

Towards Accurate and Efficient 3D Object Detection for Autonomous Driving: A Mixture of Experts Computing System on Edge

Linshen Liu^{1,*} Boyan Su^{1,*} Junyue Jiang¹ Guanlin Wu¹

Cong Guo² Ceyu Xu³ Hao Frank Yang^{1,†}

¹Johns Hopkins University ²Duke University ³HKUST

{lliu148, bsu11}@jh.edu, haofrankyang@jhu.edu

Abstract

This paper presents *Edge-based Mixture of Experts (MoE) Collaborative Computing (EMC2[‡])*, an optimal computing system designed for autonomous vehicles (AVs) that simultaneously achieves low-latency and high-accuracy 3D object detection. Unlike existing works, the EMC2 introduces a novel scenario-aware MoE architecture optimized for fusing complementary sparse 3D point clouds and dense 2D images to achieve robust multimodal representations for detection. Furthermore, EMC2 integrates an adaptive multimodal data bridge with multi-scale region proposing and scenario-aware routing, dynamically dispatching features to complementary experts based on object visibility and distance. In addition, EMC2 integrates joint hardware-software optimizations, including hardware resource utilization optimization and computational graph simplification, to ensure efficient and real-time inference on resource-constrained edge devices. Experiments on open-source benchmarks clearly show the EMC2 advancements as an end-to-end system. On the KITTI dataset, it achieves an average accuracy improvement of 3.58% and a 159.06% inference speedup compared to 15 baseline methods on Jetson platforms, with similar performance gains on the nuScenes dataset, highlighting its capability to advance reliable, real-time 3D object detection tasks for AVs.

1. Introduction

Traffic safety is a fundamental concern for both human drivers and Autonomous Driving Systems (ADS). Within an ADS, the perception module plays a pivotal role by serving as the "eyes" of the autonomous vehicle (AV), sensing the surrounding environment and delivering critical information to downstream modules such as motion planning, control, and safety guarantees. Existing conceptual safety of an AV perception system is generally defined by two core

*Equal contribution, †Corresponding author.

[‡]EMC2 Code: <https://github.com/LinshenLiu622/EMC2>

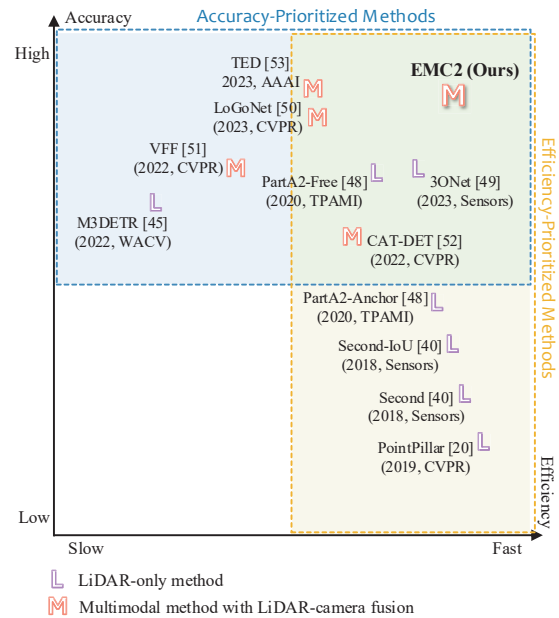


Figure 1. Comparison of 3D object detection models in accuracy and inference efficiency on the KITTI and nuScenes dataset. Most existing methods improve accuracy at the expense of inference efficiency. The proposed EMC2 achieves win-win performance by customizing MoE for autonomous driving.

dimensions: accuracy and efficiency. Accuracy ensures reliable object detection and tracking capabilities, enabling a credible understanding of the environment. Efficiency, characterized by low perception delay, is equally essential, as real-time high-level planning and control rely on fast information processing. Perception delay, defined as the sum of data acquisition, computation, and algorithm processing time, directly affects an ADS's responsiveness to dynamic environments. Research indicates that an ADS must process sensory inputs and issue feedback control commands within 100 milliseconds to guarantee prompt and safe vehicle operations [1]. **Detection precision and perception delay together define a critical safety boundary: a precise but sluggish system cannot react in time, while a fast but inaccurate system may lead to unsafe decisions.**

As illustrated in Fig. 1, achieving both high accuracy and low latency remains a fundamental challenge. Existing approaches often face inherent trade-offs between these objectives due to limited computational resources, insufficient adaptability of perception systems, and the inherent complexity of dynamic driving scenarios. Urban traffic environments require low-latency and accurate sensing capabilities to handle complex, densely interactive surroundings, while highway scenarios demand reliable detection of distant objects and prompt responses to unexpected events. Moreover, even for the same object, recognition priorities may vary dynamically within a scene depending on distance and scene complexity, posing notable challenges for single-modality models in balancing efficiency and accuracy [2]. Multimodal models, such as large-scale CNN and Transformer-based architectures that integrate LiDAR, radar, and camera data [3–5], leverage the complementary advantages of diverse sensors to enhance perception accuracy. Nevertheless, these approaches [6–9] often impose substantial hardware overhead and exhibit limited adaptability, particularly when deployed on resource-constrained edge platforms [10]. Even with the assistance of low-power machine learning compilation frameworks [11], such systems may still execute redundant computations under straightforward scenarios, introducing additional delays that undermine timely safety-critical decision-making.

Nevertheless, multimodal approaches provide a natural foundation for addressing the accuracy challenges faced by ADS. Rather than relying on a single integrated model designed to operate uniformly across diverse driving scenarios, the MoE framework [12] dynamically activates specialized sub-models, or experts, based on input characteristics and prior knowledge. MoE has demonstrated considerable effectiveness in large-scale applications, such as Large Language Models (LLMs) for natural language processing [13, 14], where it routes different tokens to appropriate experts, reducing redundant computations without sacrificing efficiency [15]. As model complexity grows, MoE enables models to scale their capacity while keeping inference hardware costs manageable. Deploying MoE in ADS, however, poses notable challenges [16], including strict real-time constraints, the need for multimodal data processing, and seamless end-to-end autonomous feedback control. Successful deployment requires efficient data pre-processing, adaptive modality selection, and robust expert routing mechanisms.

By leveraging diverse sensor data and a scenario-adaptive MoE dispatcher, we dynamically assign expert sub-models based on scenario data and prior statistical knowledge, providing an efficient, energy-efficient, and safe solution for ADS. To this end, we introduce an **Edge-based MoE Collaborative Computing (EMC2)** framework – an algorithm–hardware–software co-design solution that not

only improves accuracy through complex multimodal experts but also effectively reduces latency by utilizing an adaptive MoE dispatcher and lightweight single-modal experts. Our main contributions are summarized as follows:

1) A Multimodal 3D Object Detection Solution with MoE. The proposed EMC2 system achieves both high detection accuracy and inference efficiency, outperforming 15 existing one-size-fits-all baseline models. Designed for real-world ADS applications, EMC2 is optimized for deployment on resource-constrained platforms, making it more suitable for edge computing environments than generic alternatives.

2) Robust and Efficient Multimodal Perception through Expert Collaboration and System-Level Optimization. The proposed EMC2 framework incorporates scenario-specific experts and employs an adaptive expert selection strategy to meet heterogeneous perception demands. To mitigate the training instability commonly observed in MoE-based systems, we design a hierarchical optimization scheme that combines expert-level and global EMC2 back-propagation for effective collaborative learning. Furthermore, a distribution-aware resampling technique enhances robustness under real-world data imbalance. In addition to these designs, a tailored multiscale pooling module improves the efficiency of LiDAR–camera fusion, ensuring both efficient and accurate multimodal perception.

3) System-Level Optimization for Edge Inference. The original PyTorch-based MoE inference framework [17] suffers from inefficient management of memory and computational resources, leading to high latency on edge devices. To overcome these challenges, we propose an algorithm–hardware–software co-optimization strategy within EMC2, which incorporates efficient memory management techniques to improve L1/L2 cache hit rates and applies computational graph fusion to eliminate redundant data loading across layers. These combined optimizations notably improve inference efficiency.

4) State-of-the-art (SOTA) Accuracy and Efficiency Combined. On the KITTI dataset, EMC2 achieves 3.28% higher accuracy for pedestrians and 6.03% for bicycles under hard-level detection challenge compared to 15 baseline models. On the nuScenes dataset, it yields a 3.9% increase in mAP and a 1.8% boost in NDS. In addition, EMC2 achieves a 159.06% reduction in inference latency on the Jetson platform for KITTI, supporting its suitability for real-time edge deployment.

2. Related Work

LiDAR-only Method. The active 3D sensing of LiDAR makes it dominant in ADS 3D detection [18–20]. LiDAR-based methods process point clouds via voxelization [21], pillarization [20], or direct raw input [22]. Anchor-based heads, used for region proposal [21], and

center-based representations[23] are widely adopted. These methods achieve high accuracy on KITTI, as demonstrated by 3DSSD [24], PartA2 [19], and Voxel R-CNN [3]. However, weak LiDAR reflections for distant or occluded objects degrade detection performance [25, 26].

Multimodal Fusion Method. Combining LiDAR and camera RGB input enhances detection performance, especially for sparse LiDAR objects [27]. Fusion approaches include projection-based methods, which align image features with point clouds in 3D space [28], and model-based techniques that incorporate cross-attention [29] or unified feature spaces [30]. Recent multimodal models, such as PPF-Det [31] and RoboFusion [32], achieve top accuracy on KITTI benchmarks. However, these models often trade off accuracy for speed or vice versa.

Edge-based Compilation. ML compilation platforms are essential for improving the inference efficiency of deep learning models across diverse hardware architectures [33, 34]. TensorRT [35] is an NVIDIA inference optimizer that accelerates ML models on GPUs through layer fusion, precision calibration, and kernel auto-tuning. XLA [36] is a domain-specific compiler for TensorFlow enhancing efficiency by fusing operations and optimizing computation graphs. Open Neural Network Exchange (ONNX) Runtime [37] provides an open standard for model interoperability, enabling efficient execution across CPUs, GPUs, and accelerators. TVM [11] is an open-source deep learning compiler that automates model tuning and tensor scheduling for efficient deployment across CPUs and GPUs.

3. EMC2 Computing Methodology

We introduce EMC2, a multimodal MoE framework that minimizes latency and ensures reliable accuracy. Fig. 2 shows the EMC2 overview, EMC2 consists of: 1) *Adaptive Multimodal Data Bridge (AMDB)* to preprocess multimodal input (Sec. 3.1); 2) *Scenario-Adaptive Dispatcher* to adaptively dispatch a suitable expert for each scenario (Sec. 3.2); and 3) Three *Scenario-Optimized Experts* to decode AMDB-output features into final results (Sec. 3.3). We also design an effective training strategy to address the long-tail effect in the training process (Sec. 3.4). In addition, we deploy EMC2 on edge devices with self-designed 3D sparse convolution and Multiscale Pooling.

3.1. Adaptive Multimodal Data Bridge

Given the multimodal nature of EMC2’s sensor inputs, a module providing suitable preprocessed data for different experts is required. Thus, we propose *AMDB*, a preprocessing unit to efficiently process the required input data for each expert. The workflow of *AMDB* begins with a UNet [38] and a sparse CNN to extract LiDAR features, followed by fully connected layers that generate proposal regions along with their confidence scores. These results

serve as inputs to *Latency-Prioritized Expert* and *Versatile Efficiency Expert*. The extraction of image features depends on the *Scenario-Adaptive Dispatcher*. If further information is required for reliable detection, *AMDB* will extract image features by ResNet. Based on depth information, it projects relevant image pixels into 3D space, applying Multiscale Pooling to reduce computational overhead, and fuses them with LiDAR voxel features to produce a multimodal output. These results are inputs for *Accuracy-Prioritized Experts*. The Multiscale Pooling and multimodal fusion processes are demonstrated in Fig. 5.

3.2. Scenario-Adaptive Dispatcher

The *Scenario-Adaptive Dispatcher (SAD)* dynamically dispatches each scenario to the most suitable expert based on traffic conditions and latency-accuracy requirements. Our expert selection algorithm considers real-world traffic safety needs: specifically, how far away objects are and how clearly they can be perceived.

So, the switch among experts is determined by two key parameters: object distance \mathcal{D} and clarity, which is inferred from the confidence \mathcal{C} of proposal regions provided by the *AMDB*. We classify traffic scenarios into three categories: 1) Close and Distinct Cases. All objects are clearly visible and no distant objects are present. This is characterized by all proposal regions $< \mathcal{D}$ and having confidence $\geq \mathcal{C}$. In such scenarios, as near-field objects provide abundant LiDAR information, even when compressed into bird’s eye view (BEV), a 2D CNN can effectively capture key patterns such as object surfaces and heights [20], ensuring reliable detection accuracy with minimal latency (Sec. 4.4). These cases will be routed to the *Latency-Prioritized Expert*. 2) Mixed Visibility Cases. Some objects are either distant but clearly visible or near but unclear. Mathematically, this corresponds to cases where i) some proposal regions $< \mathcal{D}$ with confidence $< \mathcal{C}$, or ii) the distance of some objects $\geq \mathcal{D}$ with confidence $\geq \mathcal{C}$. Here, objects have insufficient voxel data, requiring more complex 3D convolution to extract meaningful patterns, and they will be dispatched to the *Versatile Efficiency Expert*. 3) Distant and Uncertain Cases. Some objects are both distant and unclear. This occurs when some proposal regions $\geq \mathcal{D}$ with confidence $< \mathcal{C}$. In these challenging scenarios, significant voxel information is missing due to long distances, steep reflection angles, or occlusions, necessitating the integration of image data to compensate for missing LiDAR details. These cases will be processed by the *Accuracy-Prioritized Expert*.

3.3. Scenario-Optimized Experts

Latency-Prioritized Expert (LPE). *LPE* uses 2D CNNs for object detection based on BEV projections. It first projects proposal regions from the *AMDB* into BEV space, then extracts 2D features for object localization and

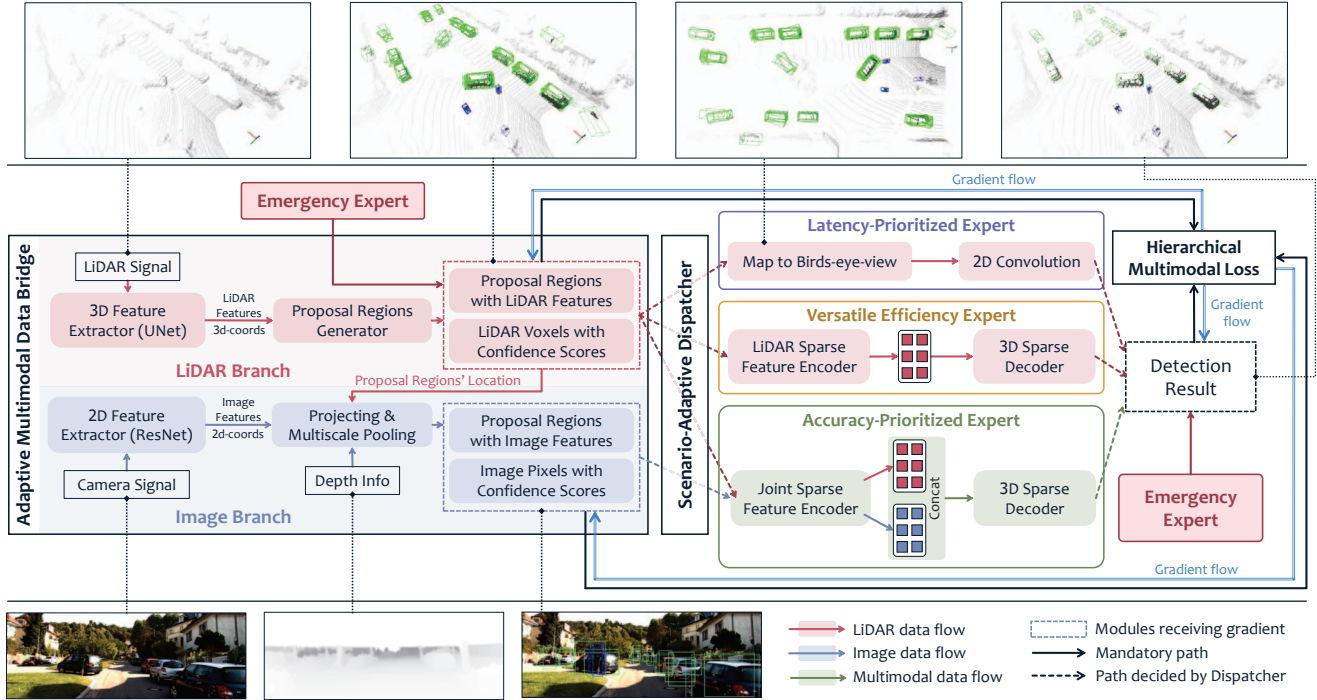


Figure 2. **System Structure of EMC2.** The central row illustrates the architecture of EMC2, while the top and bottom rows provide visual examples of core modules and relevant legends. The system integrates five main components: an *Adaptive Multimodal Data Bridge (AMDB)* for preprocessing raw multimodal input and generating expert-specific features; a *Scenario-Adaptive Dispatcher* that dispatches suitable experts based on scenario-specific features (see Fig. 3); the *Latency-Prioritized Expert* designed for simple scenarios requiring minimal inference delay; the *Versatile Efficiency Expert* designed for uncertain or distant object detection scenarios; and the *Accuracy-Prioritized Expert* responsible for handling complex, high-precision detection scenarios. Additionally, an *Emergency Expert API* is designed to support rapid response to hazardous or unseen situations. The overall training process incorporates loss terms for both final detection results and intermediate outputs from the *AMDB*, as described in Sec. 3.4.

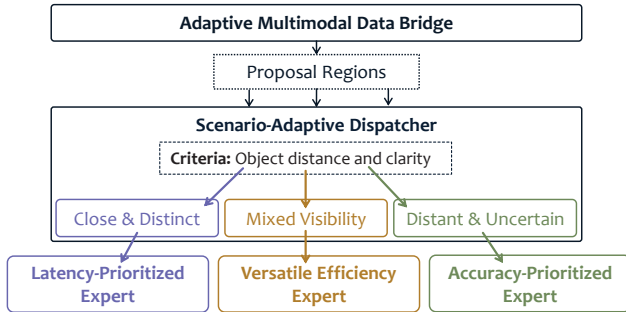


Figure 3. **Routing Strategy of the Scenario-Adaptive Dispatcher.** The Routing criteria is detailed mathematically in Sec. 3.2.

classification. It finally maps detection results back to 3D space. Compared to sparse 3D convolutions, its use of 2D operations reduces computation in outdoor ADS scenarios [20], allowing *LPE* to improve inference efficiency with minimal accuracy loss.

Versatile Efficiency Expert (VEE). *VEE* uses a 3D CNN to process proposal regions and confidence scores from the *AMDB*. After sparse convolution, voxel features are decoded into per-voxel classifications and bounding

boxes. Compared to *LPE*, 3D convolutions better preserve spatial structure, making it more robust for scenarios with insufficient LiDAR signal.

Accuracy-Prioritized Expert (APE). *APE* extends *VEE* by incorporating the multimodal signal. It employs a 3D CNN and decoder with the same structure but different parameters to process multimodal proposal regions (from *AMDB*) and fuses image data to recover missing LiDAR information. Image features compressed via Multiscale Pooling (Sec.3.5) are projected into 3D space and fused with corresponding voxels of LiDAR features, ensuring more robust understanding for occluded and distant objects while keeping computation feasible. The Multiscale Pooling and multimodal fusion are detailed in Fig. 5.

3.4. Collaborative Training of Multimodal Experts

Hierarchical Training Strategy: Multimodal Supervision with Triple Back-Propagation. During training, the performance of experts is highly dependent on the proposal regions produced by the *AMDB*, yet these proposals tend to be unreliable in the early training phase, leading to an unstable expert learning process and unstable *SAD* decisions. To address this, we adopt a hierarchical supervision strat-

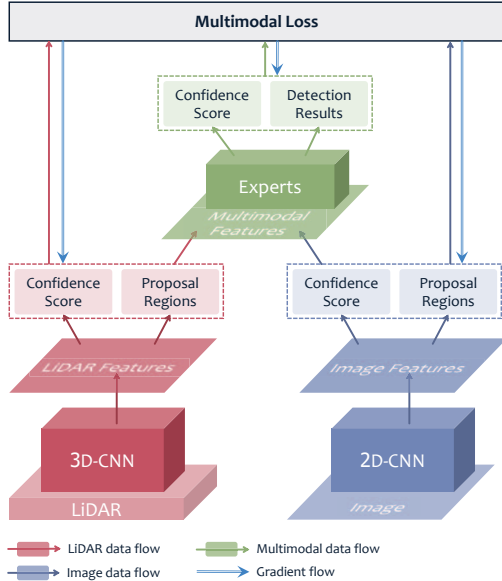


Figure 4. **Illustration of Hierarchical Training.** The three back-propagation paths originate from: 1) the final detection output, 2) the LiDAR branch output of *AMDB*, and 3) the image branch output of *AMDB*. Since expert training relies on initially unreliable *AMDB* outputs, direct supervision accelerates convergence and enhances representation learning.

egy with three distinct back-propagation pathways, as illustrated in Fig. 4: 1) supervision of the LiDAR branch outputs from the *AMDB*; 2) supervision of the image branch outputs from the *AMDB*; and 3) supervision of expert predictions and associated confidence scores. To further improve training stability, the LiDAR and image branches of the *AMDB* are pre-trained separately before joint optimization. Each back-propagation route is supervised by the following loss function, which collectively forms Multimodal Loss:

$$\mathcal{L}_{\text{route}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{reg}} \quad (1)$$

where \mathcal{L}_{cls} uses cross-entropy loss and \mathcal{L}_{reg} uses smooth- ℓ_1 loss. The Hierarchical Training not only stabilizes expert learning but also enhances the model’s adaptability to various data granularities. In turn, even under compressed data representations, it enables the effective use of our Multi-scale Pooling for memory efficiency, as detailed in Sec. 3.5.

Addressing Long-tail Effects in MoE Training. The MoE framework often suffers from long-tail effects, where certain experts receive insufficient training data, leading to imbalances in sub-datasets, and therefore affecting performance [39]. To address this, we propose a threefold training strategy: 1) Data subset division. We partition the dataset into sub-datasets based on target and auxiliary samples. Each expert \mathcal{E}_i is dispatched a sub-dataset \mathcal{S}_i ($i = 1, 2, 3$) containing target samples $\mathcal{T}_{\mathcal{S}_i}$, which are expected to be routed to the corresponding expert, and auxiliary samples $\mathcal{A}_{\mathcal{S}_i}$, which are not. We ensure that $\mathcal{T}_{\mathcal{S}_i} \cap \mathcal{T}_{\mathcal{S}_j} = \emptyset$ for any

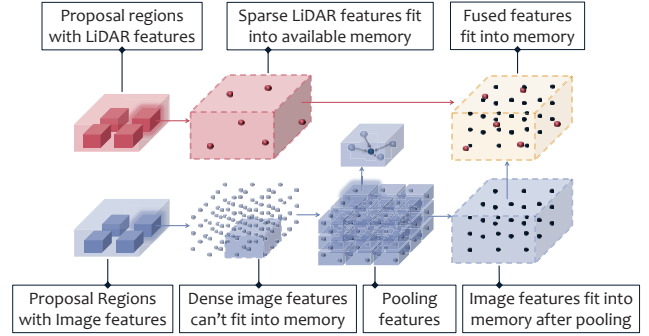


Figure 5. **Illustration of Multiscale Pooling and Multimodal Fusion.** Proposal regions with image features are first obtained from the 3D proposal space generated by the LiDAR branch of *AMDB*. After extracting the relevant image features from the corresponding image regions and projecting them into 3D space, pooling and fusion are then applied. Following the pooling step, image features of pixels that share the same 3D coordinate as a voxel are concatenated with the corresponding LiDAR voxel features, while those without matching voxel coordinates are concatenated with zero-filled features.

$i \neq j$. The inclusion of auxiliary samples, which are not intended for a given expert, helps prevent overfitting and enhances generalization. 2) Balanced sampling strategy. We adjust the selection probability \mathcal{P}_i for the sub-datasets which have fewer samples to ensure balanced training:

$$\mathcal{P}_i \times \mathcal{N}_{\mathcal{S}_i} = \mathcal{P}_j \times \mathcal{N}_{\mathcal{S}_j}, \quad \sum_{i=1}^3 \mathcal{P}_i = 1 \quad (2)$$

where $\mathcal{N}_{\mathcal{S}_i}$ represents the number of samples in sub-datasets \mathcal{S}_i . This ensures that experts dispatched to less frequent scenarios receive sufficient training iterations. 3) Adaptive optimizer. We introduce an optimizer that adapts the learning rate based on the proportion of target samples in a batch. Denoting α_0 as the base learning rate, the adaptive learning rate applied to an expert with a given batch is computed as:

$$\alpha = \left(1 + \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \mathbf{1}_{i \in \mathcal{T}} \right) \cdot \alpha_0 \quad (3)$$

where \mathcal{N} is the batch size, p is the proportion of target samples within current mini-batch, \mathcal{T} represents target samples in the batch. By regulating the gradient update rate based on the batch composition, this adaptive optimizer encourages each expert to prioritize its designated target samples while preventing excessive updates from auxiliary samples.

3.5. Algorithm Edge-adaptivity Optimization

Edge devices face challenges such as incomplete function libraries and limited cache. We adopt the open-source ONNX Runtime for compiling EMC2 on edge, which provides modules that help address these challenges and facilitate edge-oriented optimization. As 3D convolutions account for the majority of computation during inference, we

customize 3D sparse convolution and Multiscale Pooling to improve the algorithmic efficiency of EMC2.

3D Sparse Convolution. Since the 3D sparse convolution library is incomplete on most edge platforms, we develop a customized 3D sparse convolution library with parallel threads execution under ONNX Runtime. Let \mathcal{H} denote spatial resolution, \mathcal{N} the number of non-empty voxels, and \mathcal{C}_v the per-voxel computation, including channels and kernel size. By computing nonzero voxels only, 3D sparse convolution reduces complexity from $O(\mathcal{H}^3 \times \mathcal{C}_v)$ for dense convolutions to $O(\mathcal{N} \times \mathcal{C}_v)$, where $\mathcal{N} \ll \mathcal{H}^3$, cutting redundant operations by 65–80% [40].

Multiscale Pooling. To alleviate cache limitations, Multiscale 3D Sparse Pooling adaptively pools image features based on available memory before fusing them with LiDAR features (Fig.5). The pooling size is user-configurable, allowing flexibility for maximizing memory utilization. When combined with Hierarchical Training (Sec.3.4), Multiscale Pooling effectively reduces resource consumption while maintaining model reliability (Sec. 4.4).

3.6. EMC2 Computing System Optimization

Building upon the algorithm-level optimizations introduced in Sec. 3.5, we further enhance the ONNX execution efficiency through system-level improvements, specifically targeting memory management and computation graph optimization.

Memory Optimization. To further improve memory efficiency during inference, we implement three system-level techniques. 1) Overlapping communication and computation [41]. Matrix data is partitioned into segments processed and transferred in parallel, overlapping computation and communication. This reduces global memory access, improves cache usage, and mitigates memory bottlenecks. 2) Thread scheduling. We implement a staged thread management mechanism, where active threads are periodically terminated and re-initialized (Fig. 6). This approach limits unnecessary memory retention between stages, reducing system memory footprint. 3) Prefix-sum for sparse convolution. Parallel prefix-sum reduces index search time from $\mathcal{O}(\mathcal{N})$ to $\mathcal{O}(\log \mathcal{N})$ using GPU parallelism, where \mathcal{N} is the number of nonzero elements.

Computational Graph Optimization. ONNX partitions the computation graph into segments and assigns each to the most suitable hardware. Our optimization includes three stages: 1) Model pruning. Redundant parameters and operations are removed to simplify the graph; 2) Model quantization. Low-precision arithmetic replaces floating-point computation to improve hardware efficiency; 3) Computational graph fusion. Consecutive operators are merged to reduce memory access and improve execution, benefiting memory-bound CPUs and reducing kernel

launch overhead on GPUs.

4. Experiment and Result

4.1. Experiment Settings and Datasets

Experiment Platform. Experiments are conducted on the Jetson AGX Orin, an edge platform with limited computational resources, an NVIDIA Ampere GPU, a 4MB shared L2 cache, and hierarchical memory. For comparison, EMC2 is also evaluated on an NVIDIA A4000 GPU.

Dataset and Sample Difficulties. We evaluate 3D object detection on two public benchmarks: KITTI [42] and nuScenes [43]. KITTI provides 3,712 training and 3,769 validation samples, with object difficulty levels categorized as 30% easy, 40% moderate, and 30% hard. nuScenes contains 850 training and 150 test scenes, covering 10 object categories with 3D bounding box annotations. Before splitting the data, we analyze the confidence distribution of proposal regions generated by the pre-trained *AMDB*. A two-sample Kolmogorov–Smirnov test reveals a statistically significant difference between objects within and beyond a distance threshold D , which is empirically set to 23.5 meters for KITTI and 35 meters for nuScenes. The training subsets for each expert on the KITTI and nuScenes datasets are defined as follows: \mathcal{T}_{S_1} includes scenes where all objects are within D meters and labeled as “easy” or “moderate”; \mathcal{T}_{S_2} includes scenes with at least one object beyond D meters labeled as “easy” or “moderate,” or any object within D meters labeled as “hard”; and \mathcal{T}_{S_3} includes scenes where at least one object is both beyond D meters and labeled as “hard,” requiring multimodal processing. For each expert, the auxiliary subset \mathcal{A}_{S_i} is uniformly sampled from the other experts’ target subsets, with its size equal to that of the corresponding target subset.

Evaluation Metrics. KITTI uses R40 3D Average Precision (AP) with IoU thresholds of 0.7 for cars and 0.5 for pedestrians and cyclists. nuScenes adopts the NDS metric, combining mAP—computed via center distance matching (0.5 m to 4.0 m thresholds)—and five error terms: mean Average Translation (mATE), Scale (mASE), Orientation (mAOE), Velocity (mAVE), and Attribute (mAAE) Errors. Inference latency on Jetson AGX Orin is also reported to assess real-time performance.

Hierarchical Training Details. Allowing all experts to propagate gradients to *AMDB* causes instability due to overlapping training data, leading to repeated updates. To address this, *AMDB* is first pre-trained for 20 epochs, followed by joint training where gradients are propagated only from *APE*, which receives full multimodal input.

4.2. Comparison with Existing SOTA Methods

As shown in Tab. 1 and Tab. 2, EMC2 achieves SOTA accuracy and latency on KITTI and nuScenes. On the Jetson

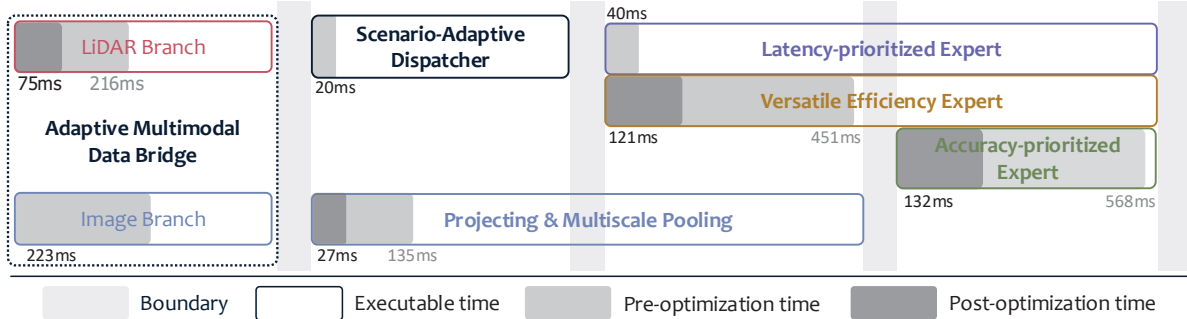


Figure 6. **Illustration of Thread Management.** Each solid box represents a module occupying threads. During inference, four boundaries are defined at which preceding threads are terminated, and their thread-specific address space is released upon crossing each boundary.

Method	Pedestrian 3D AP (R40) ↑			Car 3D AP (R40) ↑			Cyclist 3D AP (R40) ↑			Latency (ms) ↓	
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	A4000	Jetson
LiDAR-only Models											
M3DETR [45]	69.69	66.04	61.61	93.96	86.28	84.32	87.88	72.47	70.63	338	N/A
PV-RCNN [46]	N/A	N/A	N/A	91.18	87.65	83.14	N/A	N/A	N/A	45	1787
Voxel-RCNN [3]	N/A	N/A	N/A	92.23	85.04	82.50	N/A	N/A	N/A	70	N/A
GLNet-VR [47]	N/A	N/A	N/A	93.51	86.22	83.72	N/A	N/A	N/A	165	N/A
PartA2-Anchor [48]	66.87	59.68	54.60	92.15	82.92	82.10	90.34	70.05	66.89	95	N/A
PartA2-Free [48]	72.31	66.36	60.06	91.66	80.28	78.08	91.88	75.33	70.67	124	1064
PointPillar [20]	57.29	51.41	46.87	87.75	78.40	75.19	81.57	62.93	58.97	41	972
Second [40]	55.94	51.14	46.16	90.55	81.60	78.60	82.96	66.73	62.78	45	1322
Second-IoU [40]	61.10	54.66	49.50	91.53	82.36	79.62	90.73	71.23	66.26	58	N/A
3ONet [49]	72.55	65.21	60.22	94.24	87.32	84.17	92.47	75.11	71.18	100	N/A
Multimodal Models											
Voxel-RCNN [3]	N/A	N/A	N/A	92.08	85.90	83.36	N/A	N/A	N/A	331	965
LoGoNet [50]	70.02	63.72	59.46	92.04	85.04	84.31	91.74	75.35	72.42	100	N/A
VFF [51]	73.26	65.11	60.03	92.24	85.51	82.92	91.74	75.35	69.84	192	N/A
CAT-DET [52]	74.08	66.35	58.92	90.12	81.46	79.15	87.64	72.82	68.20	154	N/A
TED [53]	74.73	69.07	63.63	92.25	88.94	86.73	95.20	76.17	71.59	162	N/A
EMC2 (Ours)	74.92	69.92	66.81	94.00	89.35	88.15	95.06	79.87	77.62	160	372.5

Table 1. We evaluate different 3D object detection methods on the KITTI dataset. Some cells are denoted as *N/A* when the corresponding open-source implementation lacks support for the specific object class or is not compatible with Jetson devices.

Method	Size	mAP↑	NDS↑	Latency (ms)↓
FocalFormer [55]	189M	0.6640	0.7090	N/A
BEVFusion [56]	156M	0.6852	0.7138	N/A
EMC2 (ours)	87M	0.7241	0.7316	229.3

Table 2. Comparison of different methods on the nuScenes dataset.

AGX Orin platform, it also outperforms existing baseline models across all difficulty levels.

Results on KITTI. Pedestrians and cyclists face higher traffic risks [54], which remain challenging for most SOTA methods. EMC2 improves hard-set detection by 5% and 7% for pedestrians and cyclists, respectively, outperforming TED and LoGoNet (Tab. 1). For cars, it achieves 88.15% on the hard set, exceeding prior multimodal methods by up to 9%. It runs at 372.5 ms on Jetson, $2.5\times-2.5\times$ faster than methods requiring 965–1,787 ms.

Results on nuScenes. EMC2 improves mAP and NDS by 3.9% and 1.8%, respectively, outperforming recent

methods FocalFormer [55] and BEVFusion [56] (Tab. 2). It runs at 229.3 ms on Jetson AGX Orin, enabling real-time use on edge devices.

4.3. Analysis and Discussion

Benefits of Mixture-of-Experts. The balance between accuracy and efficiency observed in Sec. 4.2 is achieved through dynamic expert selection based on scenario characteristics, combined with algorithmic and system-level optimizations designed for edge devices, such as Jetson.

Empirical Dispatcher. The EMC2 dispatcher uses fixed distance and confidence thresholds to dispatch scenes to experts. The protocol is adapted per dataset format. This strategy is validated via expert activations on the nuScenes validation set (Fig. 7): *LPE* performs best in close-range, high-confidence scenes; *APE* in distant, low-confidence cases; and *VEE* in all others, supporting the effectiveness of empirical routing.

Optimized Parameter Size. EMC2 has 226M param-

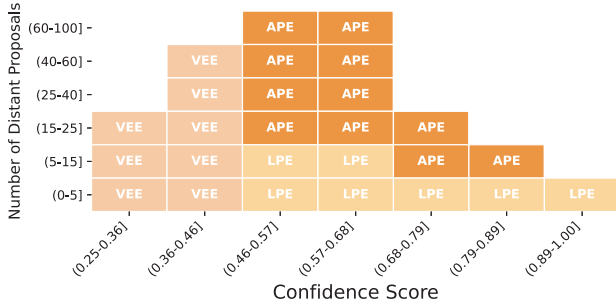


Figure 7. **Distribution of expert activation.** During EMC2 inference on the nuScenes validation set, scenes are grouped into a 2D grid based on two metrics: the y-axis indicates the number of *AMDB*-generated proposals beyond the distance threshold D (i.e., distant objects), and the x-axis represents the average confidence score of these proposals (higher values indicate clearer scenes). Each cell is labeled with the name of the expert that performs best for scenes falling into that bin.

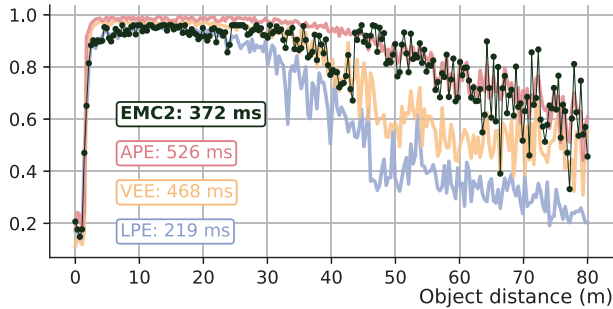


Figure 8. **Median Detection Confidence vs. Object Distances.** Four configurations are compared on the KITTI validation set: LPE (blue), VEE (peach), APE (rose), and EMC2 (green). More complex experts yield higher confidence but slower inference. By switching experts based on distance and clarity, EMC2 balances accuracy and efficiency.

eters, but its MoE framework activates only a subset during inference, averaging 87M on nuScenes and 146M on KITTI, which is fewer than those used by the full model and other SOTA baselines. System-level optimizations instead prioritize computational throughput and memory efficiency over parameter reduction.

Hardware Flexibility. The EMC2 is implemented in the ONNX format, an open format supported by various compilers and enables model compilation across diverse hardware platforms. As a result, EMC2 is not limited to Jetson and can be deployed on other edge platforms.

4.4. Ablation Study

Ablation on the Roles of Three Experts. We evaluate each expert on its target scenarios using the KITTI validation set. As shown in Tab. 3, *LPE*, *VEE*, and *APE* achieve comparable accuracy in *LPE* scenes, but *LPE* has the lowest inference latency on Jetson. In *VEE* scenarios, incor-

Method	AP \uparrow			Latency (ms) \downarrow
	Ped.	Car	Cyclist	
Ablation on <i>LPE</i>				
<i>LPE</i>	66.32	89.04	88.54	219
<i>VEE</i>	74.49	92.23	91.69	468
<i>APE</i>	74.34	93.21	92.41	526
Ablation on <i>VEE</i>				
W/ <i>VEE</i>	70.55	90.50	84.07	372.5
W/O <i>VEE</i>	68.26	86.40	79.17	512.7
Ablation on <i>APE</i>				
<i>VEE</i>	67.24	87.69	81.85	468
<i>APE</i>	70.55	90.50	84.18	526
Ablation on Hierarchical Training and Multiscale Pooling				
HT \checkmark , MP \checkmark	70.55	90.50	84.07	526
HT \checkmark , MP \times	70.55	90.50	84.18	1320
HT \times , MP \checkmark	63.11	88.68	75.98	529
HT \times , MP \times	66.34	89.10	80.83	1367

Table 3. **Results Across Three Ablation Configurations.** The main body of this table is divided into four sections corresponding to each part of the ablation study. In the third section, HT and MP are abbreviated for Hierarchical Training and Multiscale Pooling, respectively.

porating *VEE* improves both accuracy and latency over the baseline. For *APE*, which incorporates image features, accuracy improves further with only a minor latency increase, supported by the score distribution in Fig. 8.

Ablation on Hierarchical Training (HT) and Multiscale Pooling (MP). We evaluate four combinations of HT and MP on the full KITTI validation set: both applied, HT only, MP only, and neither. Tab. 3 shows that combining HT and MP reduces inference time on Jetson from 1320 ms to 526 ms without accuracy loss. For the car class, MP alone reduces accuracy from 80.83% to 75.98%, while HT alone improves it from 66.34% to 70.55%. HT facilitates multigranular feature learning, while MP enhances memory efficiency via compression.

5. Conclusion

We present EMC2, a multimodal 3D object detection solution for ADS based on a customized MoE architecture. This design enables adaptive expert selection across diverse traffic scenarios, balancing detection accuracy and efficiency. Experiments on KITTI and nuScenes show strong performance, particularly under challenging conditions. On the Jetson platform, EMC2 achieves a 159.06% speedup in inference, supporting real-time edge deployment. Future work will explore fully adaptive expert selection to enhance system responsiveness and resource efficiency.

References

- [1] Shih-Chieh Lin, Yunqi Zhang, Chang-Hong Hsu, Matt Skach, Md E. Haque, Lingjia Tang, and Jason Mars. The architectural implications of autonomous driving: Constraints and acceleration. In *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '18*, page 751–766, New York, NY, USA, 2018. Association for Computing Machinery. 1
- [2] Yi Xiao, Felipe Codevilla, Akhil Gurram, Onay Urfalioglu, and Antonio M López. Multimodal end-to-end autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(1):537–547, 2020. 2
- [3] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. *arXiv:2012.15712*, 2020. 2, 3, 7
- [4] Hao Frank Yang, Yifan Ling, Cole Kopca, Sam Ricord, and Yin Hai Wang. Cooperative traffic signal assistance system for non-motorized users and disabilities empowered by computer vision and edge artificial intelligence. *Transportation Research Part C: Emerging Technologies*, 145:103896, 2022. Elsevier.
- [5] Hao Yang, Jiarui Cai, Meixin Zhu, Chenxi Liu, and Yin Hai Wang. Traffic-informed multi-camera sensing (TIMS) system based on vehicle re-identification. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):17189–17200, 2022. IEEE. 2
- [6] Guibiao Liao, Jiankun Li, and Xiaoqing Ye. Vlm2scene: Self-supervised image-text-lidar learning with foundation models for autonomous driving scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 3351–3359, 2024. 2
- [7] Hao Frank Yang, Yang Zhao, Jiarui Cai, Meixin Zhu, Jenq-Neng Hwang, and Yiran Chen. Mitigating bias of deep neural networks for trustworthy traffic perception in autonomous systems. In *Proceedings of the 2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 633–638, 2024. IEEE.
- [8] Chenxi Liu, Hao Yang, Ruimin Ke, Wei Sun, Julia Wang, and Yin Hai Wang. Cooperative and comprehensive multi-task surveillance sensing and interaction system empowered by edge artificial intelligence. *Transportation Research Record*, 2677(9):652–668, 2023. SAGE Publications.
- [9] Hao Frank Yang, Jiarui Cai, Chenxi Liu, Ruimin Ke, and Yin Hai Wang. Cooperative multi-camera vehicle tracking and traffic surveillance with edge artificial intelligence and representation learning. *Transportation Research Part C: Emerging Technologies*, 148:103982, 2023. Elsevier. 2
- [10] Chen Zhang, Peng Li, Guangyu Sun, Yijin Guan, Bingjun Xiao, and Jason Cong. Optimizing fpga-based accelerator design for deep convolutional neural networks. In *Proceedings of the 2015 ACM/SIGDA international symposium on field-programmable gate arrays*, pages 161–170, 2015. 2
- [11] Tianqi Chen, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Yan, Haichen Shen, Meghan Cowan, Leyuan Wang, Yuwei Hu, Luis Ceze, et al. {TVM}: An automated {End-to-End} optimizing compiler for deep learning. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pages 578–594, 2018. 2, 3
- [12] Jiarui Cai, Yizhou Wang, and Jenq-Neng Hwang. Ace: Ally complementary experts for solving long-tailed recognition in one-shot. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 112–121, 2021. 2
- [13] Yunxin Li, Shen Yuan Jiang, Baotian Hu, Longyue Wang, Wanqi Zhong, Wenhan Luo, Lin Ma, and Min Zhang. Unimoe: Scaling unified multimodal llms with mixture of experts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 2
- [14] Jiachen Li, Xinyao Wang, Sijie Zhu, Chia-Wen Kuo, Lu Xu, Fan Chen, Jitesh Jain, Humphrey Shi, and Longyin Wen. Cumo: Scaling multimodal llm with co-upcycled mixture-of-experts. *Advances in Neural Information Processing Systems*, 37:131224–131246, 2024. 2
- [15] Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. A survey on mixture of experts. *arXiv preprint arXiv:2407.06204*, 2024. 2
- [16] Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. *arXiv preprint arXiv:2409.16040*, 2024. 2
- [17] Jiao He, Jiezhong Qiu, Aohan Zeng, Zhilin Yang, Jidong Zhai, and Jie Tang. Fastmoe: A fast mixture-of-expert training system. *arXiv preprint arXiv:2103.13262*, 2021. 2
- [18] Charles R. Qi, Xinlei Chen, Or Litany, and Leonidas J. Guibas. Imvotenet: Boosting 3d object detection in point clouds with image votes. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4403–4412, 2020. 2
- [19] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *arXiv preprint arXiv:1907.03670*, 2019. 3
- [20] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12689–12697, 2019. 2, 3, 4, 7
- [21] Yin Zhou and Oncel Tuzel. Voxnet: End-to-end learning for point cloud based 3d object detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. 2
- [22] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10526–10535, 2020. 2
- [23] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11779–11788, 2021. 3
- [24] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11037–11045, 2020. 3

- [25] Andrew M Wallace, Abderrahim Halimi, and Gerald S Buller. Full waveform lidar for adverse weather conditions. *IEEE transactions on vehicular technology*, 69(7):7064–7077, 2020. 3
- [26] You Li, Pierre Duthon, Michele Colomb, and Javier Ibanez-Guzman. What happens for a tof lidar in fog? *IEEE Transactions on Intelligent Transportation Systems*, 22(11):6670–6681, 2020. 3
- [27] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *2019 international conference on robotics and automation (ICRA)*, pages 3288–3295. IEEE, 2019. 3
- [28] Tengting Huang, Zhe Liu, Xiwu Chen, and Xiang Bai. Epnnet: Enhancing point features with image semantics for 3d object detection. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 35–52, Cham, 2020. Springer International Publishing. 3
- [29] Yingwei Li, Adams Wei Yu, Tianjian Meng, Ben Caine, Jiquan Ngiam, Daiyi Peng, Junyang Shen, Yifeng Lu, Denny Zhou, Quoc V. Le, Alan Yuille, and Mingxing Tan. Deep-fusion: Lidar-camera deep fusion for multi-modal 3d object detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17161–17170, 2022. 3
- [30] Yanwei Li, Yilun Chen, Xiaojuan Qi, Zeming Li, Jian Sun, and Jiaya Jia. Unifying voxel-based representation with transformer for 3d object detection. In *Advances in Neural Information Processing Systems*, 2022. 3
- [31] Guotao Xie, Zhiyuan Chen, Ming Gao, Manjiang Hu, and Xiaohui Qin. Ppf-det: Point-pixel fusion for multi-modal 3d object detection. *IEEE Transactions on Intelligent Transportation Systems*, 25(6):5598–5611, 2024. 3
- [32] Ziyang Song, Guoxing Zhang, Lin Liu, Lei Yang, Shaoqing Xu, Caiyan Jia, Feiyang Jia, and Li Wang. Robofusion: Towards robust multi-modal 3d object detection via sam. *arXiv preprint arXiv:2401.03907*, 2024. 3
- [33] Zheng Wang and Michael O’Boyle. Machine learning in compiler optimization. *Proceedings of the IEEE*, 106(11):1879–1901, 2018. 3
- [34] Hugh Leather, Edwin Bonilla, and Michael O’boyle. Automatic feature generation for machine learning-based optimising compilation. *ACM Transactions on Architecture and Code Optimization (TACO)*, 11(1):1–32, 2014. 3
- [35] NVIDIA Corporation. *NVIDIA TensorRT Developer Guide*, 2024. Accessed: 2025-02-13. 3
- [36] Amit Sabne. *XLA: Compiling Machine Learning for Peak Performance*, 2020. 3
- [37] ONNX Community. Onnx: Open neural network exchange. <https://github.com/onnx/onnx>, 2024. Accessed: YYYY-MM-DD. 3
- [38] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 3
- [39] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991. 5
- [40] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10), 2018. 6, 7
- [41] Juraj Hromkovič. *Communication complexity and parallel computing*. Springer Science & Business Media, 2013. 6
- [42] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 6
- [43] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multi-modal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 6
- [44] Muhammad Aslam. Introducing kolmogorov–smirnov tests under uncertainty: an application to radioactive data. *ACS omega*, 5(1):914–917, 2019.
- [45] Tianrui Guan, Jun Wang, Shiyi Lan, Rohan Chandra, Zuxuan Wu, Larry Davis, and Dinesh Manocha. M3detr: Multi-representation, multi-scale, mutual-relation 3d object detection with transformers. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 772–782, 2022. 7
- [46] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10529–10538, 2020. 7
- [47] Jiajia Liao, Yujun Liu, Yingchao Piao, Jinhe Su, Guorong Cai, and Yundong Wu. Gle-net: A global and local ensemble network for aerial object detection. *International Journal of Computational Intelligence Systems*, 15(1):2, 2022. 7
- [48] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE transactions on pattern analysis and machine intelligence*, 43(8):2647–2664, 2020. 7
- [49] Hiep Anh Hoang and Myungsik Yoo. 3onet: 3d detector for occluded object under obstructed conditions. *IEEE Sensors Journal*, 2023. 7
- [50] Xin Li, Tao Ma, Yuenan Hou, Botian Shi, Yuchen Yang, Youquan Liu, Xingjiao Wu, Qin Chen, Yikang Li, Yu Qiao, et al. Logonet: Towards accurate 3d object detection with local-to-global cross-modal fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17524–17534, 2023. 7
- [51] Yanwei Li, Xiaojuan Qi, Yukang Chen, Liwei Wang, Zeming Li, Jian Sun, and Jiaya Jia. Voxel field fusion for 3d object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 7
- [52] Yanan Zhang, Jiaxin Chen, and Di Huang. Cat-det: Contrastively augmented transformer for multi-modal 3d object detection. In *Proceedings of the IEEE/CVF Conference on*

Computer Vision and Pattern Recognition, pages 908–917, 2022. [7](#)

- [53] Hai Wu, Chenglu Wen, Wei Li, Xin Li, Ruigang Yang, and Cheng Wang. Transformation-equivariant 3d object detection for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2795–2802, 2023. [7](#)
- [54] Nils Lubbe, Yi Wu, and Hanna Jeppsson. Safe speeds: fatality and injury risks of pedestrians, cyclists, motorcyclists, and car drivers impacting the front of another passenger car as a function of closing speed and age. *Traffic safety research*, 2:000006–000006, 2022. [7](#)
- [55] Yilun Chen, Zhiding Yu, Yukang Chen, Shiyi Lan, Anima Anandkumar, Jiaya Jia, and Jose M Alvarez. Focalformer3d: focusing on hard instance for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8394–8405, 2023. [7](#)
- [56] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L. Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird’s-eye view representation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2774–2781, 2023. [7](#)