Ray Denoising: Depth-aware Hard Negative Sampling for Multi-view 3D Object Detection

Feng Liu[1⋆] Tengteng Huang[2] Qianjing Zhang[2] Haotian Yao[2] Chi Zhang[2]
Fang Wan[1] Qixiang Ye[1] Yanzhao Zhou[1†]

[1] University of Chinese Academy of Sciences
liufeng20@mails.ucas.ac.cn {wanfang,qxye,zhouyanzhao}@ucas.ac.cn
[2] Mach Drive
{tengteng.huang,qianjing.zhang,haotian.yao,chi.zhang}@mach-drive.com

Abstract. Multi-view 3D object detection systems often struggle with generating precise predictions due to the challenges in estimating depth from images, increasing redundant and incorrect detections. Our paper presents Ray Denoising, an innovative method that enhances detection accuracy by strategically sampling along camera rays to construct hard negative examples. These examples, visually challenging to differentiate from true positives, compel the model to learn depth-aware features, thereby improving its capacity to distinguish between true and false positives. Ray Denoising is designed as a plug-and-play module, compatible with any DETR-style multi-view 3D detectors, and it only minimally increases training computational costs without affecting inference speed. Our comprehensive experiments, including detailed ablation studies, consistently demonstrate that Ray Denoising outperforms strong baselines across multiple datasets. It achieves a 1.9% improvement in mean Average Precision (mAP) over the state-of-the-art StreamPETR method on the NuScenes dataset. It shows significant performance gains on the Argoverse 2 dataset, highlighting its generalization capability. The code is available at https://github.com/LiewFeng/RayDN.

Keywords: Multi-view 3D Object Detection · Depth-aware Hard Negative Sampling · Ray Denoising

1 Introduction

3D object detection is a crucial component in autonomous driving systems, drawing considerable interest from the computer vision community. The field of image-based 3D object detection [9, 10, 13, 14, 37, 38] is experiencing a surge in research due to its cost-effectiveness compared to LiDAR-based solutions. A key challenge in multi-view 3D object detection, which relies on images from surrounding cameras, is the difficulty in estimating depth from images, leading to duplicate predictions, as shown in Figure 1.

⋆ Work was done during internship at Mach Drive. † Corresponding Author.

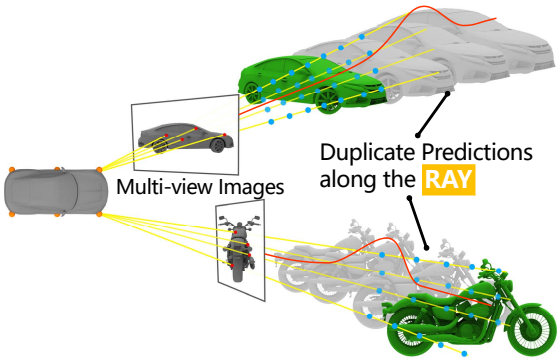


Fig. 1: The challenge of estimating depth from images in multi-view 3D object detection leads to duplicate predictions and false positive detections along camera rays. Best viewed in color.

Despite the methodological improvements, multi-view 3D object detectors struggle to reduce false positive predictions arising from depth ambiguities. Several recent studies [8, 15, 17, 19, 22, 26, 34, 41, 47] have attempted to tackle this issue by incorporating temporal information. However, these methods do not explicitly account for the 3D structure of the scene, which limits their potential for further enhancement.

Furthermore, previous works have explored applying general techniques such as Non-Maximum Suppression (NMS) and Focal Loss to mitigate duplicate predictions. NMS, a post-processing technique, targets false positive predictions with high Intersection over Union (IoU) but is less effective when these predictions are scattered along rays with low IoU. Focal Loss, a loss function designed to reduce high-confidence false positive predictions, has also been implemented. However, it has been observed that multi-view 3D object detectors using Focal Loss still face challenges in effectively resolving the problem of false positive predictions along the same rays.

Our quantitative analysis highlighted the importance of addressing false positive predictions along the same ray as Ground Truth. By utilizing the precise positional data of Ground Truth objects, we could identify and eliminate these redundant predictions within the state-of-the-art StreamPETR method [34]. This process significantly enhanced 5.4% in mean Average Precision (mAP), emphasizing the critical need for models to improve their depth estimation capabilities. The substantial improvement underscores the potential of refining depth estimation to suppress these predictions and enhance overall detection performance.

Our key observation is that false positives often occur along camera rays due to the inherent limitation in conventional multi-view object detectors. Since the depth information for each pixel is not accurately estimated, the position embedding can only encode the ray direction. As a result, queries on the same ray will consistently interact with identical visual features from the image, leading to numerous duplicate predictions (false positives) along that ray. This scenario

underscores the model's need to learn depth-aware features that can discern objects in depth despite the visual features being the same for objects along the same ray. We propose a novel method called Ray Denoising (*i.e.*, RayDN). This framework is inherently flexible and does not limit the choice of distribution for sampling depth-aware hard negative samples. Based on our ablation studies, we have chosen the Beta distribution for its effectiveness in capturing the spatial distribution of false positives that models are likely to generate. This choice enables Ray Denoising to create depth-aware hard negative samples used for denoising, thereby enhancing the model's ability to learn more robust features and representations for distinguishing false positives along the ray, as depicted in Figure 2. Ray Denoising introduces only a marginal increase in computational costs during the training stage without affecting inference speed.

To summarize our contributions, we highlight the following key points. Firstly, we have identified the persistent challenge of false positive predictions along the same ray, which acts as a bottleneck in the performance of multi-view 3D object detectors. Secondly, we introduce Ray Denoising, a novel denoising method that utilizes the Beta distribution to create depth-aware hard negative samples along rays. This method explicitly takes into account the 3D structure of the scene, offering a flexible solution compatible with any DETR-style multi-view 3D detector to address the issue of duplicate predictions along rays. Lastly, our method achieves state-of-the-art results on the NuScenes dataset [2], significantly enhancing the performance of multi-view 3D object detectors. Specifically, we have improved upon the current state-of-the-art method, StreamPETR, by 1.9% in mean Average Precision (mAP), thereby demonstrating the effectiveness of Ray Denoising.

2 Related Work

2.1 Image-based 3D Object Detection

3D object detection is a cornerstone task in autonomous driving systems. The domain has witnessed notable advancements in monocular 3D object detection, largely attributed to the KITTI benchmark [5]. This progress has been driven by a range of research studies [1, 24, 28, 31, 35, 44]. Nevertheless, monocular systems face limitations due to their dependence on a single viewpoint and limited data, which hampers their capacity to manage complex scenarios. To overcome these limitations, extensive benchmarks [2, 32, 39] have been developed, offering multi-viewpoint data that enriches the field of multi-view 3D object detection and propels the evolution of advanced detection methodologies.

The multi-view 3D object detection research is primarily bifurcated into dense Bird's Eye View (BEV)-based and sparse query-based algorithms. Our work is categorized under the sparse query-based algorithms. Dense BEV-based algorithms convert multi-view image features into a dense BEV representation using the Lift-Splat-Shoot (LSS) technique [27]. To mitigate overfitting from LSS, BEVDet [9] recommends data augmentation on BEV features. BEVDepth [14] introduces depth estimation supervision to enhance LSS precision. Compared to

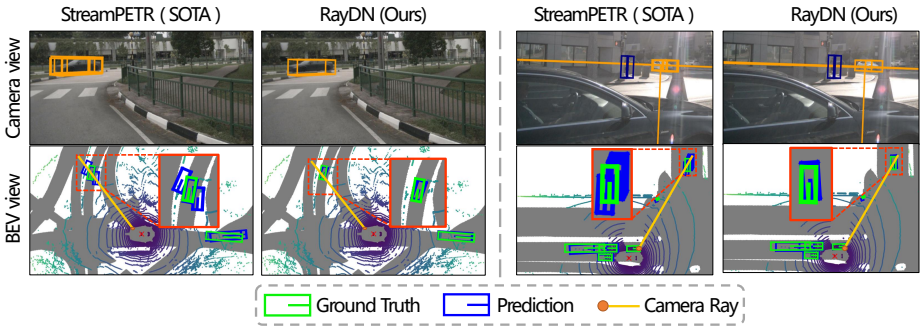


Fig. 2: The proposed Ray Denoising approach (right) effectively reduces false positive detections along the ray (highlighted by red rectangles) in the previous state-of-the-art method StreamPETR [34] (left). Best viewed by zooming on the screen.

LSS-like methods, BEVFormer-like and DETR-like methods can leverage the strong capabilities of transformers for spatial-temporal modeling. BEVFormer-like methods use dense BEV features to enable multiple tasks (*e.g.*, detection and segmentation) but also introduce high computational costs. Sparse query-based algorithms such as DETR3D [37] and PETR [21] utilize learnable 3D object queries and position encoding to engage with multi-view 2D features. DAT [43] introduces a pipeline to construct depth-aware position embeddings and performs contrastive learning on BEV features to tackle duplicate predictions. Recent studies [8, 13, 15, 17, 19, 22, 26, 34] have integrated temporal information to refine these methods further.

2.2 Hard Negative Samples Mining

Hard negative sample mining is a strategy to minimize false positive predictions in object detection. Early works like R-CNN [6] employed standard hard negative mining methods [4, 33], where misclassified samples were utilized for retraining. OHEM [29] and RetinaNet [16] introduced adaptive sampling and focal loss to address the challenges of heuristics and hyper-parameters in hard negative sample selection. These methods select hard negative samples from existing predictions based on confidence scores and matched ground-truth object classes. Our approach takes a different path by constructing new hard negative samples based on the locations of ground-truth objects rather than selecting from existing predictions or depending on prediction confidence and matched ground-truth object classes.

2.3 Denoising in Object Detection

In 2D object detection, models such as DETR [3] and its variants [20,36,46] often face convergence issues, which are partly due to the instability of bipartite graph matching and the inconsistency in optimization goals during the early stages of

training. To tackle this, DN-DETR [12] introduces the concept of feeding noisy ground-truth bounding boxes into the Transformer decoder, with the model being trained to reconstruct the original, clean boxes. This process, known as denoising, involves using queries initialized with these noisy ground-truth bounding boxes, referred to as denoising queries. DINO [42] further refines this approach by generating both positive and negative denoising queries for each ground-truth bounding box, with the negative queries incorporating more noise to improve the model's ability to reject incorrect predictions.

In the realm of 3D object detection, DETR-style methods [19, 22, 34] perpetuate the denoising approach by creating a denoising query for each ground-truth bounding box. Based on the noise level, these queries are classified as either an object or 'no object'. However, these methods neglect to incorporate knowledge of the 3D scene structure when generating denoising queries. In multi-view 3D object detection, the absence of depth information for each pixel results in models producing duplicate predictions along the same ray with different depths. To combat these false positive predictions, we harness the 3D structure knowledge of the scene. We devise multiple negative ray-denoising queries distributed along the rays, spatially distinct from the traditional negative denoising queries surrounding the ground-truth objects. This innovative design markedly improves the detector's capability to differentiate between true positive predictions and false positive predictions along the same ray.

3 Methodology

This section elaborates on our proposed method, Ray Denoising, for multi-view 3D object detection. We start with a comprehensive overview of the DETR-style multi-view 3D object detector framework in Section 3.1. Following that, we describe the three pivotal steps for integrating Ray Denoising into the framework to address the challenge of duplicate predictions. Section 3.2 outlines the ray generation process. Section 3.3 explains utilizing the Beta distribution family for sampling reference points. Section 3.4 details the construction of spatial denoising queries, referred to as Ray Queries, along each ray. Lastly, Section 3.5 delves into the essential differences between our Ray Queries and the denoising queries employed in prior works and the influence of Ray Denoising on the model training process.

3.1 Overview

Ray Denoising is designed to be integrated into any DETR-style multi-view 3D object detector [15, 17, 19, 21, 22, 34, 37]. These detectors typically consist of a convolutional network-based feature encoder and a transformer-based decoder. The process begins by feeding N surround-view images $\boldsymbol{I} = \{I_i \in \mathbb{R}^{3\times H_I\times W_I}, i = 1, 2, ..., N\}$ into the feature encoder (e.g., ResNet101 [7]) to extract image features $\boldsymbol{F} = \{F_i \in \mathbb{R}^{C\times H_F\times W_F}, i = 1, 2, ..., N\}$. Here, H_I and W_I denote the image dimensions, H_F and W_F are the feature dimensions, and C represents the

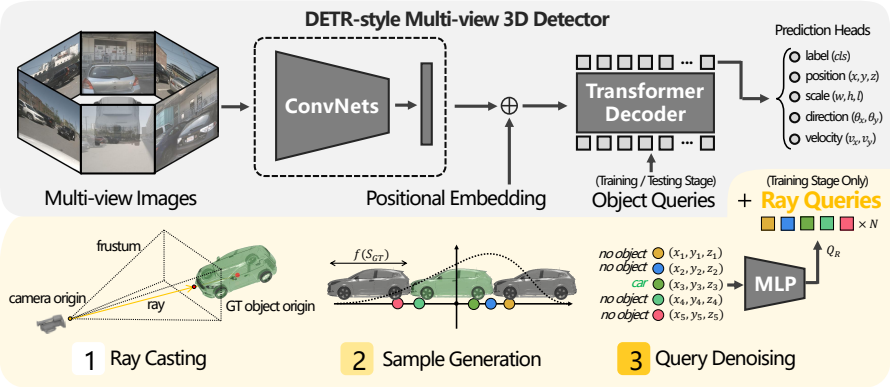


Fig. 3: Overall framework of the Ray Denoising approach, a plug-and-play training technique for DETR-style multi-view 3D object detectors, focuses on refining the model's ability to distinguish true positives from false positives in depth. Casting rays and sampling depth-aware denoising queries effectively tackle the challenge of false positives arising from the inherent difficulties in visually estimating depth, leading to substantial improvements in detection performance over strong baselines. Best viewed in color and by zooming on the screen.

number of channels in the feature. To facilitate 3D perception, multiple points within each camera's frustum are transformed and encoded into position embeddings [21]. These embeddings enable the multi-view image features to interact with 3D queries.

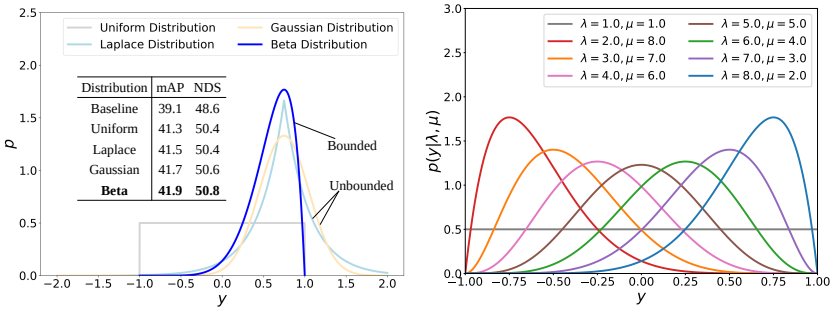
The final step involves N object queries $\mathbf{Q} \in \mathbb{R}^{N\times 256}$, which are derived from a set of learnable 3D reference points $\mathbf{P} \in \mathbb{R}^{N\times 3}$. These queries interact with the multi-view image features $\boldsymbol{F}$ in the transformer decoder, employing multi-layer cross-attention to identify objects. The decoder's output features are then processed by prediction heads (multi-layer perceptron, *i.e.*, MLP) to yield classification scores (cls), position offsets (x, y, z), scales (w, h, l), directions (θ_x, θ_y), and velocities (v_x, v_y).

3.2 Ray Casting

The reference points for our ray denoising queries are distributed along the camera ray, which extends from the camera's optical center to the ground-truth object on the image plane. To establish this ray, we project the 3D center of the ground-truth object into the camera frustum space using the following transformation:

$$\mathbf{C}' = \mathbf{K} \cdot \mathbf{C}_{GT}, \tag{1}$$

where $\mathbf{K}$ is the 4×4 transformation matrix that maps points from 3D world space to camera frustum space. $\mathbf{C}_{GT} = (x, y, z, 1)$ is the center of the ground-truth object in 3D space, and $\mathbf{C}' = (u \times d, v \times d, d, 1)$ is the corresponding



Distribution	mAP	NDS
Baseline	39.1	48.6
Uniform	41.3	50.4
Laplace	41.5	50.4
Gaussian	41.7	50.6
Beta	**41.9**	**50.8**

(a) Distribution Comparison **(b)** Beta distribution family

Fig. 4: (a) Distribution comparison showing that the Beta distribution is bounded between -1 and 1, unlike the Laplace and Gaussian distributions, which are unbounded. (b) The Beta distribution family, with the x-range adjusted from $[0, 1]$ to $[-1, 1]$ using the transformation $y = 2x - 1$. Best viewed in color.

projected center in the camera frustum, with (u, v) being the pixel coordinates and d the depth value.

With the depth of the ground-truth object's center known, we can determine the coordinates of valid reference points along the ray as follows:

$$\hat{d}_i = d + \beta_i \cdot f(\mathbf{S}_{GT}), \tag{2}$$

where f is a function that encodes the average scale of the ground-truth object, calculated as $f(\mathbf{S}_{GT}) = k \cdot \frac{w+h+l}{6}$. The radius k defines the valid distribution range for reference points, and β_i is the offset for the i-th reference point.

The position of the reference point is then obtained by:

$$\hat{\mathbf{P}}_i = \mathbf{K}^{-1} \cdot \hat{\mathbf{C}'}_i, \tag{3}$$

where $\hat{\mathbf{C}'}_i = (u \times \hat{d}_i, v \times \hat{d}_i, \hat{d}_i, 1)$.

3.3 Sample Generation

With the valid reference points along the rays defined, we proceed to sample N reference points to emulate the distribution of false positive predictions that arise from depth ambiguities. A straightforward uniform distribution has proven effective, yielding a 2.2% mAP improvement as depicted in Figure 4a. To more accurately simulate the distribution of false positives, we consider two key aspects. First, we assess whether models tend to predict objects as being closer or further away from the ego-vehicle relative to the actual positions of the ground-truth objects. Second, we examine whether models are more likely to predict objects near or far from the center of the ground-truth objects.

Distributions like the Beta, Gaussian, and Laplace are suitable for handling complex scenarios, with the Gaussian and Laplace being unbounded. However,

samples with significant offsets from the ground truth can result in large regression losses, negatively impacting optimization. As shown in Figure 4a, the Beta distribution slightly outperforms the others. Notably, the uniform distribution is a special case within the Beta family. For a versatile sampling strategy applicable to all scenarios, we select the Beta distribution family, characterized by the probability density function:

$$p(x|\lambda,\mu) = \frac{\Gamma(\lambda+\mu)}{\Gamma(\lambda)+\Gamma(\mu)} x^{\lambda-1}(1-x)^{\mu-1}, \tag{4}$$

where λ and μ are hyper-parameters that shape the distribution. The Gamma function, denoted as $\Gamma(x)$, is defined as:

$$\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt. \tag{5}$$

The beta distribution's probability density function (PDF), visualized in Figure 4b, reveals its behavior under different parameter settings. When λ equals μ, the distribution is symmetric, reflecting a scenario where the model predicts objects with equal likelihood at closer or further distances. For instance, with $\lambda = \mu = 1$, the distribution simplifies to a uniform distribution. Adjusting λ and μ to be higher shifts the reference points of the denoising queries closer to the ground-truth object's center and vice versa.

When λ exceeds μ, such as in the case of $\lambda = 8$ and $\mu = 2$, the distribution suggests that models are more likely to predict objects further away from the ego-vehicle. Conversely, when λ is less than μ, for example, $\lambda = 2$ and $\mu = 8$, the distribution indicates a propensity for models to predict objects closer to the ego-vehicle. This flexibility in the Beta distribution allows for tailored sampling strategies that can adapt to the specific challenges of depth estimation in multi-view 3D object detection.

3.4 Query Denoising

We construct ray denoising queries from the sampled N reference points. Specifically, a multi-layer perceptron (MLP) is utilized to project the normalized 3D coordinates of each point into a latent feature space. These N ray denoising queries $\boldsymbol{q}_r$ are combined with the learnable object queries $\boldsymbol{q}_o$ from the baseline detector and are input into the transformer decoder. An attention mask is employed during the self-attention operations to prevent information leakage from the ray denoising queries to the object queries. The query closest to the center of the ground truth (GT) object is designated as the positive query, inheriting the class label of the corresponding GT object. The remaining $N-1$ queries are labeled as background, *i.e.*, 'no object'. The loss for ray denoising queries adheres to the same criteria as for the learnable object queries, using Focal Loss for classification and L1 Loss for regression. This process ensures that the model is effectively trained to discern between true positives and false positives along the same ray.

3.5 Discussion

Our Ray Denoising approach is based on the pivotal observation that image-based 3D detection systems often struggle to distinguish true positives from false positives along camera rays. DETR-style multi-view 3D object detectors implicitly learn depth estimation from ground truth supervision. However, the randomly distributed reference points of learnable queries do not fully leverage the available ground truth information. While these reference points are updated during training, they fail to provide adequate hard negative samples for each object in every scene. To enhance the utilization of ground truth information, traditional denoising techniques introduce additional reference points distributed uniformly around the ground-truth object during training. These instance-specific reference points have improved detection performance [19,22,34]. However, they overlook the depth ambiguities intrinsic to multi-view 3D object detection. The absence of precise depth information for each pixel means that the position embedding can only encode ray direction, not depth. This leads to queries on the same ray interacting with the same image features, resulting in redundant predictions. Ray Denoising diverges from traditional techniques by strategically generating reference points along rays that extend from cameras to objects. This approach explicitly considers the 3D structure of each object in the scene, providing a sufficient number of hard negative samples. During training, these Ray Queries interact within the self-attention layer, effectively guiding the model to suppress depth-ambiguous spatial hard negative samples. This interaction enhances the detector's ability to differentiate between true positive predictions (objects) and false positive predictions (duplicate detections), improving detection accuracy.

4 Experiment

4.1 Dataset and Metrics

Our model's performance was assessed using two datasets: nuScenes [2] and Argoverse 2 [39]. The nuScenes dataset include 1000 video sequences. which are split into training (700 videos), validation (150 videos), and testing (150 videos) sets, each approximately 20 seconds long with annotations at 0.5-second intervals. The dataset comprises 1.4 million annotated 3D bounding boxes across ten object classes. Evaluation metrics include the mean Average Precision (mAP) and five true positive metrics: ATE, ASE, AOE, AVE, and AAE, which assess translation, scale, orientation, velocity, and attribute errors. The nuScenes Detection Score (NDS) is a comprehensive score derived from these metrics, offering an overall performance evaluation.

The Argoverse 2 dataset contains 1000 unique scenes, each 15 seconds long, annotated at 10 Hz. The scenes are divided into 700 for training, 150 for validation, and 150 for testing. The evaluation covers 26 categories within a 150-meter range, addressing long-range perception tasks. Metrics include the mAP and the Composite Detection Score (CDS), which integrates three other true positive metrics: ATE, ASE, and AOE.

Table 1: Comparison on the nuScenes validation set. † Indicates methods that benefit from perspective-view pre-training.

Methods	Backbone	Image Size	mAP	NDS	mATE	mASE	mAOE	mAVE	mAAE
BevDet4D [8]	ResNet50	256×704	32.2	45.7	0.703	0.278	0.495	0.354	0.206
PETRv2 [22]	ResNet50	256×704	34.9	45.6	0.700	0.275	0.580	0.437	0.187
BEVDepth [14]	ResNet50	256×704	35.1	47.5	0.629	0.267	0.479	0.428	0.198
BEVStereo [13]	ResNet50	256×704	37.2	50.0	0.598	0.270	0.438	0.367	0.190
BEVFormerv2 [41]†	ResNet50	-	42.3	52.9	0.618	0.273	0.413	0.333	0.188
SOLOFusion [26]	ResNet50	256×704	42.7	53.4	0.567	0.274	0.511	0.252	0.181
SparseBEV [19]†	ResNet50	256×704	44.8	55.8	0.581	0.271	0.373	0.247	0.190
StreamPETR [34]†	ResNet50	256×704	45.0	55.0	0.613	0.267	0.413	0.265	0.198
RayDN† (Ours)	ResNet50	256×704	**46.9**	**56.3**	0.579	0.264	0.433	0.256	0.187
BEVDepth [14]	ResNet101	512×1408	41.2	53.5	0.565	0.266	0.358	0.331	0.190
PETRv2 [22]†	ResNet101	640×1600	42.1	52.4	0.681	0.267	0.357	0.377	0.186
Sparse4D [17]†	ResNet101	900×1600	43.6	54.1	0.633	0.279	0.363	0.317	0.177
SOLOFusion [26]	ResNet101	512×1408	48.3	58.2	0.503	0.264	0.381	0.246	0.207
SparseBEV [19]†	ResNet101	512×1408	50.1	59.2	0.562	0.265	0.321	0.243	0.195
StreamPETR [34]†	ResNet101	512×1408	50.4	59.2	0.569	0.262	0.315	0.257	0.199
RayDN† (Ours)	ResNet101	512×1408	**51.8**	**60.4**	0.541	0.260	0.315	0.236	0.200

Table 2: Comparison on the nuScenes test set.

Methods	Backbone	Image Size	mAP	NDS	mATE	mASE	mAOE	mAVE	mAAE
DETR3D [37]	V2-99	900×1600	41.2	47.9	0.641	0.255	0.394	0.845	0.133
MV2D [38]	V2-99	640×1600	46.3	51.4	0.542	0.247	0.403	0.857	0.127
BEVFormer [15]	V2-99	900×1600	48.1	56.9	0.582	0.256	0.375	0.378	0.126
PETRv2 [22]	V2-99	640×1600	49.0	58.2	0.561	0.243	0.361	0.343	0.120
PolarFormer [10]	V2-99	900×1600	49.3	57.2	0.556	0.256	0.364	0.439	0.127
BEVStereo [13]	V2-99	900×1600	52.5	61.0	0.431	0.246	0.358	0.357	0.138
HoP [47]	V2-99	640×1600	52.8	61.2	0.491	0.242	0.332	0.343	0.109
SparseBEV [19]	V2-99	640×1600	54.3	62.7	0.502	0.244	0.324	0.251	0.126
StreamPETR [34]	V2-99	640×1600	55.0	63.6	0.479	0.239	0.317	0.241	0.119
RayDN (Ours)	V2-99	640×1600	**56.5**	**64.5**	0.461	0.241	0.322	0.239	0.114

4.2 Implementation Details

Our experimental setup leverages ResNet50 [7], ResNet101, and V2-99 [11] backbones, each with distinct pre-training configurations. We detail the performance of ResNet50 and ResNet101 models, which are pre-trained on nuImages [2], on the nuScenes validation set. To showcase the scalability of our approach, we also report results on the nuScenes test set using V2-99, initialized from the DD3D [25] checkpoint. Our models are optimized using the AdamW [23] optimizer with a batch size of 16. The learning rate is set to 4×10^{-4} for models trained only on the training set and 3×10^{-4} for those trained on both the training and validation sets. We adopt a cosine annealing policy for learning rate scheduling. For benchmarking against state-of-the-art (SOTA) methods, models are trained for 60 epochs without CBGS [45] and for 24 epochs in ablation studies in nuScenes. For experiments on Argoverse 2, models are trained for 6 epochs. Our implementation is primarily based on the StreamPETR [34] framework.

Table 3: Comparisons on the Argoverse 2 validation set. We evaluate across 26 object categories within a range of 150 meters.

Methods	Backbone	Image Size	mAP	CDS	mATE	mASE	mAOE
PETR [21]	V2-99	900×640	17.6	12.2	0.911	0.339	0.819
Sparse4Dv2 [18]	V2-99	900×640	18.9	13.4	0.832	0.343	0.723
StreamPETR [34]	V2-99	900×640	20.3	14.6	0.843	0.321	0.650
RayDN (Ours)	V2-99	900×640	**22.3**	**16.1**	0.825	0.325	0.629

4.3 Comparison with State-of-the-Art Methods

We compare the proposed Ray Denoising method with other state-of-the-art multi-view 3D object detectors on the validation and test sets of the nuScenes dataset, as well as the validation set of the Argoverse 2 dataset. It's important to note that our method does not employ test time augmentation (TTA).

nuScenes Validation Set. Table 1 presents a comparison with state-of-the-art methods on the nuScenes validation set. We evaluate both ResNet-50 and ResNet-101 backbones. With ResNet-50 as the backbone and an image size of 704×256, we achieve 46.9% mAP and 56.3% NDS, improving upon the previous state-of-the-art method, StreamPETR, by 1.9% mAP and 1.3% NDS. Using a more robust ResNet-101 backbone and increasing the image size to 512×1408, our performance reaches 51.6% mAP and 59.8% NDS, outperforming StreamPETR by 1.2% mAP and 0.6% NDS. These results demonstrate the scalability of Ray Denoising.

nuScenes Test Set. The results evaluated by the test server are detailed in Table 2. Our method includes training on both the training and validation sets. Notably, we achieve 56.5% mAP and 64.5% NDS, surpassing StreamPETR by an absolute 1.5% mAP and 0.9% NDS.

Argoverse 2 Validation Set. To assess the generalization capability of Ray Denoising, we conduct additional experiments on the Argoverse 2 dataset, as displayed in Table 3. Our method significantly outperforms previous state-of-the-art methods, achieving a 2.0% mAP and 1.5% CDS improvement on the validation set. These metrics underscore the generalization capability of our approach.

The tables demonstrate that our Ray Denoising approach notably enhances mAP. Considering that mAP is significantly impacted by incorrect false positives, this enhancement firmly validates the effectiveness of Ray Denoising in reducing redundant predictions along the camera rays.

4.4 Ablation Study

This section delves into the ablation studies performed using the validation sets from the nuScenes and Argoverse 2 datasets. Unless specified otherwise, our experiments on nuScenes are based on a ResNet50 backbone, which is pre-trained on the nuImages dataset [2]. Our input data comprises 8 frames, each with a 704×256 pixels resolution. The decoder utilizes 428 queries, and the model is trained for 24 epochs, forgoing Class-Balanced Group Sampling (CBGS) [45].

Table 4: Ablation studies on the radius of ray denoising queries.

Method	Radius	mAP	NDS
SOTA Baseline [34]	-	39.1	48.6
+RayDN (Ours)	2	41.1	49.8
	3	**41.9**	**50.8**
	4	41.3	50.1

Table 5: Ablation studies on the number of ray denoising queries.

Method	#Q	mAP	NDS
SOTA Baseline [34]	-	39.1	48.6
+RayDN (Ours)	3	41.6	50.2
	5	**41.9**	**50.8**
	7	41.1	50.5

Table 6: Ablation studies on the distribution of ray denoising queries on nuScenes dataset.

Method	λ	μ	mAP	NDS
SOTA Baseline [34]	-	-	39.1	48.6
+RayDN (Ours)	1	1	41.3	50.4
	2	8	41.5	50.1
	3	7	41.3	50.5
	7	3	41.7	50.7
	8	2	**41.9**	**50.8**
	9	1	41.6	50.4

Table 7: Ablation studies on the distribution of ray denoising queries for the Argoverse 2 dataset.

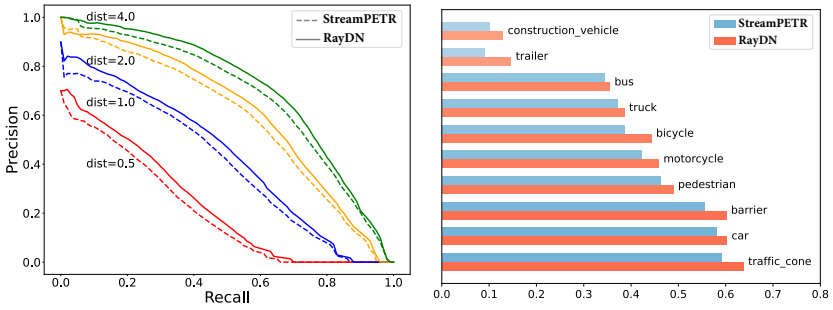
Method	λ	μ	mAP	CDS
SOTA Baseline [34]	-	-	20.3	14.6
+RayDN (Ours)	1	1	21.3	15.4
	3	7	21.7	15.8
	4	6	21.3	15.3
	6	4	21.5	15.7
	7	3	**22.3**	**16.1**
	8	2	21.5	15.6

Radius of Ray Denoising Queries. Table 4 shows how varying the radius k affects performance. We find that a radius of $k = 3$ yields the optimal results, while $k = 2$ and $k = 4$ also lead to notable enhancements in performance.

Number of Ray Denoising Queries. We explore the effect of the quantity of ray denoising queries on the model's performance in Table 5. The findings reveal that both mAP and NDS metrics improve with increasing ray queries, reaching a plateau of 5 queries. Surprisingly, upping the count to 7 queries results in a dip in mAP, which might be attributed to an imbalance in the ratio of positive to negative ray denoising queries.

Distribution of Ray Denoising Queries. We investigate the impact of the distribution of ray denoising queries with various hyper-parameters across the nuScenes and Argoverse 2 datasets. The results are detailed in Table 6 and Table 7. Our findings indicate that employing a uniform distribution, which is dataset-agnostic (*i.e.*, setting $\lambda = 1$ and $\mu = 1$), leads to substantial performance gains for Ray Denoising. Specifically, we observe a 2.2% increase in mAP for the nuScenes dataset and a 1.0% increase in mAP for the Argoverse 2 dataset, highlighting the method's strong generalization capabilities. Fine-tuning these hyper-parameters further enhances the performance, with an additional 0.6% mAP improvement on nuScenes and 1.0% mAP on Argoverse. Across both datasets, models configured with $\lambda > \mu$ slightly outperform those with $\lambda < \mu$, suggesting that false positives tend to be at greater distance from the ego-vehicle.

Precision-Recall Analysis. We assess the ability of Ray Denoising to reduce false positive predictions by plotting precision-recall curves and Average Precision (AP) for each object class. Figure 5a reveals that Ray Denoising enhances precision across a range of recall levels and under various distance thresholds.



(a) Precision-Recall Curve **(b)** AP for each class

Fig. 5: (a) Visualization of the precision-recall curves at various distance thresholds. Ray Denoising consistently enhances precision across nearly all recall levels, effectively suppressing false positives. (b) Class-wise AP comparison. Ray Denoising performs superior over the SOTA StreamPETR in all object classes. Best viewed in color.

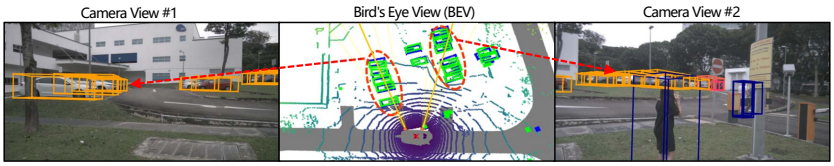


Fig. 6: Visualization of our detection results on the validation set of nuScenes. Ray Denoising effectively mitigates duplicate false positives while maintaining the ability to detect highly occluded objects along the same ray. Best viewed on the screen.

Similarly, Figure 5b shows that Ray Denoising surpasses StreamPETR in AP across all classes. These results collectively affirm that Ray Denoising effectively mitigates the occurrence of false positive predictions.

Visualization of Detection Results. Our approach creates multiple spatial hard negative samples along the same ray, potentially interfering with detecting true positives if they are also located along that ray. To test this, we present a visualization of the detection outcomes in Figure 6. The results show that Ray Denoising effectively eliminates duplicate false positives without compromising the detection of true positives, even when they are heavily occluded along the same ray.

Generalization Ability of Ray Denoising. To assess the versatility of Ray Denoising, we apply it to different models with varied input data. We use the state-of-the-art CMT [40] as a reference and test Ray Denoising with single-frame images and multi-modal data. In the single-frame scenario, we employ a ResNet50 backbone pre-trained on ImageNet [2], with input images sized at 256 $\times$ 704 pixels. For the multi-modal setup, the input image resolution is 320 $\times$ 800 pixels, and the voxel dimensions are 0.1m $\times$ 0.1m. The batch size is set to 16 due to memory constraints, and all models are trained for 20 epochs using CBGS [45]. Table 8 shows that Ray Denoising improves upon the baseline

Table 8: Ablation studies on generalization ability of Ray Denoising.

Method	Description	mAP	NDS
CMT [40]	single frame	30.6	37.7
+ RayDN	single frame	**32.1**	**39.1**
CMT [40]	multi-modal	67.6	70.4
+ RayDN	multi-modal	**68.6**	**71.3**

Table 9: Ablation studies on the effect of depth estimation.

3DPPE	RayDN	mAP	NDS
		39.1	48.6
✓		42.0	51.1
	✓	41.9	50.8
✓	✓	**43.9**	**52.9**

with a 1.5% increase in mAP and a 1.4% increase in NDS for the single-frame setup. Moreover, it achieves a 1.0% mAP and a 0.9% NDS enhancement on the already high-performing multi-modal baseline. These experiments confirm the broad applicability of Ray Denoising.

Effect of Depth Estimation on Ray Denoising. In DETR-style multi-view 3D object detectors, the lack of depth information leads to the use of camera rays as position embeddings for image features, which can be imprecise. To refine these embeddings, 3DPPE [30] suggests encoding an estimated 3D point guided by external supervision for depth estimation. To explore the role of depth estimation in conjunction with Ray Denoising, we integrate 3DPPE into our framework. The results, detailed in Table 9, show a marked improvement in performance with 3DPPE. However, Ray Denoising alone, without external depth supervision, also achieves similar performance. This underscores our core insight: Ray Denoising enhances the model's depth perception by guiding it to learn superior depth perception ability during training. When Ray Denoising is paired with 3DPPE, we further obtain a +1.9% increase in mAP and a +1.7% increase in NDS. This demonstrates that while depth estimation algorithms can improve the clarity of position embeddings, Ray Denoising's ability to refine the model's understanding of depth through robust training is a powerful complement, even when depth information is available.

5 Conclusion

We introduce Ray Denoising, a method designed to overcome the critical challenge of depth estimation inaccuracy in multi-view 3D object detection. Ray Denoising tackles the issue of false detections along camera rays, which are a direct consequence of imprecise depth information from images. By leveraging the 3D structure of the scene, Ray Denoising prompts the model to learn depth-aware features, leading to improved differentiation between false and true positives along the same ray without introducing extra inference costs. Our comprehensive experiments on the NuScenes and Argoverse 2 datasets demonstrate that Ray Denoising consistently and significantly outperforms strong baselines, achieving new state-of-the-art performance in Multi-view 3D Object Detection.

Acknowledgment. This work was supported by the Fundamental Research Funds for the Central Universities (E3E41903, E2ET1104, E3ET6201X2), the National Natural Science Foundation of China (NSFC) under Grant 62225208 and 62171431.

References

1. Brazil, G., Liu, X.: M3d-rpn: Monocular 3d region proposal network for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9287–9296 (2019)
2. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
4. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE transactions on pattern analysis and machine intelligence **32**(9), 1627–1645 (2009)
5. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013)
6. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Region-based convolutional networks for accurate object detection and segmentation. IEEE transactions on pattern analysis and machine intelligence **38**(1), 142–158 (2015)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
8. Huang, J., Huang, G.: Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. arXiv preprint arXiv:2203.17054 (2022)
9. Huang, J., Huang, G., Zhu, Z., Ye, Y., Du, D.: Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. arXiv preprint arXiv:2112.11790 (2021)
10. Jiang, Y., Zhang, L., Miao, Z., Zhu, X., Gao, J., Hu, W., Jiang, Y.G.: Polarformer: Multi-camera 3d object detection with polar transformer. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 1042–1050 (2023)
11. Lee, Y., Hwang, J.w., Lee, S., Bae, Y., Park, J.: An energy and gpu-computation efficient backbone network for real-time object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. pp. 0–0 (2019)
12. Li, F., Zhang, H., Liu, S., Guo, J., Ni, L.M., Zhang, L.: Dn-detr: Accelerate detr training by introducing query denoising. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13619–13627 (2022)
13. Li, Y., Bao, H., Ge, Z., Yang, J., Sun, J., Li, Z.: Bevstereo: Enhancing depth estimation in multi-view 3d object detection with temporal stereo. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 1486–1494 (2023)
14. Li, Y., Ge, Z., Yu, G., Yang, J., Wang, Z., Shi, Y., Sun, J., Li, Z.: Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 1477–1485 (2023)
15. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Qiao, Y., Dai, J.: Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In: European conference on computer vision. pp. 1–18. Springer (2022)
16. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)

17. Lin, X., Lin, T., Pei, Z., Huang, L., Su, Z.: Sparse4d: Multi-view 3d object detection with sparse spatial-temporal fusion. arXiv preprint arXiv:2211.10581 (2022)
18. Lin, X., Lin, T., Pei, Z., Huang, L., Su, Z.: Sparse4d v2: Recurrent temporal fusion with sparse model. arXiv preprint arXiv:2305.14018 (2023)
19. Liu, H., Teng, Y., Lu, T., Wang, H., Wang, L.: Sparsebev: High-performance sparse 3d object detection from multi-camera videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18580–18590 (2023)
20. Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., Zhu, J., Zhang, L.: Dab-detr: Dynamic anchor boxes are better queries for detr. arXiv preprint arXiv:2201.12329 (2022)
21. Liu, Y., Wang, T., Zhang, X., Sun, J.: Petr: Position embedding transformation for multi-view 3d object detection. In: European Conference on Computer Vision. pp. 531–548. Springer (2022)
22. Liu, Y., Yan, J., Jia, F., Li, S., Gao, A., Wang, T., Zhang, X.: Petrv2: A unified framework for 3d perception from multi-camera images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3262–3272 (2023)
23. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2018)
24. Ma, X., Zhang, Y., Xu, D., Zhou, D., Yi, S., Li, H., Ouyang, W.: Delving into localization errors for monocular 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4721–4730 (2021)
25. Park, D., Ambrus, R., Guizilini, V., Li, J., Gaidon, A.: Is pseudo-lidar needed for monocular 3d object detection? In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3142–3152 (2021)
26. Park, J., Xu, C., Yang, S., Keutzer, K., Kitani, K.M., Tomizuka, M., Zhan, W.: Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. In: The Eleventh International Conference on Learning Representations (2022)
27. Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16. pp. 194–210. Springer (2020)
28. Reading, C., Harakeh, A., Chae, J., Waslander, S.L.: Categorical depth distribution network for monocular 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8555–8564 (2021)
29. Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 761–769 (2016)
30. Shu, C., Deng, J., Yu, F., Liu, Y.: 3dppe: 3d point positional encoding for transformer-based multi-camera 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3580–3589 (October 2023)
31. Simonelli, A., Bulo, S.R., Porzi, L., López-Antequera, M., Kontschieder, P.: Disentangling monocular 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1991–1999 (2019)
32. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2446–2454 (2020)

33. Sung, K.K., Poggio, T.: Example-based learning for view-based human face detection. IEEE Transactions on pattern analysis and machine intelligence **20**(1), 39–51 (1998)
34. Wang, S., Liu, Y., Wang, T., Li, Y., Zhang, X.: Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3621–3631 (October 2023)
35. Wang, T., Zhu, X., Pang, J., Lin, D.: Fcos3d: Fully convolutional one-stage monocular 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 913–922 (2021)
36. Wang, Y., Zhang, X., Yang, T., Sun, J.: Anchor detr: Query design for transformer-based detector. In: Proceedings of the AAAI conference on artificial intelligence. vol. 36, pp. 2567–2575 (2022)
37. Wang, Y., Guizilini, V.C., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In: Conference on Robot Learning. pp. 180–191. PMLR (2022)
38. Wang, Z., Huang, Z., Fu, J., Wang, N., Liu, S.: Object as query: Lifting any 2d object detector to 3d detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3791–3800 (2023)
39. Wilson, B., Qi, W., Agarwal, T., Lambert, J., Singh, J., Khandelwal, S., Pan, B., Kumar, R., Hartnett, A., Pontes, J.K., et al.: Argoverse 2: Next generation datasets for self-driving perception and forecasting. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2) (2021)
40. Yan, J., Liu, Y., Sun, J., Jia, F., Li, S., Wang, T., Zhang, X.: Cross modal transformer: Towards fast and robust 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18268–18278 (2023)
41. Yang, C., Chen, Y., Tian, H., Tao, C., Zhu, X., Zhang, Z., Huang, G., Li, H., Qiao, Y., Lu, L., et al.: Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17830–17839 (2023)
42. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L., Shum, H.Y.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In: The Eleventh International Conference on Learning Representations (2022)
43. Zhang, H., Li, H., Zeng, A., Li, F., Liu, S., Liao, X., Zhang, L.: Introducing depth into transformer-based 3d object detection. arXiv preprint arXiv:2302.13002 (2023)
44. Zhang, Y., Lu, J., Zhou, J.: Objects are different: Flexible monocular 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3289–3298 (2021)
45. Zhu, B., Jiang, Z., Zhou, X., Li, Z., Yu, G.: Class-balanced grouping and sampling for point cloud 3d object detection. arXiv preprint arXiv:1908.09492 (2019)
46. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)
47. Zong, Z., Jiang, D., Song, G., Xue, Z., Su, J., Li, H., Liu, Y.: Temporal enhanced training of multi-view 3d object detector via historical object prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3781–3790 (October 2023)