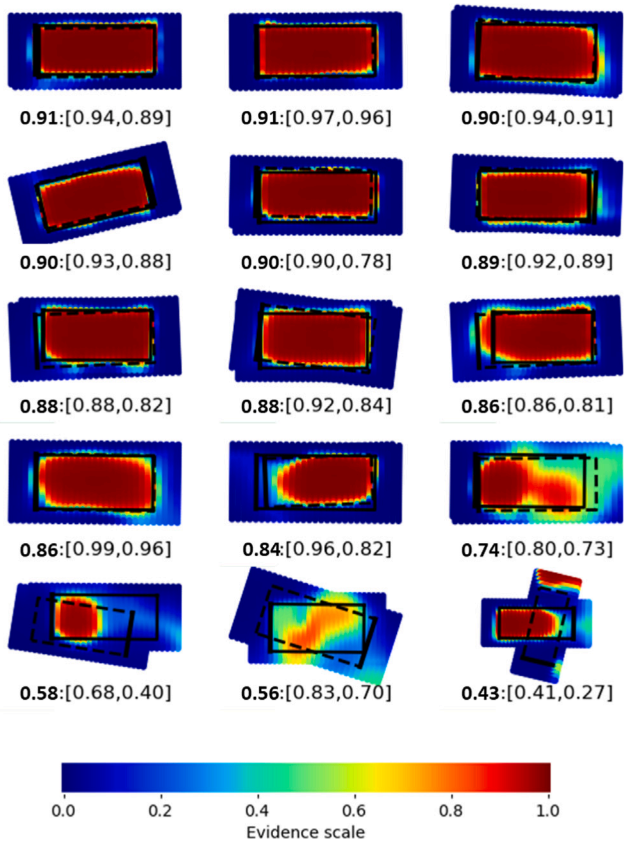

Fig. 13. Object point distribution. The dished line bounding boxes denote the detection, and the solid line bounding boxes denote the corresponding ground truth. The thick bar of each bounding box denotes the driving direction. The statistics under each sub-figure denote the average evidence and [JIoU, IoU].

perfect deterministic ground truth, even by manual labeling, so does fairly judge the model's prediction based on this. In such cases, both detection and ground truth bounding boxes have low evidence coverage over the GevBEV map. This leads to a smaller probabilistic union between these two boxes, hence a higher JIoU is derived. Compared to IoU, this is more reasonable as JIoU decouples the imperfectness of the model and the measurement. For example, the detected bounding box in the lower-left subfigure has low IoU but relatively high JIoU because there are too few evidences caused by the lankness of the observation points – measurement imperfectness. In contrast, compared to the ground truth, the detection in the lower-right subfigure has enough evidences but has both low JIoU and IoU, indicating that the inferior detection is not due to the measurement but the model's limited performance.

Table 3
Average offset of calibration plot to the perfect calibration line. Best values are highlighted in boldface and the second best values are underlined.

Model	OPV2V		V2V4real	Average
	Road	Object	Object	
BEV	0.095	0.072	0.076	0.081
EviBEV	**0.031**	**0.010**	**0.030**	**0.023**
CoBEVT	0.044	0.075	0.056	0.058
GevBEV	0.044	0.066	0.040	0.050

Desired confidence level with calibration plot. We use the calibration plot (Fig. 14) and the average offsets (Table 3) between this plot and the perfect uncertainty-accuracy line (dashed black line) as a summary to analyze the quality of uncertainty generated by the baseline model CoBEVT and BEV, the EviBEV model with $\mathcal{L}_{edl}$ loss, and our complete probabilistic model GevBEV. Since the baseline model only generates Softmax scores for each class, we convert the scores into entropy to quantify the uncertainty and compare it to the other two models with the evidential uncertainty based on a Dirichlet distribution. As revealed in Fig. 14, GevBEV and EviBEV demonstrate better confidence plots to the perfect calibration line (indicated by the diagonal dashed line) than the baseline model BEV and CoBEVT for both road and object classification. The two baseline models seem to overestimate the uncertainty than the other two models, which affirms our concerns that the deterministic model, without particularly accounting for uncertainties, may end up generating less trustworthy scores for making driving decisions. The results, on the other hand, show that assuming a Dirichlet distribution of the point class of the BEV map can provide more reliable probabilistic features for the map and, therefore, is safer to use in AV perception systems.

Interestingly, the uncertainty quality of GevBEV is worse than that of EviBEV, especially for object classification. This might be caused by the saturation of the summation of the evidences contributed by the neighboring center points. Moreover, the highly uncertain points from the objects head of GevBEV tend to be underconfident. We conjecture that some vehicles are only observed partially because of occlusion. Our limited coordinate expansion (1.2 m) is only able to cover parts of these vehicles. Therefore, only the distribution tiles of these expanded

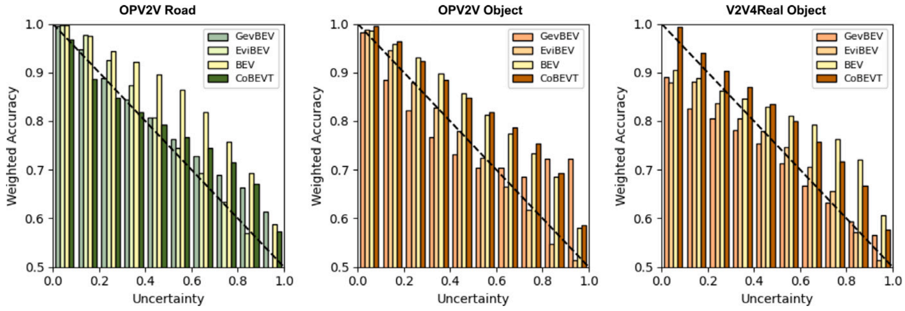

Fig. 14. Calibration plots by different models. The perfect calibration line is indicated by the diagonal dashed line.

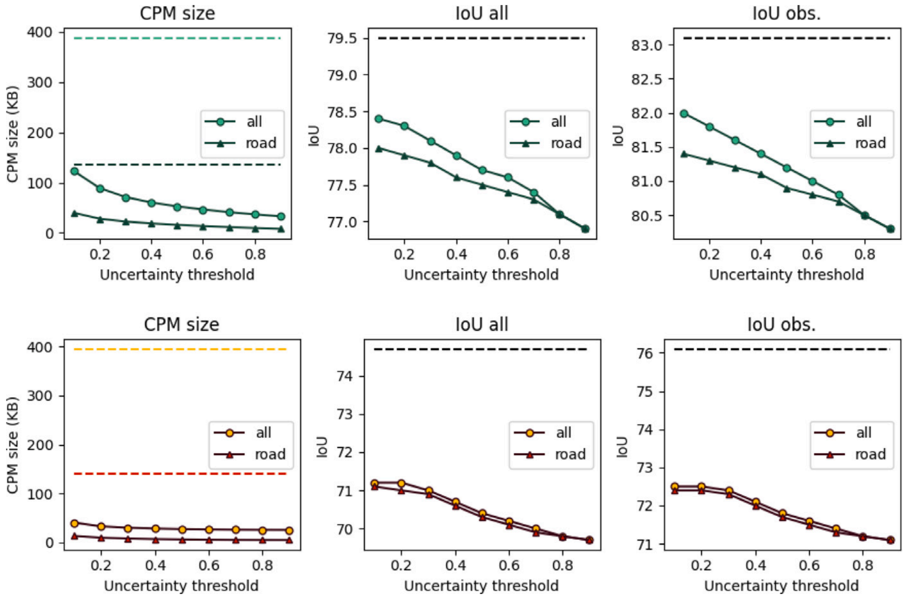

Fig. 15. Comparison of different CPM sharing strategies for co-perception. The first row shows the results of the road head (greenish) and the second the objects head (orangish). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

center points can cover the rest of the vehicle body. This then may lead to a high uncertainty but a high positive rate. Despite the uncertainty of GevBEV being slightly more conservative than that of EviBEV, still, as shown by the higher IoUs in Table 2, the learned spatial Gaussian distribution generates smoother BEV maps and draws classification distribution of any points in the continuous BEV 2D space.

4.7. The application of GevBEV for co-perception

The generated evidential GevBEV maps with different uncertainty thresholds are used to select the information communicated among CAVs. The CPM sizes before and after the uncertainty-based information selection with different uncertainty thresholds, as well as the corresponding IoUs of the classification results over all areas in the perception range (IoU_{all}) and over the observable areas (IoU_{obs}), are plotted in Fig. 15. The first row shows the results of the road head, and the second the objects head. As discussed in Section 3.3, we conducted experiments with two CPM sharing strategies, one for sharing the masked evidence maps of all perception areas (light green and orange) and one only for the road areas constrained by an existing HD map (dark green and orange). The dashed lines are the baselines of sharing CPMs without information selection, which are shown as horizontal lines over different uncertainty thresholds.

The plots in the first column show that the CPM sizes have been reduced evidently after the information selection at all uncertainty thresholds compared to the corresponding baselines. For example, when all perception areas (light green) are considered for sharing, the CPM size for the road head dropped by ca. 87% from 388 kB to 52 kB at the uncertainty threshold of 0.5. Correspondingly, the IoU_{all} and IoU_{obs} only dropped by ca. 2%. In the same configuration, the CPM sizes for the object detection dropped ca. 93%, from 395 kB to 27 kB, while the IoUs dropped ca. 4%. By only considering the road areas for sharing, CPM sizes can be further reduced to about 16 kB for the road head and 6 kB for the objects head at the uncertainty threshold of 0.5, as the dark green and orange solid lines shown with only an IoU drop within 0.5%. According to the V2V communication protocol (Arena and Pau, 2019), without considering other communication overhead, the data throughput rate can achieve 27 Mbps. Therefore, the time delay for sending the CPMs of both heads has dropped from ca. 28 ms to 0.8 ms by the data selection based on our GevBEV maps from road areas. These significantly reduced time delays are critical for real-time V2V communication.

5. Conclusion

In this paper, we proposed a novel method to interpret driving environments with observable Gaussian evidential BEV maps. These maps interpret the LiDAR sensory data in a back-traceable manner that each prediction is supported by evidences provided by the original observation points. Moreover, we designed a probabilistic classification model based on U-Net to generate statistics for this interpretation. This model assumes a spatial Gaussian distribution for each voxel of a predefined resolution so that any point in the continuous driving space can draw itself a Dirichlet distribution of the classification based on the evidences drawn from the spatial Gaussian distributions of the neighboring voxels.

We test our proposed GevBEV on benchmarks OPV2V and V2V4Real of BEV map interpretation for cooperative perception in simulated and real-world driving scenarios, respectively. The experiments show that GevBEV outperforms the baseline of the image-based BEV map by a large margin. By analyzing the predictive uncertainty, we also proved that evidential classification can score the classification result in a less overconfident and better-calibrated manner than the deterministic counterpart of the same model. Furthermore, the spatial Gaussian distribution assigned to each observable point is also proven beneficial in closing the gaps of sparse point clouds with a controllable range and smoothing the BEV maps. By virtue of this spatial distribution, one can draw the Dirichlet classification result for any points in the continuous driving space. This probabilistic result can be used to make safer decisions for autonomous driving by its ability to quantify the uncertainty using the measurement evidences.

Although our model is only applied to the point cloud data of the co-perception scenario in this paper, it is straightforward to be used on other modalities of data sources or scenarios, such as images for co-perception, or satellite images and airborne point clouds in the field of remote sensing. We leave these applications for future work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 227198829/GRK1931 and MSCA European Postdoctoral Fellowships under the 101062870 – VeVuSafety project.

References

Arena, F., Pau, G., 2019. An overview of vehicular communications. Future Internet 11, 27.

Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O., 2020. nuScenes: A multimodal dataset for autonomous driving. In: Conference on Computer Vision and Pattern Recognition (CVPR).

Chen, Q., 2019. F-cooper: feature based cooperative perception for autonomous vehicle edge computing system using 3D point clouds. In: Proceedings of the 4th ACM/IEEE Symposium on Edge Computing.

Choy, C., Gwak, J., Savarese, S., 2019. 4D spatio-temporal ConvNets: Minkowski convolutional neural networks. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3075–3084.

Cui, J., Qiu, H., Chen, D., Stone, P., Zhu, Y., 2022. Coopernaut: End-to-end driving with cooperative perception for networked vehicles. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 17231–17241.

Dempster, A.P., 2008. A generalization of Bayesian inference. In: Classic Works of the Dempster-Shafer Theory of Belief Functions.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V., 2017. CARLA: An open urban driving simulator. In: Conference on Robot Learning. PMLR, pp. 1–16.

Fang, L., You, Z., Shen, G., Chen, Y., Li, J., 2022. A joint deep learning network of point clouds and multiple views for roadside object classification from LiDAR point clouds. ISPRS J. Photogramm. Remote Sens. 193, 115–136.

Feng, D., Cao, Y., Rosenbaum, L., Timm, F., Dietmayer, K.C.J., 2020. Leveraging uncertainties for deep multi-modal object detection in autonomous driving. In: IEEE Intelligent Vehicles Symposium (IV). pp. 877–884.

Feng, D., Harakeh, A., Waslander, S.L., Dietmayer, K., 2022. A review and comparative study on probabilistic object detection in autonomous driving. IEEE Trans. Intell. Transp. Syst. 23 (8), 9961–9980.

Feng, D., Rosenbaum, L., Timm, F., Dietmayer, K.C.J., 2019. Leveraging heteroscedastic aleatoric uncertainties for robust real-time LiDAR 3D object detection. In: IEEE Intelligent Vehicles Symposium (IV). pp. 1280–1287.

Gal, Y., 2016. Uncertainty in deep learning. PhD thesis.

Gawlikowski, J., Tassi, C.R.N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A.M., Triebel, R., Jung, P., Roscher, R., Shahzad, M., Yang, W., Bamler, R., Zhu, X., 2021. A survey of uncertainty in deep neural networks. arXiv arXiv:2107.03342.

Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR).

Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., Qu, R., 2019. A survey of deep learning-based object detection. IEEE Access 7, 128837–128868.

Kendall, A., Gal, Y., 2017. What uncertainties do we need in Bayesian deep learning for computer vision? In: Advances in Neural Information Processing Systems (NeurIPS).

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. In: ICLR.

Kirillov, A., He, K., Girshick, R.B., Rother, C., Dollár, P., 2018. Panoptic segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9396–9405.

Lafferty, J.D., McCallum, A., Pereira, F., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: International Conference on Machine Learning.

Lakshminarayanan, B., Pritzel, A., Blundell, C., 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In: Advances in Neural Information Processing Systems (NeurIPS).

Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O., 2019. PointPillars: Fast encoders for object detection from point clouds. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12689–12697.

Li, Y., Ren, S., Wu, P., Chen, S., Feng, C., Zhang, W., 2021. Learning distilled collaboration graph for multi-agent perception. In: Advances in Neural Information Processing Systems (NeurIPS), Vol. 34. pp. 29541–29552.

Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Yu, Q., Dai, J., 2022a. BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In: European Conference on Computer Vision.

Li, J., Xu, R., Ma, J., Zou, Q., Ma, J., Yu, H., 2022b. Domain adaptive object detection for autonomous driving under foggy weather. In: IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 612–622.

Lin, Y., Vosselman, G., Cao, Y., Yang, M.Y., 2021. Local and global encoder network for semantic segmentation of airborne laser scanning point clouds. ISPRS J. Photogramm. Remote Sens. 176, 151–168.

Loukkal, A., Grandvalet, Y., Drummond, T., Li, Y., 2021. Driving among flatmobiles: Bird-eye-view occupancy grids from a monocular camera for holistic trajectory planning. In: IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 51–60.

Lu, C., van de Molengraft, M.J.G., Dubbelman, G., 2019. Monocular semantic occupancy grid mapping with convolutional variational encoder–decoder networks. IEEE Robot. Autom. Lett. 4, 445–452.

Mackay, D.J.C., 1991. A practical Bayesian framework for backprop networks. Neural Comput..

Malinin, A., Gales, M.J.F., 2018. Predictive uncertainty estimation via prior networks. In: Advances in Neural Information Processing Systems (NeurIPS).

Meyer, G.P., Laddha, A.G., Kee, E., Vallespi-Gonzalez, C., Wellington, C.K., 2019. LaserNet: An efficient probabilistic 3D object detector for autonomous driving. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12669–12678.

Miller, D., Dayoub, F., Milford, M., Sünderhauf, N., 2019. Evaluating merging strategies for sampling-based uncertainty techniques in object detection. In: International Conference on Robotics and Automation (ICRA). pp. 2348–2354.

Neal, R.M., 1995. Bayesian learning for neural networks.

Pan, B., Sun, J., Leung, H.Y.T., Andonian, A., Zhou, B., 2020a. Cross-view semantic segmentation for sensing surroundings. IEEE Robot. Autom. Lett. 5, 4867–4873.

Pan, H., Wang, Z., Zhan, W., Tomizuka, M., 2020b. Towards better performance and more explainable uncertainty for 3D object detection of autonomous vehicles. In: International Conference on Intelligent Transportation Systems (ITSC). pp. 1–7.

Qiu, H., Yu, B., Tao, D., 2022. GFNet: Geometric flow network for 3D point cloud semantic segmentation. Trans. Mach. Learn. Res. (ISSN: 2835-8856).

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 234–241.

Sensoy, M., Kaplan, L., Kandemir, M., 2018. Evidential deep learning to quantify classification uncertainty. In: Advances in Neural Information Processing Systems (NeurIPS), Vol. 31.

Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., Anguelov, D., 2020. Scalability in perception for autonomous driving: Waymo open dataset. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2443–2451.

Vasudevan, V.T., Sethy, A., Ghias, A.R., 2019. Towards better confidence estimation for neural models. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 7335–7339.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. In: Advances in Neural Information Processing Systems (NeurIPS), Vol. 30.

Wang, Z., Feng, D., Zhou, Y., Zhan, W., Rosenbaum, L., Timm, F., Dietmayer, K.C.J., Tomizuka, M., 2020a. Inferring spatial uncertainty in object detection. In: International Conference on Intelligent Robots and Systems (IROS). pp. 5792–5799.

Wang, T.-H., Manivasagam, S., Liang, M., Yang, B., Zeng, W., Urtasun, R., 2020b. V2vnet: Vehicle-to-vehicle communication for joint perception and prediction. In: European Conference on Computer Vision. Springer, pp. 605–621.

Xu, R., Guo, Y., Han, X., Xia, X., Xiang, H., Ma, J., 2021. OpenCDA: an open cooperative driving automation framework integrated with co-simulation. In: International Intelligent Transportation Systems Conference (ITSC). pp. 1155–1162.

Xu, R., Li, J., Dong, X., Yu, H., Ma, J., 2022a. Bridging the domain gap for multi-agent perception. arXiv preprint arXiv:2210.08451.

Xu, R., Tu, Z., Xiang, H., Shao, W., Zhou, B., Ma, J., 2022b. CoBEVT: Cooperative bird's eye view semantic segmentation with sparse transformers. In: Conference on Robot Learning (CoRL).

Xu, R., Xia, X., Li, J., Li, H., Zhang, S., Tu, Z., Meng, Z., Xiang, H., Dong, X., Song, R., Yu, H., Zhou, B., Ma, J., 2023a. V2V4Real: A real-world large-scale dataset for vehicle-to-vehicle cooperative perception. In: The IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR).

Xu, R., Xiang, H., Han, X., Xia, X., Meng, Z., Chen, C.-J., Ma, J., 2023b. The OpenCDA open-source ecosystem for cooperative driving automation research. arXiv preprint arXiv:2301.07325.

Xu, R., Xiang, H., Tu, Z., Xia, X., Yang, M.-H., Ma, J., 2022c. V2X-ViT: Vehicle-to-everything cooperative perception with vision transformer. arXiv preprint arXiv:2203.10638.

Xu, R., Xiang, H., Xia, X., Han, X., Li, J., Ma, J., 2022d. OPV2V: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In: IEEE International Conference on Robotics and Automation (ICRA).

Yang, B., Luo, W., Urtasun, R., 2018. PIXOR: Real-time 3D object detection from point clouds. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7652–7660.

Yuan, Y., Cheng, H., Sester, M., 2022. Keypoints-based deep feature fusion for cooperative vehicle detection of autonomous driving. IEEE Robot. Autom. Lett. PP, 1.

Zang, A., Xu, R., Li, Z., Doria, D., 2017. Lane boundary extraction from satellite imagery. In: Proceedings of the 1st ACM SIGSPATIAL Workshop on High-Precision Maps and Intelligent Applications for Autonomous Vehicles. pp. 1–8.

Zhang, Y., Carballo, A., Yang, H., Takeda, K., 2023. Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. ISPRS J. Photogramm. Remote Sens. 196, 146–177.

Zheng, W., Tang, W., Chen, S., Jiang, L., Fu, C.-W., 2021. CIA-SSD: Confident iou-aware single-stage object detector from point cloud. In: AAAI.

Zhou, B., Krähenbühl, P., 2022. Cross-view transformers for real-time map-view semantic segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR).

Zhou, Y., Tuzel, O., 2018. VoxelNet: End-to-end learning for point cloud based 3D object detection. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4490–4499.