Uncertainty-Aware Maritime Point Cloud Detector (U-MPCD) for Autonomous Surface Vehicles

Yongchang Xie, Peng Wu, *Member, IEEE*, Brendan Englot, *Senior Member, IEEE*, Cassandra Nanlal, and Yuanchang Liu, *Member, IEEE*

***Abstract*—Autonomous surface vehicles (ASVs) operating in busy and constrained maritime environments (e.g., inland waterways, harbors, ports, and marinas) require robust perception modules for real-time boat detection, with LiDAR serving as one of the practical sensors for environmental perception. However, these environments present challenges, such as large variations in boat sizes, sparse point cloud data at longer distances, and occlusions from the restricted field of view of onboard LiDAR and surrounding obstacles, leading to high predictive uncertainty. Small boats rely on local features (e.g., fine-grained geometric details), while large boats require global features (e.g., overall shape and structural continuity) for accurate detection. To address these challenges, we propose the maritime point cloud detector (MPCD), which integrates an attention-based point feature net for pillar-level local feature extraction and a hybrid 2-D backbone combining multiscale MobileViT with a 2-D convolutional neural network for enhanced global feature learning, achieving a 12.8% improvement in detection accuracy over the baseline. To further enhance reliability, we extend MPCD with the multi-input multi-output method, forming uncertainty-aware MPCD (U-MPCD). U-MPCD estimates both epistemic and aleatoric uncertainties, improves detection accuracy by 2%, and maintains an inference speed of 15 Hz, providing critical insights into prediction confidence for safer ASV navigation. Our model was tested on real-world data sets collected under normal hydrographic survey conditions (6 h per day over four days, covering about 11.4 km) along the River Thames in central London, which features high maritime traffic and diverse boat types and sizes.**

***Index Terms*—3-D point cloud, autonomous surface vehicle (ASV), environmental perception, object detection, predictive uncertainty.**

I. Introduction

ENVIRONMENTAL perception is essential for autonomous surface vehicles (ASVs) to operate in busy and constrained maritime environments (e.g., inland waterways, harbors, ports, marinas, etc.) with higher levels of autonomy. By detecting and tracking both moving and stationary objects in real time, ASVs obtain the critical information needed for mission planning and decision making. To achieve this, they often rely on sensors, such as cameras, LiDAR, and radar. Cameras offer rich semantic information but face significant challenges in maritime environments, including water reflections, dazzling light, and sea fog [1]. Maritime radar is robust in various weather conditions and has a greater detection range than cameras; however, its low update frequency and limited semantic information restrict real-time object detection in busy maritime settings. LiDAR bridges these gaps by providing high-frequency updates, more reliable performance than cameras in many weather conditions (though it can be affected by foggy or rainy conditions due to scattering effects), and rich point cloud data, making it particularly suitable for maritime perception [2]. LiDAR generates 3-D point cloud scans that contain each point's distance and reflection intensity, offering a detailed representation of the surrounding environment. Nevertheless, the sparsity, unstructured nature, and unordered format of 3-D point cloud data pose unique challenges for processing tasks, such as feature extraction and object detection.

In recent years, intensive research and the deployment of deep learning methods have significantly advanced 3-D point cloud-based object detection. Point cloud detection models are generally categorized into three types based on their data processing strategies: point-based, voxel-based, and projection-based methods [3]. Point-based methods directly process raw point cloud data, preserving geometric details and achieving high accuracy, but their considerable computational demands make them less suitable for real-time applications [4]. Voxel-based methods convert point clouds into structured 3D voxels, enabling the use of sparse 3D convolutional neural networks (CNNs) for efficient feature extraction. Although these methods are computationally more efficient and can handle larger scenes, fixed voxel resolutions can result in the loss of fine-grained information, particularly for smaller objects [5], [6]. Projection-based methods project 3-D point clouds onto 2-D planes—such as a bird's-eye view (BEV)—and apply 2-D CNNs. This approach leverages well-established 2-D CNN architectures to achieve faster inference times than the other approaches, but it often compromises spatial and geometric fidelity [7].

Maritime environments present unique challenges due to the wide range of target sizes, from small buoys (less than 1 m)

Received 11 March 2025; revised 22 August 2025; accepted 7 September 2025. This work was supported in part by the Port of London Authority (PLA) and in part by the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/Y000862/1 and Grant EP/X034909/1. *(Corresponding author: Yuanchang Liu.)*

Associate Editor: A. Munafò.

Yongchang Xie, Peng Wu, and Yuanchang Liu are with the Department of Mechanical Engineering, University College London, WC1E 7JE London, U.K. (e-mail: yuanchang.liu@ucl.ac.uk).

Brendan Englot is with the Department of Mechanical Engineering, Stevens Institute of Technology, Hoboken, NJ 07030 USA.

Cassandra Nanlal is with the Department of Civil, Environmental and Geomatic Engineering, University College London, WC1E 6BT London, U.K.

Data set is available online at https://figshare.com/articles/dataset/Maritime_LiDAR-based_Ship_Detection_Datasets/26144536.

Digital Object Identifier 10.1109/JOE.2025.3612726

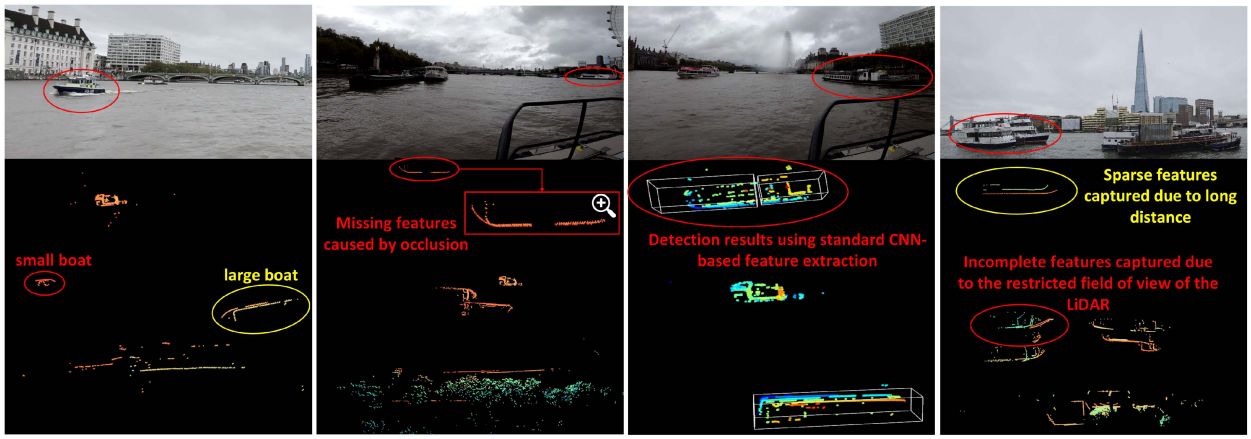


Fig. 1. Illustration of challenges in maritime point cloud detection. (a) Distinct local and global features for a small boat (red oval) and a large boat (yellow oval). (b) Occlusion splitting the global structure of a large boat. (c) Fragmented detection of a large boat due to CNN backbone limitations (white oriented bounding boxes show the visualized detection results). (d) Sparse point cloud features caused by distance (yellow oval) and incomplete point cloud features caused by occlusions (red oval).

to large ships (spanning hundreds of meters). Achieving robust performance in such settings requires point cloud detectors to effectively capture both local and global features. In voxel-based methods, local features refer to the point cloud characteristics within individual voxels or across a small group of neighboring voxels, whereas global features represent aggregated information linking numerous voxel features to capture broader spatial context. As shown in Fig. 1(a), the point cloud features of a small boat (highlighted in the red oval) form a distinct geometric shape but occupy only a small region. This distinct shape can be considered a local feature, and detecting small boats requires precise local feature extraction to identify their unique geometry. In contrast, the point cloud features of a large boat (highlighted in the yellow oval) span a much larger region, necessitating strong global feature extraction to understand the overall structure and spatial layout of the boat in the scene. Furthermore, as shown in Fig. 1(b), the point cloud features of a large boat are split into two parts due to occlusion. In such cases, robust global feature extraction is critical to link these discrete parts, enabling a comprehensive understanding of the boat's overall hull structure. This challenge underscores the need for models that can effectively bridge the gap between local precision and global comprehension. However, to meet real-time performance constraints, most point cloud detectors primarily rely on CNN-based architectures. While CNNs can effectively extract local features, such as edges, corners, and fine-grained structure, their ability to capture global features, which are crucial for broader spatial understanding, remains limited [8].

To address this limitation, various techniques, such as multiscale feature fusion [3], have been proposed. These methods combine feature maps extracted at different resolutions or layers of CNNs, thereby enhancing CNNs' ability to represent global features. Despite these improvements, they only partially mitigate the limitations of CNN-based approaches and remain insufficient for the diversity of objects in maritime environments. As shown in Fig. 1(c), the detection results generated by PointPillar, which uses a CNN backbone, incorrectly identify the large boat in the red oval as two smaller boats due to missing features in the middle. This outcome highlights the model's limitation in capturing global features to connect the fragmented parts of the object. Beyond CNN-based architectures, numerous studies have explored transformer-based methods for point cloud processing. Transformers leverage attention mechanisms to capture global features effectively, as illustrated by models like Point Transformer [9] and VoxSet [10]. However, they are computationally more demanding than CNNs, making them less practical for real-time applications.

In addition to the wide range of target sizes, point cloud data captured in maritime environments often suffer from sparse features due to long-distance targets and incomplete features caused by occlusions from the restricted field of view of the LiDAR and the presence of other objects, as shown in Fig. 1(d). These challenges lead to higher detection uncertainty in the model's predictions. Most existing detectors produce deterministic predictions and therefore cannot estimate the probability of perception errors. However, capturing these errors or uncertainties is critical for ensuring safe ASV operations. Such uncertainties, arising from perception inaccuracies or sensor noise, provide valuable insights into perception performance and enable autonomous systems to adapt accordingly. Moreover, reliable uncertainty estimation enhances human interpretability of autonomous systems' decisions, fostering trust in this rapidly evolving technology [11]. Bayesian neural networks (BNNs) offer a principled approach to generating accurate uncertainty estimates, making them a promising solution for safety-critical applications. The key concept of a BNN is to assign a prior probability distribution to network parameters and use Bayesian inference to estimate the posterior distribution [12]. However, because Bayesian inference for large-scale networks is computationally expensive and often impractical, much research focuses on approximation methods that provide uncertainty

estimates while keeping computational overhead manageable. Examples of such practical methods include Monte Carlo Dropout (MC-Dropout) [13], deep ensembles [14], and multi-input multi-output (MIMO) [15].

In this article, we propose uncertainty-aware maritime point cloud detector (U-MPCD), a novel point cloud detection network specifically designed for maritime environments, with a focus on boat detection. U-MPCD improves detection accuracy by effectively capturing both local and global features. Local features—such as the fine-grained geometric details of a boat's hull, edges, or superstructure—are particularly important for small boat detection, while global features—such as the overall shape and structural continuity—are crucial for accurately detecting large boats. In addition, U-MPCD provides predictive uncertainty estimates to enhance model reliability in real-time maritime scenarios. We adopt PointPillar as our baseline due to its fast inference speed and competitive detection accuracy. To address the unique challenges of maritime environments, our key contributions are as follows.

1) Considering the sparse, large-scale, and complex distribution of maritime targets, we first propose maritime point cloud detector (MPCD), a novel deterministic point cloud detection model. MPCD incorporates a multihead attention mechanism within the pillar feature network to enhance local feature encoding in sparse point clouds, allowing the model to capture finer geometric details of target objects. In addition, the global context modeling capability of a Transformer architecture—specifically MobileViT—is integrated into the 2-D CNN backbone. This combination effectively merges local and global information, significantly improving detection of large-scale and distant targets while adding only a slight increase in computational cost.
2) To enable predictive uncertainty estimation, we integrate the MIMO uncertainty estimation approach into MPCD, forming our final model, U-MPCD. This method generates multiple prediction outcomes within a single forward pass, offering efficient uncertainty estimation while preserving the computational efficiency required for real-time applications.
3) We expanded an existing real-world maritime LiDAR data set [2] by deploying LiDAR on a survey boat to collect additional data, particularly focusing on large boat targets in motion. This effort enriched the data set with more diverse targets and scenarios. We collected nearly 150 000 frames in total, of which 1000 were labeled.
4) We validate our model on a real-world LiDAR data set containing various types of boats and conduct extensive ablation studies to evaluate the impact of specific design choices. Experimental results show that our proposed model outperforms the baseline in detection accuracy, effectively estimates predictive uncertainty, and maintains high inference speed, making it suitable for real-time applications.

The rest of this article is organized as follows. Section II reviews related work, while Section III details the methods employed in our approach. Section IV evaluates and discusses the results of the proposed method, and Section V presents a detailed ablation study to support our design choices. Finally, Section VI concludes this article.

II. Related Work

A. Deep Learning on Point Cloud

The pipeline structure of point cloud detection models comprises three stages: data representation, feature extraction, and the dense head [3]. In the data representation stage, high-dimensional and unstructured point clouds are transformed into a structured format that can be processed in subsequent stages. A variety of strategies have been explored in the literature for this representation, including point-based [4], [25], [26], voxel-based [5], [6], [7], [21], [22], projection-based [18], [19], [20], and hybrid approaches [23], [24].

Point-based methods directly process raw point clouds, resulting in a sparse structure where each point is associated with a feature vector derived from its neighbors [4], [25]. Voxel-based methods divide the point cloud into a grid of uniformly spaced voxels. Each voxel contains a subset of points, allowing the network to learn local and global features on a per-voxel basis, thus reducing overall dimensionality and computational load [5], [6]. A variant of voxel-based frameworks is the pillar-based method, illustrated by PointPillar [7], which organizes points into pillars in the x–y plane, disregarding the z dimension. This simplification further reduces computational complexity during feature extraction while preserving sufficient spatial information in the horizontal plane. Projection-based approaches convert the 3-D point cloud into 2-D images, aiming to reduce the cost of direct 3-D processing. This approach can leverage mature 2-D CNN frameworks and often yields faster inference, but risks losing certain 3-D spatial cues and can introduce distortions or occlusion issues in the projected views [3].

Feature extraction is a critical stage in 3-D object detection and directly influences both accuracy and efficiency in tasks, such as bounding-box estimation and classification (CLS). Research in this area has introduced various strategies that can be broadly grouped into point-wise, segment-based, and convolutional (2-D or 3-D) approaches. In point-wise methods, such as PointNet [4] and PointNet++ [25], low-dimensional features are derived from each point independently and then aggregated to form higher dimensional descriptors. While these techniques effectively capture fine-grained details, they tend to increase runtime by operating on every point in the cloud. In contrast, segment-based approaches partition the point cloud into volumetric segments (either voxels or pillars), thereby reducing the overall number of points to be processed. One prominent example is the voxel feature extractor (VFE) [6], [27], which extends PointNet concepts to volumetric data by stacking multiple set abstraction layers that expand the receptive field and incorporate richer contextual information [5], [6]. This coarse subdivision often yields faster inference times while maintaining sufficient spatial resolution. Although segment-based feature extraction is crucial for encoding fine-grained local features within individual voxels, it requires an additional CNN to capture spatial relationships and integrate local and global information.

TABLE I
OVERVIEW OF POINT CLOUD-BASED DETECTION NETWORKS DEVELOPED FOR OUTDOOR AUTONOMOUS DRIVING SCENARIOS

Model	Data Representation	Feature Extraction
PointRCNN [16]	Points	PointNet++
3DSSD [17]	Points	PointNet++ (with feature FPS sampling)
RT3D [18]	Projection	2-D CNN
HDNet [19]	Projection	Data Augmentation → 2-D CNN
PIXOR [20]	Projection	Light Data Augmentation → 2-D CNN
VoxelNet [6]	Voxel	VFE → 3-D CNN → collapse to BEV → 2-D CNN
SECOND [5]	Voxel	VFE → 3-D Sparse CNN → collapse to BEV → 2-D CNN
Voxel-FPN [21]	Voxel	VFE → multi scale FPN-like structure in 2-D BEV
PointPillar [7]	Pillar	PointNet → collapse to BEV → 2-D CNN
Part A2 Net [22]	Voxel	3-D Sparse CNN & Region Proposals in Voxel
PV-RCNN [23]	Hybrid (Voxel and Points)	PointNet → collapse to BEV → 2-D CNN and PointNet
STD [24]	Hybrid (Voxel and Points)	3-D CNN & PointNet++

Feature extraction outlines the sequential methods used to derive features from the 3-D point cloud, which are then passed to a dense head for the final detection stage.

Convolutional approaches adapt established 2-D or 3-D CNNs to learn features from point clouds represented in structured formats, such as voxel grids or 2-D projections. 2-D CNN backbones, such as visual geometry group (VGG) network [28], residual network (ResNet) [29], or densely connected convolutional network (DenseNet) [30], were originally developed for image-based tasks but also perform effectively when the point cloud is collapsed into a 2-D pseudoimage. This collapse can occur in projection-based methods (e.g., HDNet) or within voxel and pillar frameworks (e.g., VoxelNet, SECOND, and PointPillars) by collapsing the data along one axis into a 2-D grid. These networks employ 2-D convolution for feature extraction and utilize a dense head for the final detection stage. On the other hand, 3-D convolutions provide a more comprehensive understanding of spatial relationships by preserving the 3-D voxel structure instead of collapsing it into a 2-D grid, but they are computationally demanding, particularly for sparse LiDAR scans. Consequently, sparse convolution methods [31] have been introduced to focus computational effort on active regions and avoid redundant operations in empty space. This choice significantly reduces runtime and makes CNN-based feature extraction more practical for large-scale, sparse point-cloud data.

Table I provides an overview of several point cloud detection networks developed for autonomous vehicles in recent years, highlighting their respective data representations and feature extraction strategies. This survey places emphasis on networks that rely solely on multilayer perceptron (MLP)- and CNN-based feature extraction. By examining these foundational works, one can identify suitable candidates to serve as baseline models. For outdoor point cloud detection, the speed of inference is a key priority. Consequently, point-based and hybrid (two-stage) methods were excluded from the outset. Projection-based approaches were also set aside due to the distortion and occlusion they can introduce when transforming 3-D data into 2-D views. While voxel-based methods ultimately compress 3-D point clouds into 2-D pseudoimages in their final processing steps, they first learn features from each point within a voxel. This intermediate step preserves spatial and geometric details more effectively than projection-based techniques.

Among voxel-based methods, PointPillars stands out for its efficiency. Unlike other voxel-based networks that heavily rely on computationally intensive 3-D CNNs, PointPillars does not use 3-D convolutions, substantially reducing computational requirements. Nevertheless, it still utilizes PointNet to capture point-level features within each pillar, balancing detection accuracy with inference speed. Consequently, PointPillars is selected here as the baseline model. Building on this foundation, our goal is to develop a maritime-specific point cloud detector that addresses the unique challenges of maritime conditions while maintaining real-time performance and reliable detection accuracy.

B. *Self-Attention and Transformer*

In recent years, the Transformer architecture [32], which relies on self-attention mechanisms, has achieved significant success in various vision tasks. Vision transformer (ViT) was the first to extensively apply the Transformer framework to image processing and quickly became a strong alternative to CNNs [33]. The key idea of ViT lies in leveraging multihead self-attention to capture long-range dependencies, thereby extracting more global visual features. Building on this foundation, many ViT variants have been proposed. For example, Swin Transformer [34] adopts a hierarchical strategy with shifted window attention to progressively process feature maps, effectively reducing computational overhead while maintaining local and global information across multiple scales. DeiT [35] addresses ViT's dependence on large-scale data sets by incorporating knowledge distillation and efficient data augmentation, enabling it to achieve strong results even on smaller data sets. Meanwhile, MobileViT [36] is optimized for lightweight scenarios, retaining the Transformer's ability to model long-range dependencies while reducing network depth and parameter size for mobile or resource-constrained devices.

Inspired by these developments, recent researches began adapting the Transformer architecture to 3-D point clouds. Since point clouds are unordered and sparse, applying a standard Transformer directly poses significant challenges in terms of

both computational cost and data sparsity [37]. In the 3-D point cloud scenario, Transformer applications broadly split into two directions: those based on voxel representations, and those based on raw point-wise representations. A notable point-based approach is Point Transformer [9], [38], [39], which applies an enhanced multihead self-attention mechanism directly to points, thereby capturing both local and global geometric information. In contrast, voxel-based methods include VoTr [40], which uses local and dilated attention on sparse voxels, along with custom voxel queries to efficiently model sparse voxel grids. Extending this line of work, VoxSeT [10] introduces a voxel-based set attention module that divides a single global self-attention step into two cross-attention operations, modeling features in a latent code-induced hidden space and thus preserving global dependencies. DSVT [41] further evolves the "Voxel + Transformer" paradigm with dynamic sparse window attention and a rotated set partitioning strategy to improve efficiency. Pyramid vision transformer (PVT) [42] demonstrates a hybrid voxel-based approach. It processes 3-D voxels using sparse window attention and relative-attention modules, integrating efficient sparse computations with fine-grained point-level modeling within a purely Transformer-driven framework. Rather than applying Transformers directly to 3-D point clouds or voxel representations, some research projects the point cloud onto a 2-D pseudoimage and then employs standard Transformers, such as ViT. For instance, RangeViT [43] uses LiDAR range-view representations in conjunction with the ViT framework, enabling comprehensive modeling of extensive scenes and thereby enhancing the accuracy of 3-D semantic segmentation. By combining the efficiency of range-view projection with the long-range attention capabilities of Transformers, this method offers superior fine-grained recognition for autonomous driving scenarios.

In general, Transformers offer stronger global context modeling than CNNs, whether applied at the 3-D voxel or point level, or after projecting the point cloud into a 2-D pseudoimage for ViT-based feature extraction. However, implementing Transformers directly on raw 3-D voxels or points typically leads to higher computational costs, prompting some methods to convert point clouds into 2-D representations to mitigate sparsity and leverage established ViT frameworks. Our approach follows this strategy by integrating a well-established ViT into the 2-D CNN backbone to enhance global feature extraction. Meanwhile, we incorporate a multihead attention layer at the pillar-level feature extraction stage, enabling a more refined local feature representation within each pillar.

C. Bayesian Reasoning

Uncertainty in model predictions can be decomposed into epistemic uncertainty and aleatoric uncertainty. Epistemic uncertainty arises from the model's capacity to explain the observed data sets, whereas aleatoric uncertainty comes from inherent noise within the data itself [11]. Many researches in deep learning have focused on methods to estimate predictive uncertainties. Among these, BNNs [12], which infer posterior distributions of network weights from data sets and combine them with observed data to form a predictive distribution, have long been regarded as the gold standard for probabilistic inference in neural networks. While the posterior distribution can be analytically computed in simple cases, large-scale and deep network architectures typically exhibit high-dimensional, multimodal posteriors that make exact solutions impractical. Consequently, various approximate inference techniques have been proposed, including Variational Inference [44], Markov Chain Monte Carlo [45], Stochastic Gradient Descent approximations [46], and Laplace approximations [47]. However, these methods are often highly sensitive to hyperparameters and struggle to scale effectively with large data sets or complex network architectures [48]. As a result, selecting appropriate model priors and inference techniques remains a significant challenge.

In response to these challenges, several practical approaches have been proposed in recent years, commonly involving some form of model ensemble. Such techniques draw multiple model instances from an approximate posterior and aggregate their outputs to estimate the predictive distribution. For example, MC-Dropout [13] leverages dropout at inference as an approximation of Bayesian variational inference: by maintaining dropout during multiple forward passes, one can approximate the predictive distribution. Deep ensembles [14] similarly estimate predictive probabilities by viewing each model's output as an independent sample from a mixture distribution. However, these methods experience increased computational overhead due to the repeated forward passes required. To address this overhead, the multiple-input multiple-output (MIMO) framework was introduced [49], enabling the integration of several subnetworks within a single model and thus producing multiple predictions through a single forward pass. Building on this idea, LiDAR-MIMO [15] was developed for LiDAR-based 3-D detection. This method improves computational efficiency by duplicating 2-D pseudoimages rather than replicating raw point clouds, and by incorporating multiple dense heads to generate several predictions concurrently. Consequently, uncertainty estimation can be performed within a single forward pass.

Drawing on these techniques, we integrate MIMO methods into the proposed deterministic detection networks, enabling them to capture inherent uncertainties in their predictions. This integration promises more robust and interpretable solutions for 3-D object detection in ASVs, ultimately enhancing safety while supporting real-time applications.

III. Methodology

In this section, we introduce U-MPCD for 3D object detection in maritime applications. The first subsection describes how an attention mechanism and a Transformer-based backbone are integrated to produce deterministic detection results, referred to as MPCD. The subsequent subsection presents the Bayesian inference design for MPCD, enabling the model to capture predictive uncertainty, thereby yielding U-MPCD.

A. Maritime Point Cloud Detector

Most existing 3-D point cloud detectors for outdoor environments are specifically developed for autonomous driving scenarios, targeting objects, such as cars, cyclists, and pedestrians [50].

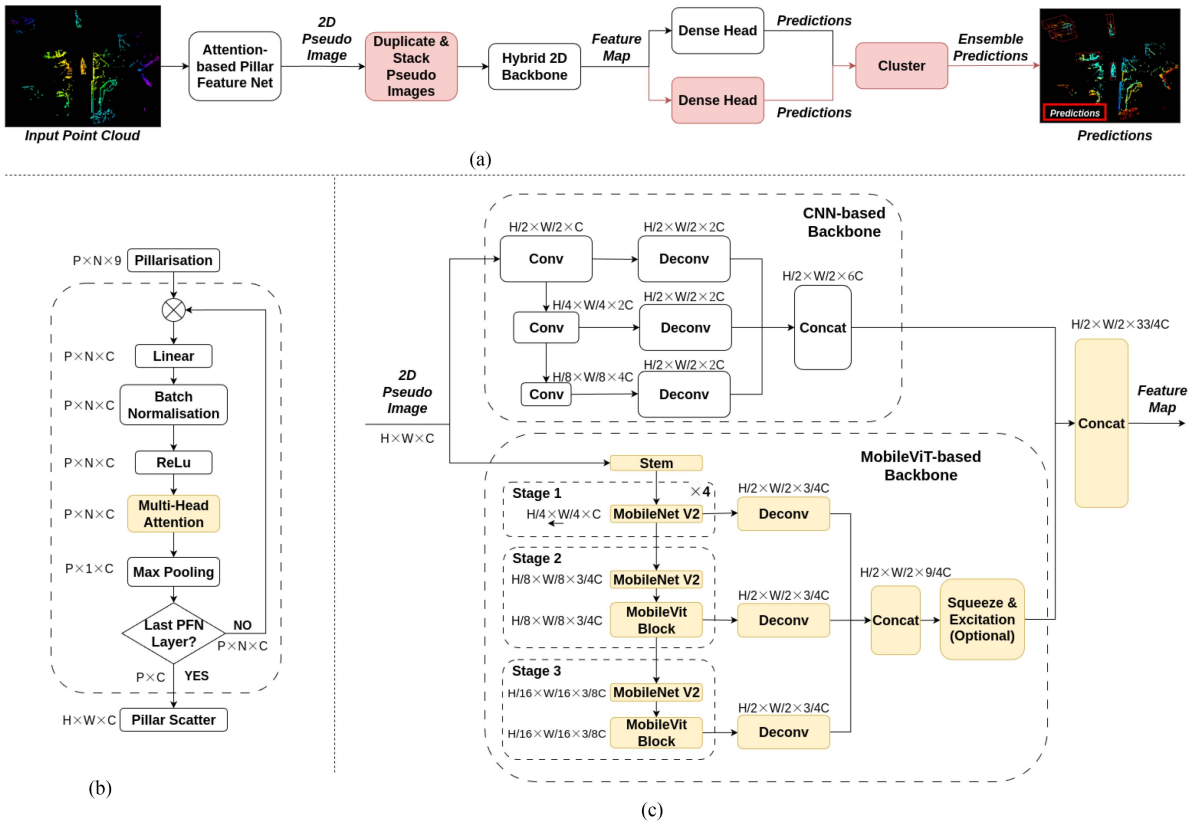


Fig. 2. (a) Overall architecture of U-MPCD. White blocks represent the modules for MPCD, and red blocks indicate the additional modules for MIMO integration. (b) Detailed design of the attention-based pillar feature net. (c) Detailed design of the hybrid 2-D backbone, which integrates the modified MobileViT with the original 2-D CNN backbone. The yellow blocks highlight the changes.

However, in maritime contexts, objects are often larger, vary significantly in size, and are located at longer distances, resulting in extremely sparse point cloud captures. Consequently, to improve the performance of these detectors in maritime environments, they must possess enhanced capability to extract fine-grained local details for smaller objects and robust global features for larger targets.

Following the principle of improving detection accuracy while maintaining efficiency, we adapt the PointPillar architecture [7]. Two main modifications are introduced to capture detailed local features from sparse LiDAR points while also leveraging global context in the 2-D backbone: 1) a multihead attention module within the pillar feature net (PFN) for enhanced local feature encoding, and 2) a Parallel transformer module integrated with the original 2-D CNN backbone to strengthen large-scale contextual reasoning. Fig. 2(a) illustrates the overall architecture of the proposed U-MPCD model. It comprises the MPCD network together with a multi-input and multi-output (MIMO) framework for uncertainty estimation, where the white blocks indicate the deterministic (MPCD) prediction networks and the red blocks are additional modules introduced by the MIMO framework to effectively infer predictive uncertainty. This section focuses on the design of the deterministic prediction network, while the integration of the MIMO-based method, built on MPCD, to achieve efficient uncertainty estimation is described in Section III-B.

The MPCD deterministic detection network comprises three main stages: the attention-based pillar feature net, a hybrid 2-D backbone, and a dense head. First, the raw point cloud data are processed by the pillar feature net, converting the 3D point cloud into a 2D Bird's BEV pseudoimage. This pseudoimage is then fed into the hybrid 2-D backbone to generate a feature map. Finally, the dense head produces the prediction results. Design details for each module are provided below.

1) Attention-Based Pillar Feature Net: The pillar feature net (PFN) is the primary innovation in the PointPillars framework and begins with a process termed pillarization, in which the 3-D point cloud is partitioned into pillars in the XY-plane. Each pillar spans the full Z-axis but occupies only one grid cell in the XY-plane, so all points whose (x, y) coordinates fall within that cell are grouped together. Once these pillars are formed, each point is augmented with both a mean offset (the difference between a point's coordinates and the mean of all points in that pillar) and a center offset (the distance between each point and the pillar's geometric center). These offsets are then concatenated with the original (x, y, z) coordinates and intensity. As a result, the channel dimension increases from the original three features to nine, enriching each point's representation by providing both local (relative) and global (pillar-level) spatial context. This enhancement enables the network to better interpret the spatial structure and relationships within each pillar. The resulting data is represented as a tensor of shape (P, N, C), where

P denotes the total number of pillars, N is the maximum number of points per pillar, and C captures both the original and offset features.

After the pillarization and feature augmentation steps, the pillar feature net (PFN) processes each pillar through multiple layers of linear transformations (linear, batch normalization, and rectified linear unit (ReLU)), followed by max pooling to extract a global pillar-level feature, as illustrated in Fig. 2(b). This is a recursive step, where in intermediate PFN layers, the global feature is concatenated with the original point features along the channel dimension. This concatenation allows each point to perceive both its local characteristics and the broader context of the pillar as a whole. However, relying solely on a single max pooling step can lead to the loss of fine-grained details. For example, when detecting a boat, features like the slight curve of its hull near the water or the tall, narrow shape of its mast can provide important clues about its orientation and identity. Because max pooling selects only the strongest activations across all points, these features may be overshadowed. By introducing a self-attention mechanism before pooling, the network allows each point to dynamically weigh critical local features (e.g., a curved edge or a distinctive reflection rate) and assign them higher importance. This attention mechanism ensures that even minor but meaningful attributes of the object are preserved. As a result, the network's capacity to recognize and classify objects with complex shapes and varying scales is enhanced, mitigating the limitations of max pooling.

Self-attention is a mechanism that enables each element in an input sequence (set of points within a pillar) to weigh the importance of all other elements, thereby capturing relationships across the entire input. The crucial step is to define three sets of learnable parameter matrices, $\mathbf{W}^Q \in \mathbb{R}^{d_{\text{in}} \times d_k}$, $\mathbf{W}^K \in \mathbb{R}^{d_{\text{in}} \times d_k}$, and $\mathbf{W}^V \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$, where d_{in}, d_k, and d_{out} are the dimensionalities of the input features, the query/key vectors, and the output features, respectively. These parameter matrices project the input $\mathbf{X}$ into queries $\mathbf{Q} = \mathbf{X}\mathbf{W}^Q$, keys $\mathbf{K} = \mathbf{X}\mathbf{W}^K$, and values $\mathbf{V} = \mathbf{X}\mathbf{W}^V$. Each element in $\mathbf{X}$ uses $\mathbf{Q}$ to determine how strongly it "attends" to the keys $\mathbf{K}$, and then aggregates information from $\mathbf{V}$ accordingly. Mathematically, self-attention is computed as [32]

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{1}$$

where $\mathbf{Q}\mathbf{K}^\top$ produces pairwise similarity scores (dot product) between the n elements of $\mathbf{X}$. Dividing by $\sqrt{d_k}$ keeps these values from growing too large, while the softmax distribution determines the degree to which each query element attends to each key. The weighted sum of $\mathbf{V}$ then aggregates the most relevant features from across the sequence.

When extended to multihead attention [32], this process is replicated across a number of different heads, N_h. Each head learns its own set of parameter matrices: $\mathbf{W}^{Q(i)}$, $\mathbf{W}^{K(i)}$, and $\mathbf{W}^{V(i)}$, which can be expressed as

$$\text{MultiHead}(\mathbf{X}) = \underset{i\in[N_h]}{\text{Concat}}\left[\text{Attention}\left(\mathbf{X}\mathbf{W}^{Q(i)}, \mathbf{X}\mathbf{W}^{K(i)}, \mathbf{X}\mathbf{W}^{V(i)}\right)\right]\mathbf{W}^O \tag{2}$$

where distinct parameter matrices $\mathbf{W}^{Q(i)}, \mathbf{W}^{K(i)} \in \mathbb{R}^{d_{\text{in}} \times d_k}$, and $\mathbf{W}^{V(i)} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$ are learned for each head $i \in N_h$. An additional parameter matrix $\mathbf{W}^O \in \mathbb{R}^{N_h\, d_{\text{out}} \times D_{\text{out}}}$ projects the concatenation of the N_h head outputs (each in $\mathbb{R}^{d_{\text{out}}}$) to the output space $\mathbb{R}^{D_{\text{out}}}$.

In a multihead setup, multiple parallel attention heads each compute separate sets of Q, K, and V, allowing the network to focus on various aspects of local structure within the same pillar. By generating multiple attention weight matrices, the network can simultaneously emphasis diverse features, thereby preserving subtle local details—such as slight hull deformations or specialized material reflections—and incorporating them into a robust pillar representation. The multihead attention-based PFN is shown in Fig. 2(b). We set the number of heads to four, and an ablation study in Section V supports this design choice.

2) Hybrid 2-D Backbone: In PointPillars, the CNN-based backbone performs multilevel feature extraction by applying downsampling to progressively capture high-level spatial features. It then combines this with feature fusion through upsampling and concatenating multiscale features, enabling the network to incorporate broader contextual information effectively [3]. Despite these improvements, the inherent locality of convolutional operations still limits the ability of CNNs to fully capture long-range dependencies or truly global context. For sparse point clouds, especially those involving large-scale targets, relying on local convolutional features often fails to establish effective links between distant pillars. For instance, when a network needs to integrate features from both the bow and stern of a ship to infer its overall contour (e.g., a long curved outline), conventional CNNs cannot adequately handle these distant dependencies.

The introduction of the ViT has made it feasible to capture global context more effectively. By leveraging multihead self-attention, ViT can bridge spatial distances and directly assess correlations between distant features. This characteristic is particularly advantageous for maritime scenarios, where point clouds are sparse and large ships span extensive areas [33]. Thus, the global modeling capability of ViT can significantly enhance detection or CLS. Motivated by this potential, we replace the original CNN backbone with ViT, thereby enabling a more comprehensive capture of global context.

In practice, we experimented with two ViTs: Swin ViT [34] and MobileViT [36]. However, Swin ViT proved overly complex for our relatively small-scale data set, as it strongly depends on large-scale data and pretrained models. Even when we reduced Swin ViT by retaining only a few Transformer blocks, the results remained suboptimal. Consequently, we selected MobileViT, a lightweight variant tailored to real-time scenarios. MobileViT was originally tailored for image CLS. Accordingly, we modified it by removing the final convolution layer, the global pooling layer, and the linear CLS layer. We also omitted the last MobileViT block, retaining only two blocks to better match our input–output requirements and reduce computational overhead. Although MobileViT surpassed Swin ViT in performance, it remained less than ideal, possibly due to its greater emphasis on global information at the expense of local detail. In this situation, critical local features in the point cloud, such as subtle edges or

textures, may not be adequately captured, ultimately affecting overall accuracy.

In response, we adopted a multiscale feature fusion approach inspired by "Fusion on Fusion" [51]. The main strategy uses an encoder–decoder feature pyramid structure to aggregate multiscale outputs by applying a uniform spatial transformation (for example, deconvolution) and then concatenating them. This design compensates for the detailed information lost when relying on a single path output. Nevertheless, even with MobileViT plus multiscale fusion, performance remained marginally below that of the original CNN backbone. One likely explanation is that MobileViT, which emphasizes a lightweight design and global context, produces lower resolution feature maps in each layer, making it harder to retain the spatial details required for local feature extraction, an area where CNNs have traditionally excelled.

Hence, we propose a hybrid approach that integrates modified MobileViT with the original CNN, aiming to benefit from both local feature extraction and global modeling. Our experiments confirm that this hybrid model delivers a notable performance improvement while only modestly raising computational overhead, which is much less than initially expected. Fig. 2(c) shows our final hybrid CNN and MobileViT backbone. We retain the existing CNN-based backbone and add a modified MobileViT-based backbone, treating MobileViT as an encoder whose outputs at each stage feed into a deconvolution-based decoder for spatial alignment and concatenation. Ultimately, the MobileViT outputs are merged with those of the CNN before being passed to the dense head. Detailed design choices, such as the number of output channels in each MobileViT stage, which intermediate layers to concatenate, and whether to incorporate additional modules (for example, a squeeze and excitation block, which recalibrates channel-wise feature responses by modeling interdependencies between channels), will be discussed in Sections V along with our experimental evaluations.

B. Bayesian Inference

BNNs [12] implement a probabilistic approach to model the network's weights, thereby producing a predictive distribution that reflects the network's uncertainty about the target y given an input $\mathbf{x}$ and a training data set $\mathcal{D}$. The key formulation is often expressed as [11]

$$p(y \mid \mathbf{x}, \mathcal{D}) = \int p(y \mid \mathbf{x}, \mathbf{W})\, p(\mathbf{W} \mid \mathcal{D})\, \mathrm{d}\mathbf{W} \tag{3}$$

where $p(y \mid \mathbf{x}, \mathbf{W})$ denotes the observation likelihood, and $p(\mathbf{W} \mid \mathcal{D})$ is the posterior distribution over weights derived from a training data set $\mathcal{D}$.

In conventional supervised learning, the network parameters $\mathbf{W}$ are treated as point estimates, resulting in deterministic predictions $\hat{y} = f(\mathbf{x}, \mathbf{W})$. In a BNN, however, one attempts to perform approximate inference on the posterior $p(\mathbf{W} \mid \mathcal{D})$, so that the resulting model outputs explicitly capture the network's perceived uncertainty at inference time. However, high-dimensional and multimodal tasks often make it computationally intractable to directly solve $p(\mathbf{W} \mid \mathcal{D})$. Practical applications often depend on approximate inference or sampling strategies to achieve a tractable approximation of the posterior distribution. In the following sections, we introduce three feasible methods, MC-dropout, deep ensemble, and MIMO for incorporating Bayesian inference into our deterministic MPCD networks. We employ MIMO to construct our final uncertainty-capturing model, termed U-MPCD, which achieves real-time performance. Meanwhile, MC-Dropout and deep ensemble serve as baselines to evaluate the uncertainty estimation performance of the MIMO model. At the end of this section, we discuss the evaluation metrics used to quantitatively assess performance.

1) Monte Carlo Dropout: MC-Dropout is a sampling-based approximate Bayesian inference method. In standard network training, dropout is typically employed as a regularizer only during the training phase. However, MC-Dropout maintains dropout during inference, causing the network to sample a different, partially zeroed-out set of weights on each forward pass and thereby generating multiple distinct network instances. Suppose the inference phase conducts T forward passes, and let $\mathbf{W}_t$ denote the network weights during the tth pass. A conceptual illustration of MC-Dropout is shown in Fig. 3(a). The model's predictive distribution can then be approximated by [13]

$$p(y \mid \mathbf{x}, \mathcal{D}) \approx \frac{1}{T} \sum_{t=1}^{T} p(y \mid \mathbf{x}, \mathbf{W}_t). \tag{4}$$

For CLS tasks, the predictive distribution is a probability mass function. In each forward pass, the network outputs a predicted softmax score vector $\hat{\mathbf{s}}_t \in \mathbb{R}^C$, where C is the number of classes. Consequently, the observation likelihood for a specific class c can be written as $p(y = c \mid \mathbf{x}, \mathbf{W}_t)$. Combining with (4), the predictive distribution of class c is approximated by averaging the predicted softmax scores across all forward passes, which can be expressed as

$$p(y = c \mid \mathbf{x}, \mathcal{D}) \approx \frac{1}{T} \sum_{t=1}^{T} \hat{s}_{c,t} \tag{5}$$

where $\hat{s}_{c,t}$ is the softmax probability of class c in the tth forward pass.

For regression (REG) tasks, which in the context of point cloud detection involve predicting the length, location, and orientation of a bounding box (represented as a vector of continuous values), the observation likelihood can be expressed as a probability density function under a Gaussian likelihood assumption [11], which can be expressed as

$$(\mathbf{y} \mid \mathbf{x}, \mathbf{W}_t) = \mathcal{N}\left(\mathbf{y} \mid f_{\mathbf{W}_t}(\mathbf{x}), \beta^{-1}\mathbf{I}\right) \tag{6}$$

where $f_{\mathbf{W}_t}(\mathbf{x})$ denotes the predicted mean under the tth dropout sample, while $\beta^{-1}\mathbf{I}$ represents the covariance matrix, with β^{-1} indicating the variance for each output dimension. Consequently, the predictive distribution can be determined by the sample mean and variance.

However, in standard deterministic networks, variance is not estimated. The model only produces a single best prediction, representing the predicted mean. Consequently, to capture variance, we introduce an additional output layer to predict it alongside the

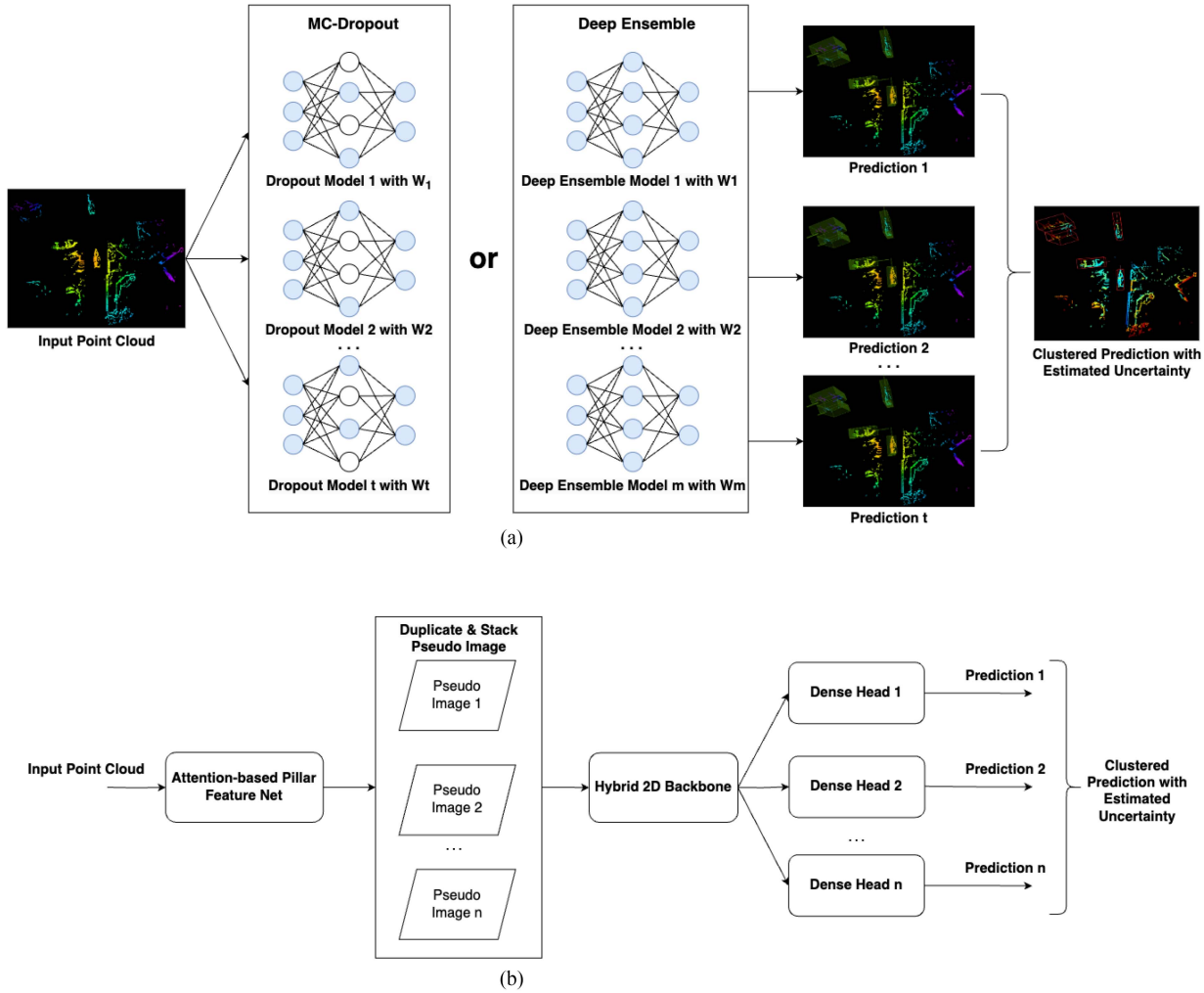


Fig. 3. (a) Conceptual illustration of MC dropout and deep ensemble, where activated neurons are shown in blue. (b) Conceptual illustration of the multi-input multi-output architecture workflow.

mean, adapting a direct modeling approach. The loss function must be designed to learn the variance, σ^2, for both linear REG (covering the length and centroid position of the bounding box) and angular REG (for heading estimation). Consequently, we adopt a variance loss function for linear REG [52], which can be expressed as

$$\mathcal{L}_{\text{var,linear}} = \frac{1}{2}\exp(-\lambda^T)\left\|\mathbf{y}_{\text{gt}} - f_{\mathbf{W}}(\mathbf{x})\right\| + \frac{1}{2}\lambda^T\mathbf{1} \quad (7)$$

where $\lambda = \log\sigma^2$ denotes the log variance for numerical stability, and y_{gt} as well as $f_W(x)$ represent the ground truth and predicted values, respectively. Unless otherwise stated, $\log(\cdot)$ denotes the natural logarithm (base e). The $(1/2)\lambda^T\mathbf{1}$ term penalizes the model if the training data exhibit high aleatoric uncertainty.

For angular variance REG, we employ the von Mises loss, whose mathematical form is presented as

$$\begin{aligned}\mathcal{L}_{\text{var},\theta} = {} & \log I_0\big(\exp(-\lambda)\big) - \exp(-\lambda)\cos\big(\theta - \theta_{\text{gt}}\big) \\ & + \lambda_V\,\text{ELU}\big(\lambda - \lambda_0\big) \end{aligned} \quad (8)$$

where λ_V is the regularization coefficient, and λ_0 controls the position of the ELU. I_0 is the zeroth-order modified Bessel function. θ and θ_{gt} represent the predicted and ground truth headings, respectively. For further details on the von Mises angular variance loss, please refer to [53].

2) Deep Ensemble: Deep ensemble [14] trains multiple networks with the same architecture, each starting from distinct initial parameters, whereas MC-Dropout draws multiple weight samples from a single network by enabling dropout. Suppose there are M such networks, each with weights $\{\mathbf{W}^{(m)}\}_{m=1}^{M}$. At inference time, given an input $\mathbf{x}$, one evaluates $p(y \mid \mathbf{x}, \mathbf{W}^{(m)})$ across the M independently trained networks and then aggregates their outputs to estimate uncertainty. Conceptually, this procedure approximates weight-space sampling by independently training multiple models, each potentially converging to a distinct local minimum and thus producing different predictions that indicate model uncertainty. In general, MC-Dropout and deep ensemble employ similar principles, since MC-Dropout can be viewed as an ensemble of networks. Their primary difference lies in the method used to sample weights. The conceptual architecture of deep ensemble is illustrated in Fig. 3(a).

3) Multi-Input Multi-Output: MIMO [15] obtains multiple predictive samples in a single forward pass by configuring multiple dense heads within one network, with each head receiving either a copy of the same input or a slightly perturbed version. The conceptual architecture of MIMO is shown in Fig. 3(b). To preserve efficiency, rather than directly feeding multiple raw point clouds, the additional inputs are generated after collapsing the raw point cloud data into a 2-D pseudoimage. During training, each head may process an individually pillarized and feature extracted representation, such as separate 2-D pseudoimages created from different frames for point cloud detection, to ensure enough diversity and enable the dense head to learn distinct representations. At inference, the approach can be simplified by extracting features only once from the raw point cloud input at the pillar level, creating a single 2-D pseudoimage that is then duplicated. Each duplicated pseudoimage is passed through the 2-D backbone to extract features that are forwarded to different dense heads. This setup allows the network to produce multiple outputs in a single pass, facilitating uncertainty estimation without repeated inferences or multiple networks. Since all heads share the same backbone and vary only in head-specific parameters, MIMO requires only a single backbone computation, with negligible additional cost from the multiple dense heads. By contrast, MC Dropout requires repeated forward passes through the backbone, even if dropout is applied only to the dense layers. As a result, MIMO achieves greater efficiency at test time while still producing diversified predictions comparable to those of MC-Dropout or deep ensembles. The architecture of the final U-MPCD model, integrating MIMO with MPCD, is shown in Fig. 2(a).

4) Uncertainty Evaluation: In our experiments, we employ different methods to quantify both epistemic and aleatoric uncertainty in CLS and REG tasks, respectively [54]. For CLS, we use Shannon Entropy (SE), which is a widely recognized measure of the uncertainty in a variable's possible outcomes. It effectively captures both epistemic and aleatoric uncertainty because the entropy is computed directly from the predictive distribution $p\big(y = c \mid \mathbf{x}, \mathbf{D}\big)$. The predictive distribution is obtained by averaging the predicted softmax scores across all forward passes, as demonstrated in (5). The SE [55] is defined as

$$\mathcal{H}(y \mid \mathbf{x}, \mathcal{D}) = -\sum_{c=1}^{C} p(y = c \mid \mathbf{x}, \mathcal{D}) \log p(y = c \mid \mathbf{x}, \mathcal{D}). \tag{9}$$

However, SE quantifies only the overall predictive uncertainty and does not distinguish between epistemic and aleatoric components. To address this limitation, we introduce Aleatoric Entropy (AE), sometimes referred to as conditional SE, to specifically capture aleatoric uncertainty arising from data noise. In a Bayesian context, AE is computed as the expectation of conditional SE over the model's weights. The expression for AE is given by

$$\mathrm{AE}(\mathbf{x}) = \mathbb{E}_{p(\mathbf{W}|\mathcal{D})}[\mathcal{H}(y \mid \mathbf{x}, W)] \tag{10}$$

where $\mathcal{H}(y \mid \mathbf{x}, W)$ represents the conditional SE of the class predictions y for a given input x and a specific model configuration defined by weights W.

Mutual information (MI) is also used to capture epistemic uncertainty in CLS. It measures the mutual dependence between two random variables by quantifying the information gained about one variable through observing the other. It can be expressed simply as the difference between SE and AE. Mathematically, we have [55]

$$\mathrm{MI}(\mathbf{x}) = \mathcal{H}\big(y \mid \mathbf{x}, \mathcal{D}\big) - \mathbb{E}_{p(\mathbf{W}|\mathcal{D})}\big[\mathcal{H}\big(y \mid \mathbf{x}, \mathbf{W}\big)\big]. \tag{11}$$

For the REG task, we use epistemic total variance (ETV) and Aleatoric total variance (ATV) to capture epistemic and aleatoric uncertainty, respectively [52]. The ETV quantifies model uncertainty by evaluating the variability of REG predictions across multiple forward passes or ensemble models. Specifically, the covariance matrix $C_{\mathrm{ETV}}(\mathbf{x})$ is constructed using the REG values from a bounding box cluster as [52]

$$C_{\mathrm{ETV}}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N}\big(\mathbf{y}_i - \bar{\mathbf{y}}\big)\big(\mathbf{y}_i - \bar{\mathbf{y}}\big)^{\top} \tag{12}$$

where N is the number of forward passes or ensemble models, $\mathbf{y}_i$ represents the REG prediction from the ith model, and $\bar{\mathbf{y}}$ is the mean of the predictions. The ETV is then calculated as the trace of the covariance matrix, $\mathrm{trace}\big(C_{\mathrm{ETV}}(\mathbf{x})\big)$.

To capture aleatoric uncertainty, the ATV is calculated using the predicted variances from the models for a bounding box cluster. The covariance matrix $C_{\mathrm{ATV}}(\mathbf{x})$ is created by averaging the predicted variance matrices across all forward passes, $(1/N)\sum_{i=1}^{N}\sigma_i^2$. Then, the ATV is obtained as the trace of the covariance matrix, $\mathrm{trace}\big(C_{\mathrm{ATV}}(\mathbf{x})\big)$.

Beyond quantifying uncertainty itself, we also employ several metrics to evaluate the quality of predictive distributions, which can be divided into two aspects: calibration and sharpness. Calibration evaluates whether the predicted probabilities are statistically consistent with the actual outcomes. For example, in a well-calibrated model, predictions with a 70% likelihood should correspond to the event occurring approximately 70% of the time. A well calibrated model therefore provides reliable probability estimates, which is crucial for safety-critical ASV operations.

Sharpness, on the other hand, measures how concentrated or narrow the predictive distribution is around the ground truth. A sharper distribution indicates that the model makes confident predictions, whereas a flatter distribution reflects greater uncertainty. Unlike calibration, sharpness is an intrinsic property of the predictive distribution and does not depend on the actual outcomes. Ideally, a model should achieve both good calibration (accurate probabilities) and high sharpness (confident predictions).

To evaluate the calibration of predictive uncertainty, we use average calibration error (ACE) for REG tasks and marginal calibration error (MCE) for CLS tasks [56]. ACE quantifies the average absolute error between the predicted scores and the empirical ground truth values across all equally divided score intervals, focusing specifically on the calibration quality for the ground truth class. In contrast, MCE extends ACE by assessing the calibration of the full predicted distribution across all possible classes.

In adition, we measure the sharpness of uncertainty in CLS by examining the softmax distribution for each prediction. Two proper scoring rules are employed for this purpose: negative log-likelihood (NLL) and the Brier Score (BS). NLL is a local and strictly proper scoring rule that evaluates the softmax distribution at the ground truth label. A lower NLL value implies a better fit for that specific ground truth label. Suppose we have N prediction samples and C classes, with $\hat{s}_{c,n}$ as the softmax probability for the cth class and $y_g t$ as the corresponding one-hot encoded ground truth label. The NLL for CLS is defined by [14]

$$\mathrm{NLL}_{\mathrm{CLS}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} y_{\mathrm{gt}}\,\log\left(\hat{s}_{c,n}\right). \tag{13}$$

The BS is a nonlocal and strictly proper scoring rule that evaluates the entire softmax distribution by taking the squared error between the predicted probability and the one-hot ground truth label. The BS is computed as [57]

$$\mathrm{BS} = \frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C}\left(\hat{s}_{c,n} - y_{\mathrm{gt}}\right)^2. \tag{14}$$

Finally, in REG tasks, we evaluate uncertainty sharpness by comparing the predicted means and variances for each prediction sample to the corresponding ground truth object. Two scoring rules are employed here: the NLL and the energy score (ES). The formulation for the REG NLL is presented as [53]

$$\mathrm{NLL}_{\mathrm{REG}} = \frac{1}{2N}\sum_{n=1}^{N}\left(\left(y_{\mathrm{gt}}-\mu_n\right)^{\top}\Sigma_n^{-1}\left(y_{\mathrm{gt}}-\mu_n\right)+\log\det\Sigma_n\right) \tag{15}$$

where μ and Σ represent the predictive mean and covariance matrix, respectively. The first term penalizes any discrepancy between the predicted mean and the ground truth, while the second term acts as a regularization mechanism to prevent the model from assigning unbounded variance.

The ES is a strictly proper, nonlocal metric for REG. Derived from the energy distance, it can be approximated using Monte Carlo sampling for multivariate Gaussians. Let N be the total number of objects in the test set and M the number of ensemble members. For each ground truth y_{gt}, we draw ith samples $y_{n,i}$ from $\mathcal{N}\left(\mu_n, \Sigma_n\right)$. The ES minimizes in a similar fashion to the NLL but also penalizes distributions with high entropy, promoting tighter, better-calibrated predictions [58]

$$\mathrm{ES} = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{M}\sum_{i=1}^{M}\left\|y_{n,i}-y_{\mathrm{gt}}\right\| - \frac{1}{2(M-1)}\sum_{i=1}^{M-1}\left\|y_{n,i}-y_{n,i+1}\right\|\right). \tag{16}$$

IV. Experiments

This section presents experimental evaluations of the proposed 3-D point cloud detector tailored for maritime environments. The experiments proceed in two stages. In the first stage, we examine the performance of the proposed deterministic detection networks, MPCD. Specifically, we evaluate the impact of the multihead attention based pillar feature net, the modified 2-D backbone (consisting of the original 2-D CNN and a modified MobileViT backbone), and the combination of these two components. In the second stage, we assess how the proposed MPCD architecture performs when integrated with three practical BNN methods for uncertainty estimation. Using MC-Dropout and deep ensemble as baselines, we then evaluate the reliability of the MIMO approach in capturing predictive uncertainties, leading to our final U-MPCD.

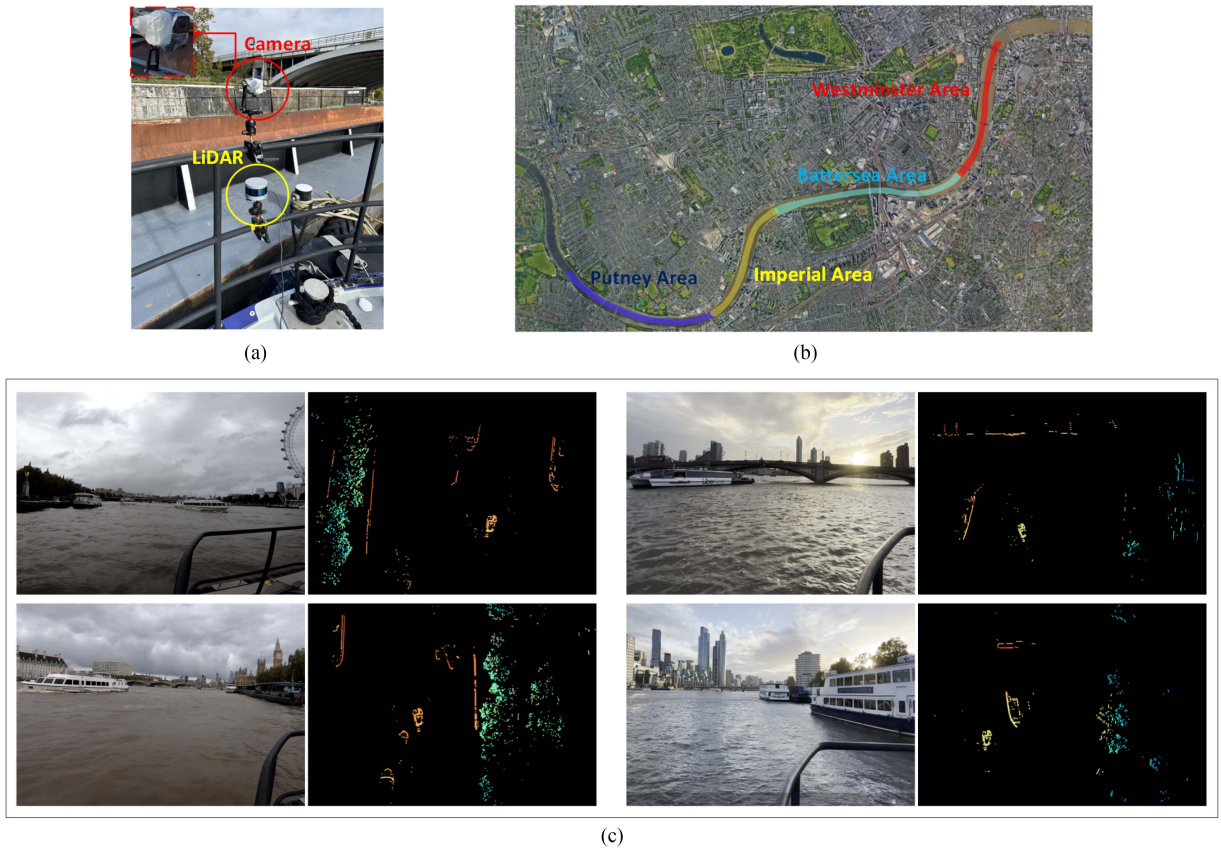
A. Experiment Setup

All experiments were conducted on a server equipped with an AMD EPYC 7T83 CPU and an NVIDIA RTX 4090 graphics processing unit (GPU), using the PyTorch environment for both training and evaluation.

1) Data Sets: We use the maritime LiDAR data set introduced in [2], which was originally obtained by placing a Velodyne 16-line LiDAR on a tripod along the River Thames and in nearby marinas. This data set classifies vessels into three length-based categories: small boat (length less than 7 m), medium boat (length ranging from 7 to 13 m), and large boat (length greater than 13 m). Because both the LiDAR and most boats remained stationary, either moored in marinas or near the riverbank, consecutive frames exhibit only minor differences, reducing data diversity and potentially causing overfitting. As a result, this limitation might restrict the evaluation of the actual performance of the proposed model. To address this limitation, we enhanced the original data set by installing the same Velodyne 16-line LiDAR on a survey boat and collecting additional data in busier areas of the River Thames, where many boats were in motion. These new data incorporate factors, such as target boat motion and a moving LiDAR coordinate frame, thereby introducing greater variation between consecutive frames. Fig. 4 depicts the onboard LiDAR setup and provides samples of data captured between Embankment Pier and Westminster Pier. Altogether, we recorded nearly 150 000 frames, each capturing reasonably clear point cloud features of the target boats.

However, labeling the entire data set would be highly labor-intensive. Our primary objective here is simply to augment the original data set with varied scenarios and additional large-boat samples, thereby broadening its diversity and allowing a more effective evaluation of our methods' capacity for target detection and uncertainty capture. Consequently, we labeled only 1000 frames from the new data and combined them with the original data set, yielding approximately 10,000 frames in total. Of these, 90% are used for training and 10% for evaluation. Following [15] for uncertainty evaluation, we further split the evaluation data in half, using the first half to recalibrate the predictive uncertainty, and the second half to measure the final performance of the calibrated models.

It should be noted that although the data set is sufficient for validating the proposed framework, it was collected in a limited area, which restricts the diversity of samples and scenarios. This limited diversity means that detection performance may vary when applied to different water bodies, sea states, or weather conditions. For example, wave dynamics, sensor noise



(a) (b)

(c)

Fig. 4. (a) LiDAR and camera setup on board. (b) Survey conducted for data set collection; four surveys were performed in the River Thames near Central London. (c) Example frames from the collected data set.

from rain or fog, background clutter in new environments, or the presence of previously unseen vessel types could reduce detection accuracy. Such issues are common to deep learning methods, where generalization improves with larger and more diverse training data but cannot be fully guaranteed. In such cases, the uncertainty estimation provided by our approach may serve as a useful indicator of reduced reliability. Future work will therefore focus on expanding the data set to include more varied environments, which will enable a more thorough evaluation of generalization.

2) Model Variants and Hyperparameter Setup: For evaluating the deterministic point cloud detector, the PointPillars framework is selected as the baseline model. The baseline is first modified by enabling an attention-based pillar feature net, referred to as PointPillar+ATT. A hybrid 2-D backbone is then introduced, which combines the modified MobileViT and the original CNN backbone, resulting in PointPillar+Hybrid Backbone. Finally, both modifications are merged into the baseline, producing the final proposed deterministic model, MPCD. To estimate predictive uncertainty from MPCD, three practical BNN approaches are examined: MC Dropout, deep ensemble, and MIMO. Accordingly, the derived models are MPCD+MC Dropout, MPCD+deep ensemble, and MPCD+MIMO. The final proposed uncertainty aware model is U-MPCD (MPCD+MIMO). The MC-Dropout and deep ensemble implementations serve as baselines to measure the effectiveness of the MIMO approach in capturing predictive uncertainty.

We set the LiDAR range to $\pm 70.4\,\mathrm{m}$ in the x-direction, $\pm 40\,\mathrm{m}$ in the y-direction, and $\pm 10\,\mathrm{m}$ in the z-direction. The corresponding pillar grid size is $(0.2\,\mathrm{m}, 0.2\,\mathrm{m}, 20\,\mathrm{m})$. Table II lists the predefined anchors used to initialize bounding boxes for matching with ground truth vessels during the training phase, where the specific dimensions are derived from statistical analysis of the entire data set for each class.

Except for U-MPCD (MPCD+MIMO), a batch size of 3 is used due to computational constraints. All other models employ a batch size of 6. Training is conducted for 120 epochs. For models that include the 2-D hybrid backbone (PointPillar+Hybrid Backbone, MPCD, and their Bayesian variations), we set the learning rate to 5×10^{-4}. Empirical observations indicated that Transformer-like components require a smaller learning rate than conventional CNN layers. Consequently, for purely CNN-based models, those without any Transformer elements, the learning rate is 3×10^{-3}.

When implementing MC-Dropout, dropout remains active during inference, and four forward passes are performed. The dropout rate is set to 0.4. In deep ensemble, four independent networks are trained by using different random seeds for frame shuffling. The reasoning behind these design choices is elaborated in the ablation study in Section V. With MIMO, two inputs and two outputs are configured within a single forward pass. In

TABLE II
PREDEFINED ANCHORS

Classes	Anchor Size (L, W, H) (m)	Anchor Rotations (rad)	Anchor Bottom Height (m)	Matched Threshold (%)	Unmatched Threshold (%)
Small boat	$(8.64, 2.67, 6.86)$	$[0, 1.57]$	-4.71	50	35
Medium boat	$(14.09, 4.54, 12.03)$	$[0, 1.57]$	-5.34	50	35
Large boat	$(27.25, 5.78, 10.87)$	$[0, 1.57]$	-6.49	60	35

Anchor rotation refers to the orientation adjustments of anchor boxes to better match possible object orientations. The matched and unmatched thresholds define the IoU criteria for assigning positive or negative labels to anchors.

TABLE III
DETECTION PERFORMANCE AND INFERENCE SPEED FOR EACH MODEL

Model	Average Precision (%)			mAP (%)	Inference Speed (Hz)
	small boat	medium boat	large boat		
Baseline (PointPillar)	63.63	66.83	64.03	64.83	**24.32**
PointPillar + ATT	50.22	72.28	53.42	58.64	22.47
PointPillar + Hybrid Backbone	66.24	61.91	55.44	61.20	19.96
MPCD	**66.30**	**78.27**	**74.72**	**73.10**	19.13 ↓

other words, during the testing phase, the 2-D pseudoimages generated after pillarization are duplicated once, and two dense heads are employed. This setup is primarily dictated by hardware constraints, which preclude the use of more inputs or outputs.

B. Experiment Results

1) MPCD — Deterministic Networks: In this section, the experimental results for deterministic model are presented. The original PointPillar architecture serves as the baseline. First, the effect of enabling an attention-based pillar feature net (PFN) within this baseline is investigated. Next, a hybrid 2-D backbone, combining the original CNN backbone and a modified MobileViT in parallel, is evaluated as a replacement for the baseline's 2-D backbone. Finally, the proposed model, MPCD, which integrates both the attention-based PFN and the hybrid 2-D backbone, is examined.

Table III presents the overall detection performance for each model in terms of mean average precision (mAP) and per-class average precision (AP) for small, medium, and large boats, along with the average inference time per frame. The proposed MPCD model achieves highest performance across all classes than the baseline, including notable improvements of 17.1% and 16.7% on medium and large boats, respectively. The overall mAP increases from 64.83% for the baseline to 73.1%, representing an improvement of approximately 12.8%. Although the average inference speed reduces from 24.32 to 19.14 HZ, this inference speed remains sufficient for real-time applications, given that a Velodyne VLP-16 typically operates in the range of 5–20 Hz.

It should be noted that PointPillar was selected as the main baseline due to its efficiency and suitability for real-time deployment. Other state-of-the-art detectors, such as SECOND and point-voxel region-based convolutional neural network (PV-RCNN), can achieve higher accuracy but at the cost of significantly greater computational complexity. In our previous work on the same data set [2], we found that SECOND and PV-RCNN outperformed PointPillar in terms of mAP, but their inference times were substantially slower, limiting their practicality for real-time ASV operations. By contrast, the proposed MPCD achieves a favorable balance: it improves accuracy over PointPillar and SECOND (68.7% mAP) while maintaining real-time inference speed, making it more suitable for maritime scenarios.

Although these findings show that MPCD outperforms the baseline, they do not clearly reveal the primary drivers behind this improvement. For instance, replacing the baseline 2-D backbone with a MobileViT-based hybrid backbone does not necessarily guarantee stronger global feature capture for medium and large boats, nor does adding an attention mechanism alone ensure better local feature extraction in pillars for small boat detection. Indeed, Table III indicates that modifying only the PFN with attention or only the 2-D backbone with the hybrid design actually reduces performance below the baseline. For PointPillar with the attention-based PFN, the overall performance drops to 58.64% ; although medium boat AP rises, small and large boat AP both decrease sharply. For PointPillar with the hybrid 2-D backbone, the overall performance falls to 61.2% ; small boat AP increases, but medium and large boat AP suffer significant declines. These results conflict with the initial expectation that attention in the PFN would yield finer grained local features (pillar-level), particularly for small boats that occupy fewer pillars, and that a MobileViT-based hybrid backbone would offer an expanded field of view and contextual information for capturing the overall shape of larger vessels.

A more detailed examination is needed to account for the behavior outlined above, rather than relying solely on each

TABLE IV
AVERAGE PRECISION FOR EACH CLASS ACROSS VARIOUS DISTANCE INTERVALS IN EACH DETERMINISTIC MODEL

Class	Distance Range (m)	Sample Counts	Average Precision (%)			
			PointPillar	PP+ATT	PP+Hybrid Backbone	MPCD
small boat	Dist ≤ 10	0	0	0	0	0
	$10 <$ Dist ≤ 20	12	32.66	21.95	**62.52**	41.46
	$20 <$ Dist ≤ 30	22	31.18	37.06	**55.83**	29.73
	Dist > 30	249	71.82	58.86	68.71	**72.59**
medium boat	Dist ≤ 10	98	73.94	76.19	50.06	**89.21**
	$10 <$ Dist ≤ 20	707	79.99	81.40	68.35	**86.97**
	$20 <$ Dist ≤ 30	605	78.04	78.48	61.61	**86.79**
	Dist > 30	1568	56.47	62.65	60.27	**70.30**
large boat	Dist ≤ 10	0	0	0	0	0
	$10 <$ Dist ≤ 20	57	84.09	42.80	37.73	**92.11**
	$20 <$ Dist ≤ 30	144	63.25	60.46	58.80	**70.58**
	Dist > 30	208	59.71	55.30	64.66	**73.93**

The bold values indicate the models with the best performance. Specifically, the bold value represents the highest detection accuracy for that class of objects.

category's average precision for evaluation. Since point clouds become increasingly sparse at longer distances, this sparsity significantly affects detection performance. Therefore, detection performance is broken down by distance range for each class, as shown in Table IV. The sample counts indicate the number of targets within each distance interval in the evaluation set, and the corresponding distribution in the training set is generally comparable.

From Table IV, it can be observed that when multihead attention is added to the PFN in PointPillar, the detection performance for small boat unexpectedly declines at closer distance ranges ($10\,\text{m} < \text{Dist} \leq 20\,\text{m}$). A possible explanation is that small boats at close distances have rich point cloud within each pillar but occupy fewer pillars overall. This increases the sensitivity of the attention mechanism to noise and minor variations in the point distribution within pillar-level features. Furthermore, the limited number of small boat samples at this range in the training data set worsens the problem. At mid-range distances ($20\,\text{m} < \text{Dist} \leq 30\,\text{m}$), performance improves modestly, likely because there is a better balance between the number of points in each pillar and the number of pillars spans, enabling the model to learn stable feature representations. In other words, the attention-based PFN effectively extracts pillar-level features, and the moderate number of points in each pillar reduces the sensitivity of the attention mechanism to noise and minor variations. In addition, since small boats span fewer pillars at this range, extensive cross-pillar correlation is not required, making the model's capacity sufficient for accurate detection. However, at longer ranges ($\text{Dist} > 30\,\text{m}$), performance drops again as the point clouds for small boats become excessively sparse, making it difficult for the attention module to robustly extract key features. Meanwhile, large boats display performance declines across all distance intervals. Although attention may refine local features at the pillar level, if it focuses too heavily on localized features without effectively aggregating information across the entire extent of the boats, it may fail to represent the boat's overall shape cohesively, resulting in fragmented detection or underestimation of its true boundaries. The performance for medium boats at all distance intervals exceeds that of the baseline. A likely reason is that medium boats are more abundantly represented in the training samples, enabling the attention module to learn optimal representations for targets of moderate scale. In addition, medium boats distribute a balanced number of points across pillars—more points per pillar than small boats but fewer than large boats—and span a moderate number of pillars. This balance enables the attention module to effectively capture contextual information within each pillar, resulting in improved performance.

In short, multihead attention in the pillar feature net improves detection performance only when the target object has a moderate number of pillar spans and a balanced number of points in each pillar; otherwise, it may negatively impact detection performance. If an object has many points per pillar but spans fewer pillars, the attention mechanism becomes more sensitive to noise and minor variations in point distribution, which can degrade performance. Conversely, if there are too few points in each pillar, the sparsity of point clouds, especially at longer distances, makes feature extraction by the attention module less effective. In addition, if an object spans too many pillars, the attention mechanism may focus excessively on local features without aggregating cross-pillar information, leading to fragmented detection or underestimation of the object's true boundaries. Therefore, to achieve an overall performance improvement with attention-based PFN, it is essential to incorporate additional global feature extraction capabilities to effectively link pillar-level features.

When PointPillar is equipped with a hybrid backbone composed of both original CNN backbone and modified MobileViT,

the detection performance for medium and large boats at longer distances (Dist $>$ 30 m) shows improvement. In these far-range scenarios, the point cloud becomes increasingly sparse and noisy, making convolutional operations vulnerable to overlooking scattered key points. By integrating MobileViT's global self-attention mechanism into the BEV feature map, distant regions of interest can be interconnected (merge these dispersed pillar features), enabling the network to capture the overall outline of distant targets more effectively. However, performance for medium and large boats at shorter distances (Dist $\leq$ 30 m) decreases. Although the hybrid backbone emphasizes global information, its integration with the original CNN layers may lead to conflict with the local feature extraction of the 2-D CNN. In other words, the network might fail to achieve the optimal balance between global modeling and local feature extraction because the fine local features extracted by the CNN are weakened by the influence of MobileViT. Therefore, to improve overall performance when incorporating the hybrid backbone, enhancing the network's ability to effectively extract fine local details is required.

Interestingly, small boat detection improves in the interval of (10 m $<$ Dist $\leq$ 30 m). This outcome likely stems from the fact that, at this range, the point cloud feature captures for small boats remain sparse relative to medium or large vessels. Instead of detailed local features, broader global characteristics appear more decisive here. Consequently, incorporating the MobileViT into the backbone enhances detection in a similar manner to how it benefits medium and large boats beyond 30 m.

Nevertheless, based on the analysis above, the detection of small boats remains particularly challenging at both close and far distances. At closer ranges, small boats often occupy only a few pillars despite having many points per pillar, which makes the attention mechanism overly sensitive to noise and local variations. At longer ranges, by contrast, point clouds become excessively sparse, limiting the model's ability to extract reliable features. These factors, combined with the relatively small number of training samples for small boats, contribute to their reduced detection performance. Several remediation strategies may help mitigate these issues. First, augmenting the training data set with additional small-boat samples across different distance intervals could improve generalization and address data imbalance. Second, leveraging temporal information from sequential LiDAR frames may stabilize predictions for targets with very few pillar spans. Third, multimodal fusion (e.g., incorporating camera data) could complement LiDAR sparsity by providing richer appearance cues. Finally, higher resolution LiDAR sensors or adaptive pillarization strategies may enhance point density for small objects, thereby reducing sensitivity to both noise and sparsity. While these directions lie beyond the scope of the present study, they represent promising avenues for future work to further improve small-boat detection.

In summary, PointPillar integrated only with the attention-based PFN fails to achieve consistent improvements. This outcome likely arises because the local attention mechanism, although helpful in specific scenarios, lacks broader contextual insights and can be hindered by sparse or large targets. Similarly, employing only the hybrid backbone improves global feature modeling but does not adequately preserve detailed local features for particular classes or distance ranges. Once the attention-based PFN and the hybrid backbone are integrated, however, local geometric cues in each pillar are refined and reinforced by more effective global feature fusion in the 2-D backbone. This joint arrangement compensates for each module's individual weaknesses, explaining why the combined approach outperforms either single-module extension of the baseline.

2) U-MPCD — Probabilistic Networks: In this section, the proposed MPCD network is combined with three practical Bayesian methods—MC-Dropout, deep ensemble, and MIMO—to quantify predictive uncertainty. For each method, we evaluate the prediction accuracy, uncertainty quality, and inference speed. The uncertainty quality is assessed based on two key aspects: sharpness (the concentration of the predicted probability distribution) and calibration (the alignment between predicted probabilities and actual outcomes). Table V presents the detailed evaluation results, where the MPCD baseline does not explicitly output uncertainty, leaving all six uncertainty quality-related metrics listed as NA.

From the perspective of mAP, deep ensemble achieves the highest detection accuracy at 86.57%, significantly higher than the baseline's 73.1%. MIMO follows at 74.58%, still outperforming the baseline, whereas MC-Dropout attains only 71.4%, which is below baseline performance. This variance arises from how each method manages weights and produces different sets of predictions. MC-Dropout discards a portion of network weights at inference (with a dropout rate of 0.4 here), intending to approximate the predictive distribution via multiple random samples. However, with a relatively high dropout rate of 0.4, chosen to ensure sufficient stochasticity in the predictive distribution, randomly zeroing out "critical" weights can result in larger performance fluctuations for the entire network, which partly explains the weaker mAP. This limitation is further compounded by the inherent reliance of MC-Dropout on random weight masking rather than maintaining full model capacity as in deep ensembles. Deep ensemble, on the other hand, combines the outputs of four fully trained networks (identical architecture but different initializations and random shuffles). Each individual network retains its complete capacity and converges to a distinct local minimum during training, so integrating their predictions merges multiple perspectives, resulting in excellent accuracy and robust uncertainty estimates. MIMO duplicates the pseudoimage within the same network and assigns two independently learned dense heads, merging their predictions to obtain multiple outputs from a single forward pass. Although this does not match the comprehensive weight diversity of a "multinetwork aggregation" as in deep ensemble, it still considerably improves mAP and able to provides predictive distribution.

Further evaluation explores predictive uncertainty in terms of sharpness. For CLS, the NLL and BS are employed, while for REG, NLL and the ES are used. Deep ensemble attains the lowest values among all models for both CLS and REG, indicating a more concentrated predictive distribution that places higher probabilities near the true target, resulting in strong confidence regarding detected objects. Although MIMO and MC-Dropout do not achieve the same level of sharpness as

TABLE V
EVALUATION OF DETECTION ACCURACY AND PREDICTIVE UNCERTAINTY FOR EACH PRACTICAL BNN IMPLEMENTATION APPLIED TO THE MPCD, WHERE BS DENOTES THE BS, ES IS THE ES, AND MCE AND ACE REFER TO MARGINAL AND ACE, RESPECTIVELY

Model	mAP (%)	Sharpness				Calibration		Inference Speed (Hz)
		CLS		REG		CLS	REG	
		NLL	BS	NLL	ES	MCE	ACE	
Baseline (MPCD)	73.10	NA	NA	NA	NA	NA	NA	19.13
MC-Dropout	71.45	0.058	0.019	−5.989	0.236	0.053	**0.056**	4.72
Deep Ensemble	**86.57**	**0.040**	**0.014**	**−7.410**	**0.177**	**0.021**	0.062	4.55
MIMO (U-MPCD)	74.58	0.057	0.022	−6.402	0.239	0.036	0.104	**15.03**

TABLE VI
DETECTION ACCURACY AND UNCERTAINTY METRICS ACROSS DISTANCE RANGES FOR EACH BOAT CLASS

Class	Distance Range (m)	AP (%)	Samples Count	CLS			REG	
				SE	AE	MI	ETV	ATV
small boat	$\text{Dist} \leq 10$	0.00	0	NA	NA	NA	NA	NA
	$10 < \text{Dist} \leq 20$	37.90	12	0.2702	0.2699	0.0003	0.0010	0.0821
	$20 < \text{Dist} \leq 30$	47.28	22	**0.0686**	**0.0686**	**0.0000**	**0.0004**	0.0493
	$\text{Dist} > 30$	**75.13**	249	0.1591	0.1588	0.0003	0.0005	**0.0158**
medium boat	$\text{Dist} \leq 10$	**91.80**	98	**0.1014**	**0.1014**	**0.0000**	**0.0002**	**0.0128**
	$10 < \text{Dist} \leq 20$	83.54	707	0.1154	0.1154	0.0000	0.0007	0.0307
	$20 < \text{Dist} \leq 30$	83.67	605	0.1116	0.1115	0.0001	0.0014	0.0208
	$\text{Dist} > 30$	74.97	1568	0.1225	0.1224	0.0001	0.0263	0.0326
large boat	$\text{Dist} \leq 10$	0.00	0	NA	NA	NA	NA	NA
	$10 < \text{Dist} \leq 20$	**88.51**	57	**0.1012**	**0.1011**	**0.0001**	**0.0075**	**0.1030**
	$20 < \text{Dist} \leq 30$	77.91	144	0.1096	0.1094	0.0002	0.0088	0.1563
	$\text{Dist} > 30$	68.55	208	0.1176	0.1174	0.0001	0.0139	0.2015

For CLS, the total predictive uncertainty is assessed using SE, with AE and MI capturing aleatoric and epistemic uncertainty, respectively. For REG, ETV and ATV are performed.

deep ensemble, their predictive outputs remain adequately concentrated. MIMO's predictive distribution, although it scores slightly higher than deep ensemble, remains sufficiently concentrated to inspire a reasonable degree of confidence in real-world scenarios. By contrast, MC-Dropout exhibits higher CLS and REG NLL values relative to MIMO, implying that randomly discarding weights during inference increases distribution variability and undermines predictive consistency. Turning to calibration metrics, the MCE assesses how closely CLS probabilities align with actual occurrences, and the ACE gauges how well REG outputs match ground truth. Deep ensemble achieves the lowest MCE and relatively low ACE values, reflecting superior calibration of its predictive distribution for both CLS and REG tasks. MC-Dropout, in contrast, records the lowest ACE but the highest MCE, implying a slight mismatch between CLS probabilities and ground truth relative to deep ensemble, yet better alignment in REG outputs. This could be attributed to the regularization capacity of dropout. MIMO obtains an MCE of 0.0364 which, while not as low as deep ensemble, still outperforms MC-Dropout in CLS calibration. However, MIMO's REG calibration appears weaker, possibly because the shared PFN and backbone modules restrict the model's capacity to capture diverse perspectives on uncertainty, leading to less reliable variance estimates.

Regarding inference speed, the baseline model achieves 19.13 Hz, whereas deep ensemble and MC-Dropout, which both require four forward passes, operate at only about 4.55 and 4.72 HZ, respectively. Although these methods impose a high computational cost, they deliver the most robust performance and uncertainty estimates. MIMO, on the other hand, produces multiple outputs in a single forward pass, achieves 15.03 Hz, indicating far greater efficiency than deep ensemble or MC-Dropout. Considering the earlier findings, although MIMO does not match deep ensemble in detection accuracy and uncertainty quality, it still provides an effective balance among accuracy, reliable uncertainty, and near-real-time speed, making it a practical choice for real-world applications that require both efficient inference and robust uncertainty estimates.

Table VI presents the detection accuracy (AP), sample sizes, and multiple uncertainty indicators (SE, AE, MI, ETV, and ATV) for the U-MPCD (MPCD+MIMO) model across three boat categories (small, medium, and large) at various distance

intervals. The results show that across all classes and distance intervals, aleatoric uncertainty (measured by AE and ATV) contributes the most to the total uncertainty for both CLS and REG tasks, compared to epistemic uncertainty (measured by MI and ETV). This indicates that observational noise are the primary drivers of uncertainty in this case, while the model's knowledge (epistemic uncertainty) plays a minimal role. Furthermore, as distance increases and point clouds become sparser, the CLS task remains relatively unaffected in terms of epistemic uncertainty, as MI values stay consistently low across all distances and classes. This suggests that the model is sufficiently trained and confident in its ability to classify objects, even under challenging sparse point cloud conditions. However, for REG tasks, the increasing sparsity of the point cloud has a more significant impact, as reflected in the steady rise of ATV with distance for all classes. Interestingly, there is an outlier for small boats at ($10\,\text{m} < \text{Dist} \leq 20\,\text{m}$), where ETV is 0.0010 higher than at greater distances. This is likely due to the very small sample size for training in these conditions, which prevents the model from learning properly.

Overall, the experimental results highlight that the proposed U-MPCD effectively estimates predictive uncertainties, capturing both epistemic and aleatoric components in real-time while improving detection accuracy compared to the baseline MPCD. These advancements enhance the safety and reliability of ASV operations.

3) Visualized Results: Fig. 5 presents sample visualizations of detection results from the baseline model (PointPillar), the proposed deterministic network (MPCD), and the uncertainty-aware variant (U-MPCD) in BEV. White bounding boxes indicate deterministic predictions, while U-MPCD additionally provides color-coded uncertainty-aware boxes. In scenario 1, PointPillar's predicted bounding box is too narrow to reflect the target boat's actual width, appears overly long, and is shifted to the right rather than centered on the boat. By contrast, MPCD's bounding box prediction more accurately represents the boat's size and position. Consequently, in U-MPCD, the predictive uncertainty for bounding box position, size, and heading remains small, as indicated by the green bounding box closely overlapping the white one, and the correspondingly low quantified uncertainty values. In scenario 2, PointPillar mistakenly treats one large ship as two separate boats. MPCD addresses this issue more effectively thanks to its stronger global context modeling. However, since the bow of the vessel is not captured, predicting its true length is challenging. As shown by the U-MPCD results, the variance in the boat's length is higher than its width, indicating relatively greater uncertainty in length estimation.

In scenario 3, the model faces difficulties owing to limited point cloud information for the target boat. As shown in the visualized results for both PointPillar and MPCD, discrepancies arise in the heading and size of the predicted bounding boxes. Consequently, the predictive uncertainty remains high. In the U-MPCD output, the variance of the bounding box size and heading is notably large, resulting in a significant gap between the red and white bounding boxes. In scenario 4, the challenging boat highlighted in the scene has point cloud features consisting only of a mast, which results in high predictive uncertainty. Beyond this extreme case, boats located closer to the LiDAR generally benefit from richer point cloud feature and therefore display lower uncertainty, as demonstrated by the green box closely matching the white one. In contrast, predictive uncertainty grows with distance, where sparser point cloud features translate into a wider gap between the green and white bounding boxes. In addition, a false detection appears in the PointPillar results but not in MPCD's. The detailed uncertainty patterns of these cases can be further examined in the 3-D visualizations provided in Fig. 6. In scenario 5, background point cloud is incorrectly identified as a boat, resulting in extremely high predictive uncertainty for this erroneous detection.

V. Ablation Study

A. Multihead Attention

To identify the optimal number of attention heads in the pillar feature extraction stage, experiments were conducted with 1, 2, 3, and 4 heads, as shown in Table VII. With a single head, the mAP is lowest at 36.20%, and the AP values for all classes are also significantly small. This result likely arises from a single head focusing exclusively on the most critical features in each pillar and overlooking others, leading to an imbalance in feature representation. Increasing the number of heads to two raises the mAP to 42.72%, with modest improvements in each class, reflecting a more balanced approach to feature extraction.

With three heads, the mAP climbs further to 51.21%, driven by a substantial increase in the AP of medium and large boats, although the AP for small boats drops to 40.70%. This suggests that the extra attention heads may prioritize features beneficial for medium and large boats, possibly due to the sample distribution in the training set. Finally, using four heads yields the highest mAP of 58.64%, with notable improvements for medium and large boats, demonstrating the capacity of multihead attention to comprehensively capture features across a variety of object classes. However, inference speed decreases slightly as more attention heads are introduced. Considering the potential performance benefits and the associated increase in computational cost, configurations with more than four heads were not explored. Overall, a configuration with four heads is selected, providing the best balance between detection accuracy and computational efficiency.

B. Hybrid Backbone

This section provides an ablation study of various 2-D backbone designs as shown in Table VIII, keeping a four-head attention PFN fixed. Initially, Swin ViT was adopted as the 2-D backbone, but its detection accuracy drop significantly below that of the baseline PointPillar. A possible reason is that Swin ViT requires larger data sets and carefully tuned hyperparameters to function effectively. The limited data set in this study prevents full optimization of the Swin ViT parameters.

Subsequently, a modified lightweight MobileViT was used as the 2-D backbone, showing better overall performance than Swin ViT and suggesting that a lighter model is more appropriate for

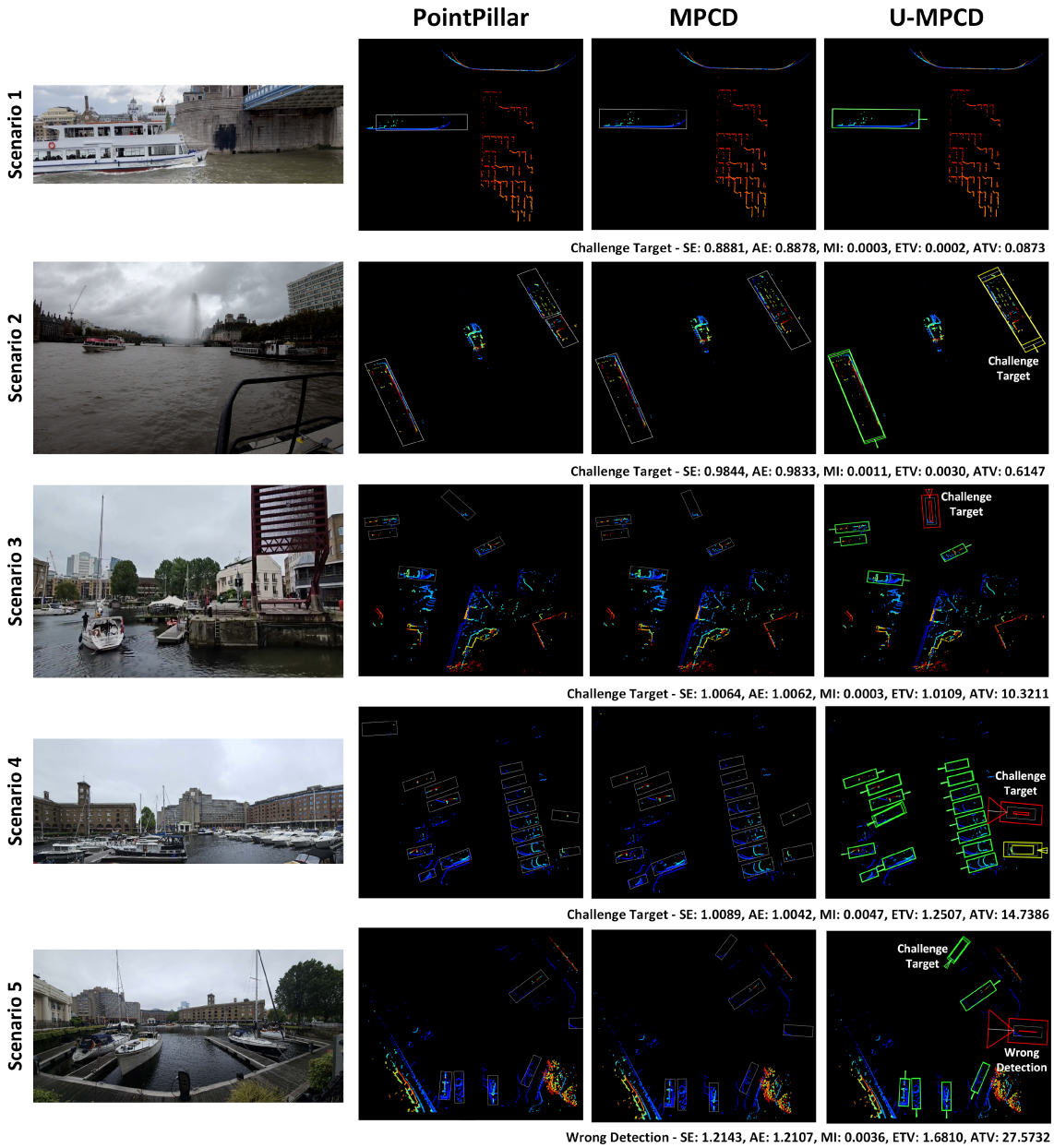


Fig. 5. Visualization results of detection across five challenging real-world scenarios for PointPillar (baseline), MPCD, and U-MPCD in BEV. The first column shows the RGB image, while the following columns present LiDAR-based detections for each model. For PointPillar and MPCD, bounding boxes are shown in white to indicate deterministic predictions. For U-MPCD, both white bounding boxes (deterministic predictions) and additional uncertainty-aware bounding boxes are shown, with colors representing predictive uncertainty (green=low, yellow=moderate, red=high). The quantitative predictive uncertainty for the most challenging object in each scenario is also reported.

TABLE VII
COMPARISON OF VARIOUS MULTIHEAD ATTENTION CONFIGURATIONS IN THE PILLAR FEATURE NET

Number of Heads	Average Precision (%)			mAP (%)	Inference Speed (Hz)
	small boat	medium boat	large boat		
1	44.98	41.01	22.61	36.20	23.68
2	**56.07**	46.77	25.31	42.72	**23.73**
3	40.70	65.30	47.64	51.21	22.92
4	50.22	**72.28**	**53.42**	**58.64**	22.47

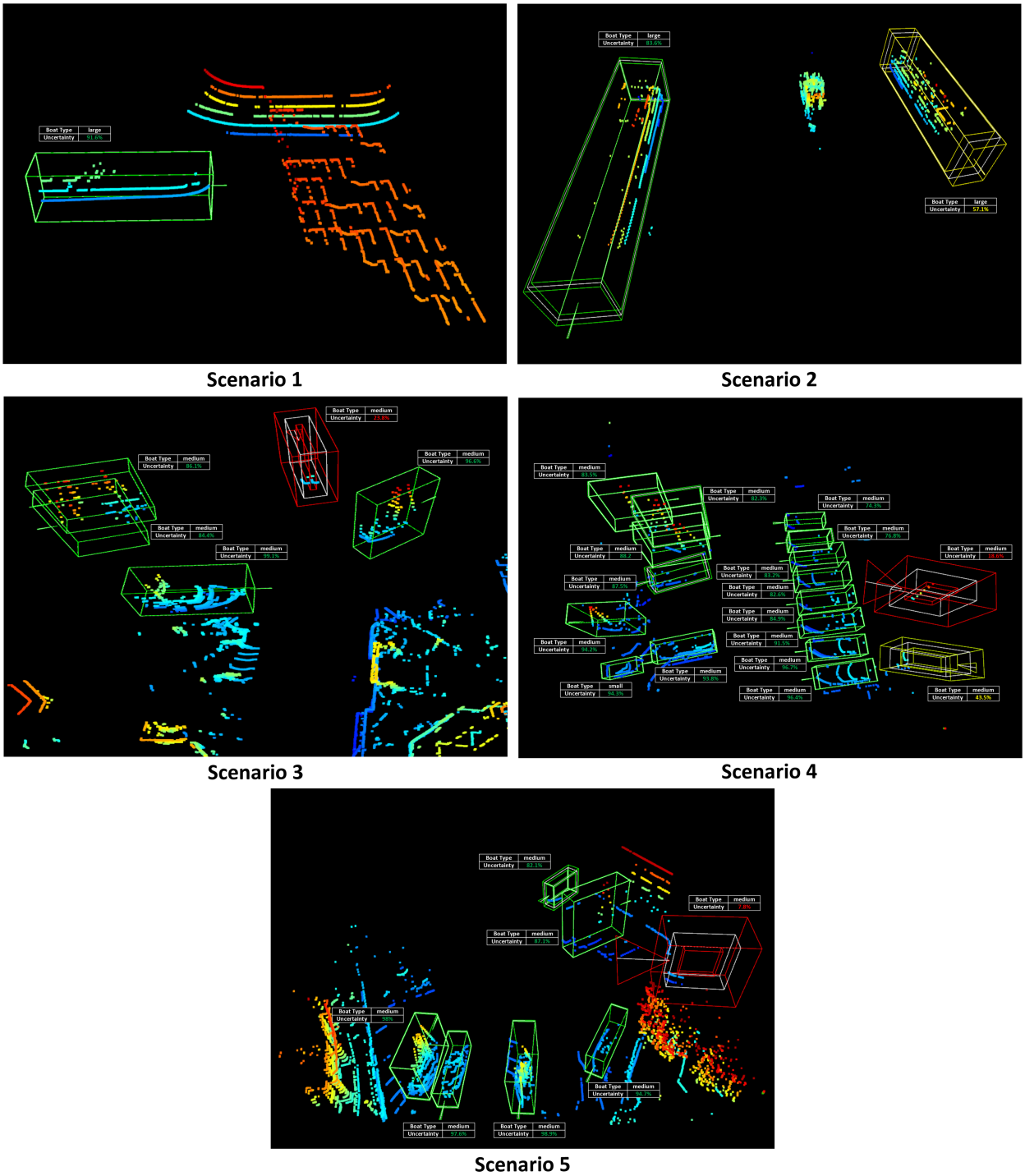


Fig. 6. Additional qualitative visualization results focusing exclusively on U-MPCD outputs in 3-D view. Both predicted classes and confidence scores are shown, along with uncertainty-aware bounding boxes (green=low, yellow=moderate, red=high). Compared to the BEV visualizations, these 3-D results emphasize how uncertainty estimates complement CLS and REG outputs, offering more interpretable and spatially informative indicators of detection reliability in diverse maritime scenarios.

the small size data set. However, MobileViT did not substantially improve detection across all classes. For instance, medium and large boat AP values increase significantly, but small boat AP remained low, implying a limited capacity to capture detailed local features. To address this limitation, a multiscale fusion strategy was introduced in MobileViT, extracting feature maps from different scales, reshaping the output of each scale to 128 channels, and then concatenating them to enhance local feature representation. Although these changes led to improved small-boat detection accuracy, the expected overall performance was not achieved, likely due to MobileViT's inherent constraints in extracting local features.

TABLE VIII
COMPARISON OF VARIOUS 2-D BACKBONE CONFIGURATIONS

2D Backbone	Average Precision (%)			mAP (%)	Inference Speed (Hz)
	small boat	medium boat	large boat		
Swin ViT	24.89	45.32	38.26	36.16	7.37
MobileViT	37.76	60.45	58.28	52.16	**24.54**
Multi-scale MobileViT	55.85	61.37	58.06	58.43	23.25
Hybrid (8 channels)	17.45	32.14	31.51	27.04	18.72
MPCD (48 channels)	66.30	**78.27**	**74.72**	73.10	19.13
MPCD + Squeeze & Excitation	**69.83**	77.55	73.07	**73.48**	18.96

Therefore, to leverage the CNN backbone's ability to capture fine local features, the multiscale MobileViT is integrated with the original CNN backbone to form a hybrid design, aiming to learn both local and global features more effectively. Because two substantial modules were merged, the initial design choice was to reshape each scale's feature output in MobileViT to only eight channels to reduce computational cost. However, experimental results showed that overly compressing features during upsampling led to an increased proportion of noise and a final mAP of just 27.04. Ultimately, assigning 48 output channels per MobileViT stage created a more balanced tradeoff between local and global feature extraction, producing a 73.1% mAP and only marginally lower inference time. This configuration represents the final MPCD model. In addition, incorporating a squeeze-and-excitation module further improved accuracy for certain classes but incurred a slight increase in inference time.

VI. CONCLUSION AND LEARNED LESSONS

In this article, we first introduced the MPCD, specifically designed for ASVs operating in maritime environments. The proposed model addresses the real-time demands of maritime object detection using LiDAR-captured point clouds. The proposed model is built upon PointPillar for its fast inference speed and notable detection performance. Maritime environments present unique challenges, such as large variance in target object size, requiring a detection model capable of effectively capturing both local features (e.g., fine-grained geometric details of a boat's edges or corners) and global features (e.g., the overall structure or spatial layout of a boat in the scene).

To address these challenges, we proposed two modules within the proposed MPCD model: the attention-based point feature net (PFN) and the hybrid 2-D backbone. The attention-based PFN employs a multihead attention mechanism to correlate points within each pillar, enabling refined pillar-level feature representation, which constitutes local features. These features are then scattered back to form a 2-D pseudoimage. The hybrid 2-D backbone combines the original 2-D CNN backbone from PointPillar with a modified MobileViT in parallel. By adopting a multiscale encoder–decoder structure, this design leverages the CNN's capacity for local feature extraction while utilizing MobileViT for effective global feature fusion, ensuring robust learning of both fine details and broader spatial relationships. Experimental results demonstrate that integrating the attention-based PFN and the hybrid backbone improves the capacity for capturing pillar-level local features through the attention mechanism and enhances the ability to extract global features using the hybrid backbone. This synergistic combination achieves a 12.8% increase in detection accuracy compared to the baseline model.

To further enhance the reliability of the proposed deterministic model, we integrated it with the multi-input multi-output (MIMO) approach, a practical implementation of BNNs. This integration enables the model to estimate predictive uncertainty, resulting in the U-MPCD. Experimental results demonstrate that U-MPCD effectively captures both epistemic and aleatoric uncertainty while slightly improving detection accuracy by 2% compared to the deterministic MPCD. With an inference speed of 15.03 Hz, U-MPCD aligns well with the Velodyne VLP-16 LiDAR update frequency of range from 5 to 20 Hz, making it suitable for real-time ASV operations. These advancements significantly improve the safety and reliability of ASV deployments.

The lessons learned through this research can be broadly categorized into two aspects: data collection trials and detection model development. Regarding data collection, the key insights are as follows.

1) *LiDAR Performance and Point Cloud Sparsity:* The onboard Velodyne 16-line LiDAR, specified to have a range of up to 100 m, exhibits significant point cloud sparsity in real-world maritime data collection. Once a target boat exceeds 30 m, the captured features become extremely sparse, often reducing large ships to a single line of points and making object detection challenging. This limitation is even more critical for small boats, whose returns may disappear almost entirely at longer distances, resulting in reduced detection accuracy. Unlike ground vehicles, which are typically closer to LiDAR sensors, boats in maritime environments are farther away, further

exposing the limitations of low-line-count LiDAR in terms of both operational range and vertical resolution. With only 16 channels, the wide vertical spacing between beams leads to poor coverage of large vessels beyond 20–30 m, with long ships (e.g., 40 m in length) often appearing as just a thin line of points. These factors collectively highlight the importance of considering LiDAR specifications when deploying perception systems in maritime environments and point to the need for sensors with higher resolution and extended range to support reliable detection.

2) *LiDAR Occlusion and Sensor Placement:* The LiDAR setup on a boat is more likely to experience self-occlusion due to the vessel's large structure. In contrast, LiDAR on ground vehicles can be easily mounted on the roof, providing an unobstructed 360° field of view with minimal self-occlusion. To mitigate this issue, multiple LiDAR sensors can be strategically positioned at different locations on the boat to achieve comprehensive 360° coverage while minimizing occlusions caused by the vessel itself. This setup requires careful calibration between sensors to ensure accurate data fusion. Alternatively, a single LiDAR sensor can be used but should be optimized to focus specifically on the area of interest.

Regarding detection model development, the lessons learned are as follows.

1) *Training Sample Imbalance:* The model's performance is constrained by the relatively small and imbalanced data set used for training. Specifically, certain object categories, such as small boats, may be underrepresented, leading to biased learning and reduced detection accuracy in those cases. Future work should focus on expanding and diversifying the training data set to better evaluate the model's generalization capabilities. In addition, to reduce the need for extensive manual labeling, exploring unsupervised or semi-supervised learning methods could be beneficial, enabling the model to learn from unlabeled or partially labeled data while improving scalability and adaptability.
2) *Balancing Local and Global Feature Extraction:* Achieving effective boat detection in maritime point clouds requires a careful balance between local and global feature extraction. Multihead attention in the PFN enhances pillar-level feature extraction but struggles when objects span too few or too many pillars, making it overly sensitive to noise or failing to capture complete object structures. Meanwhile, the hybrid backbone improves global feature aggregation, particularly for long-range targets, but may suppress fine-grained local details at shorter distances due to conflicts with CNN-based feature extraction. Our experiments show that using only one of these modules can degrade overall performance, while their integration refines local geometric cues and strengthens global spatial relationships, leading to a well-balanced, high-performance detection model. This finding underscores the necessity of jointly leveraging both local and global information to overcome the challenges of maritime perception.
3) *Challenges in Detection Accuracy Under Sparse Point Clouds:* The model's REG accuracy decreases at longer distances due to the increased sparsity of point cloud data, which reduces the geometric information available for bounding box estimation. This limitation is particularly pronounced for small or distant boats, where the limited number of reflected points makes accurate detection especially challenging. Despite improvements in feature extraction, the absence of sufficient raw data remains a fundamental constraint. To address this, future work could investigate advanced feature enhancement strategies, such as multiframe temporal fusion, point cloud upsampling, or multi-modal fusion with complementary sensors (e.g., cameras or radar). On the hardware side, higher-resolution or multi-LiDAR configurations could help alleviate sparsity. In controlled experiments, adding reflective panels or markers to small vessels may also increase the density of captured points. Collectively, these approaches represent promising directions for improving both small-object detection and long-range performance in sparse maritime point clouds.
4) *Deployment on Constrained Computing Platforms:* All experiments in this study were conducted using a high-end GPU (RTX 4090), which may not reflect the hardware constraints of embedded systems or edge devices typically available on ASVs. Although our model achieves real-time performance relative to LiDAR update rates in this setup, deployment on embedded platforms will likely require additional optimization. Potential approaches include model compression techniques (e.g., pruning, quantization, or knowledge distillation), efficient backbone architectures tailored for edge devices, and hardware acceleration using FPGAs or dedicated artificial intelligence (AI) processors. Future work will evaluate the tradeoffs between detection accuracy, uncertainty estimation, and inference speed when deploying the proposed method on embedded systems in real maritime environments.

   While this study focuses on maritime environments, the proposed framework has the potential to be extended to other domains, such as autonomous driving with autonomous ground vehicles (AGVs) and aerial perception with autonomous aerial vehicles (AAVs). In road-based scenarios, where objects, such as cars and trucks, are typically closer to the sensor and exhibit less point cloud sparsity, the framework could be used to investigate how uncertainty estimation contributes to safety-critical decision-making, especially for small or distant targets. Similarly, in aerial robotics, UAVs often encounter sparse point clouds due to altitude and sensor limitations, presenting challenges analogous to maritime perception. Evaluating our method in such cross-domain settings would not only highlight its adaptability but also help to systematically identify its limitations. These studies represent a promising avenue for future research and could further demonstrate the importance of tailoring perception frameworks to the unique sensing and operational constraints of each domain.

REFERENCES

[1] X. Cao, S. Gao, L. Chen, and Y. Wang, "Ship recognition method combined with image segmentation and deep learning feature extraction in video surveillance," *Multimed. Tools Appl.*, vol. 79, no. 13–14, pp. 9177–9192, Apr. 2020.

[2] Y. Xie, C. Nanlal, and Y. Liu, "Reliable LiDAR-based ship detection and tracking for autonomous surface vehicles in busy maritime environments," *Ocean Eng.*, vol. 312, no. 3, Nov. 2024, Art. no. 119288.

[3] D. Fernandes et al., "Point-cloud based 3D object detection and classification methods for self-driving applications: A survey and taxonomy," *Inf. Fusion*, vol. 68, pp. 161–191, Apr. 2021.

[4] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 77–85.

[5] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, Oct. 2018, Art. no. 3337.

[6] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4490–4499.

[7] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12689–12697.

[8] R. Khanam, M. Hussain, R. Hill, and P. Allen, "A comprehensive review of convolutional neural networks for defect detection in industrial applications," *IEEE Access*, vol. 12, pp. 94250–94295, 2024.

[9] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 16239–16248.

[10] C. He, R. Li, S. Li, and L. Zhang, "Voxel set transformer: A set-to-set approach to 3d object detection from point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8407–8417.

[11] D. Feng, A. Harakeh, S. L. Waslander, and K. Dietmayer, "A review and comparative study on probabilistic object detection in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 9961–9980, Aug. 2022.

[12] D. J. C. MacKay, "A practical bayesian framework for backpropagation networks," *Neural Comput.*, vol. 4, no. 3, pp. 448–472, May 1992.

[13] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.

[14] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6405–6416.

[15] M. Pitropov, C. Huang, V. Abdelzad, K. Czarnecki, and S. Waslander, "LiDAR-MIMO: Efficient uncertainty estimation for LiDAR-based 3D object detection," in *Proc. IEEE Intell. Veh. Symp.*, 2022, pp. 813–820.

[16] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 770–779.

[17] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: Point-based 3D single stage object detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11037–11045.

[18] W. Niu et al., "RT3D: Achieving real-time execution of 3D convolutional neural networks on mobile devices," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 9179–9187.

[19] B. Yang, M. Liang, and R. Urtasun, "HDNET: Exploiting HD maps for 3d object detection," in *Proc. Conf. Robot Learn.*, 2018, pp. 146–155.

[20] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7652–7660.

[21] H. Kuang, B. Wang, J. An, M. Zhang, and Z. Zhang, "Voxel-FPN: Multi-scale voxel feature aggregation for 3d object detection from LiDAR point clouds," *Sensors*, vol. 20, no. 3, Jan. 2020, Art. no. 704.

[22] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2647–2664, Aug. 2021.

[23] S. Shi et al., "PV-RCNN: Point-voxel feature set abstraction for 3d object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10526–10535.

[24] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "STD: Sparse-to-dense 3d object detector for point cloud," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1951–1960.

[25] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet : Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5105–5114.

[26] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Oct. 2019.

[27] Q. Meng, J. Tong, S. Yang, T. Xie, and X. Jin, "A enhanced feature extraction for 3d object detection," in *Proc. IEEE 25th China Conf. Syst. Simul. Technol. Appl.*, 2024, pp. 685–689.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–14.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[30] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.

[31] B. Graham, "Sparse 3D convolutional neural networks," in *Proc. Brit. Mach. Vis. Conf.*, 2015, pp. 150.1–150.9.

[32] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[33] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[34] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2999–10002.

[35] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10347–10357.

[36] S. Mehta and M. Rastegari, "Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer," in *Proc. Int. Conf. Learn. Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=vh-0sUt8HlG

[37] J. Zeng, D. Wang, and P. Chen, "A survey on transformers for point cloud processing: An updated overview," *IEEE Access*, vol. 10, pp. 86510–86527, 2022.

[38] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point transformer v2: Grouped vector attention and partition-based pooling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 33330–33342.

[39] X. Wu et al., "Point transformer v3: Simpler, faster, stronger," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 4840–4851.

[40] J. Mao et al., "Voxel transformer for 3d object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3144–3153.

[41] H. Wang et al., "DSVT: Dynamic sparse voxel transformer with rotated sets," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 13520–13529.

[42] C. Zhang, H. Wan, X. Shen, and Z. Wu, "PVT: Point-voxel transformer for point cloud learning," *Int. J. Intell. Syst.*, vol. 37, no. 12, pp. 11985–12008, Sep. 2022.

[43] A. Ando, S. Gidaris, A. Bursuc, G. Puy, A. Boulch, and R. Marlet, "Rangevit: Towards vision transformers for 3d semantic segmentation in autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 5240–5250.

[44] A. Clark, M. Bronckers, and Y. Wang, "Weight uncertainty in neural networks," in *Proc. Int. Conf. Mach. Learn*, 2015, pp. 1613–1622.

[45] T. Chen, E. Fox, and C. Guestrin, "Stochastic gradient hamiltonian Monte Carlo," in *Proc. Int. Conf. Mach. Learn*, 2014, pp. 1683–1691.

[46] X. Chen, J. D. Lee, X. T. Tong, and Y. Zhang, "Statistical inference for model parameters in stochastic gradient descent," *Ann. Stat.*, vol. 48, no. 1, pp. 251–273, Feb. 2020.

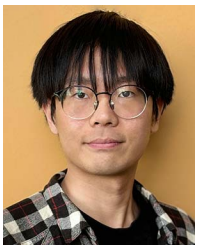
[47] H. Ritter, A. Botev, and D. Barber, "A scalable laplace approximation for neural networks," in *Proc. Int. Conf. Learn. Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=Skdvd2xAZ

[48] W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson, "A simple baseline for Bayesian uncertainty in deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 13153–13164.

[49] M. Havasi et al., "Training independent subnetworks for robust prediction," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=OGg9XnKxFAH

[50] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Aug. 2013.

[51] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.
[52] D. Feng, L. Rosenbaum, and K. Dietmayer, "Towards safe autonomous driving: Capture uncertainty in the deep neural network for LiDAR 3D vehicle detection," in *Proc. Int. Conf. Intell. Transp. Syst.*, 2018, pp. 3266–3273.
[53] Y. Zhong, M. Zhu, and H. Peng, "Uncertainty-aware voxel based 3d object detection and tracking with von-mises loss," 2020. [Online]. Available: https://doi.org/10.48550/arXiv.2011.02553
[54] Y. Ovadia et al., "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 14003–14014.
[55] J. Gawlikowski et al., "A survey of uncertainty in deep neural networks," *Artif. Intell. Rev.*, vol. 56, no. Suppl 1, pp. 1513–1589, Jul. 2023.
[56] A. Kumar, P. S. Liang, and T. Ma, "Verified uncertainty calibration," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 3792–3803.
[57] D. Feng, "Uncertainty estimation for object detection using deep learning approaches," Ph.D. dissertation, Inst. Meas., Control Microtechnology, Ulm Univ., Ulm, Germany, Dec. 2021. [Online]. Available: http://dx.doi.org/10.18725/OPARU-40486
[58] W. Liu, X. Wang, J. Owens, and Y. Li, "Energy-based out-of-distribution detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 21464–21475.



Yongchang Xie received the M.Sc. degree in marine engineering in 2020 from University College London, London, U.K., where he is currently working toward the Ph.D. degree in robotics with the Department of Mechanical Engineering.

His main research interests pertaining to the autonomous system are in perception and state estimation.

Peng Wu (Member, IEEE) received the Ph.D. in marine engineering from University College London (UCL), London, U.K., in 2020, focusing on decarbonising coastal shipping using fuel cells and batteries.

He is a Lecturer in Propulsion Systems Design/Integration with UCL. He was a Research Fellow with UCL (2020–2021), developing intelligent anomaly detection for marine autonomous systems. His current research interests include advanced power and propulsion systems for both road and marine applications, focusing on integrating low-carbon and zero-carbon power sources and developing related intelligent algorithms.

Brendan Englot (Senior Member, IEEE) received the S.B., S.M., and Ph.D. degrees in mechanical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2007, 2009, and 2012, respectively.

From 2012 to 2014, he was a Research Scientist with United Technologies Research Center, East Hartford, CT, USA. He is currently a Professor with the Department of Mechanical Engineering, Stevens Institute of Technology, Hoboken, NJ, USA, and the Director of the Stevens Institute for Artificial Intelligence. His research interests include motion planning, localization, and mapping for mobile robots, learning-aided autonomous navigation, and marine robotics.

Dr. Englot was the recipient of the 2017 National Science Foundation CAREER award and the 2020 Office of Naval Research Young Investigator Award.

Cassandra Nanlal received the Ph.D. degree in surveying and land information from the University of the West Indies, St. Augustine, Trinidad and Tobago, in 2019.

She is a Lecturer in marine geospatial science with the Department of Civil, Environmental, and Geomatic Engineering, UCL, London, U.K. Her areas of research interests include vertical separation models for land and marine, tides, coastal habitat mapping using innovative and automated survey and data processing techniques, hydrographic surveying standards and qc, real time reliable water quality data for decision makers, coastline change detection, sea level change, and carbon cost of data.

Yuanchang Liu (Member, IEEE) received the M.Sc. degree in power systems engineering and the Ph.D. degree in marine control engineering from University College London, London, U.K., in 2011 and 2016, respectively.

He was a Research Fellow in robotic vision and autonomous vehicles with the Surrey Space Centre, University of Surrey, Guildford, U.K. He is currently an Associate Professor with the Department of Mechanical Engineering, University College London. He is supervising a group of Ph.D. students with projects funded by UKRI, EU, and Royal Society. His research interests include automation and autonomy, with a special interest in the exploration of technologies for sensing and perception, and guidance and control of intelligent and autonomous vehicles.