This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

FocalFormer3D : Focusing on Hard Instance for 3D Object Detection

Yilun Chen[1*] Zhiding Yu[3†] Yukang Chen[1]
Shiyi Lan[3] Anima Anandkumar[2,3] Jiaya Jia[1] Jose M. Alvarez[3]
[1]The Chinese University of Hong Kong [2]Caltech [3]NVIDIA

Abstract

False negatives (FN) in 3D object detection, e.g.*, missing predictions of pedestrians, vehicles, or other obstacles, can lead to potentially dangerous situations in autonomous driving. While being fatal, this issue is understudied in many current 3D detection methods. In this work, we propose Hard Instance Probing (HIP), a general pipeline that identifies FN in a multi-stage manner and guides the models to focus on excavating difficult instances. For 3D object detection, we instantiate this method as FocalFormer3D, a simple yet effective detector that excels at excavating difficult objects and improving prediction recall. FocalFormer3D features a multi-stage query generation to discover hard objects and a box-level transformer decoder to efficiently distinguish objects from massive object candidates. Experimental results on the nuScenes and Waymo datasets validate the superior performance of FocalFormer3D. The advantage leads to strong performance on both detection and tracking, in both LiDAR and multi-modal settings. Notably, FocalFormer3D achieves a 70.5 mAP and 73.9 NDS on nuScenes detection benchmark, while the nuScenes tracking benchmark shows 72.1 AMOTA, both ranking 1st place on the nuScenes LiDAR leaderboard. Our code is available at `https://github.com/NVlabs/FocalFormer3D`.*

1. Introduction

3D object detection is an important yet challenging perception task. Recent state-of-the-art 3D object detectors mainly rely on bird's eye view (BEV) representation [1–3], where features from multiple sensors are aggregated to construct a unified representation in the ego-vehicle coordinate space. There is a rich yet growing literature on BEV-based 3D detection, including multi-modal fusion [4–10], second-stage refinements (surface point pooling [3], RoIPool [11–14], and cross attention modules [4, 15]).

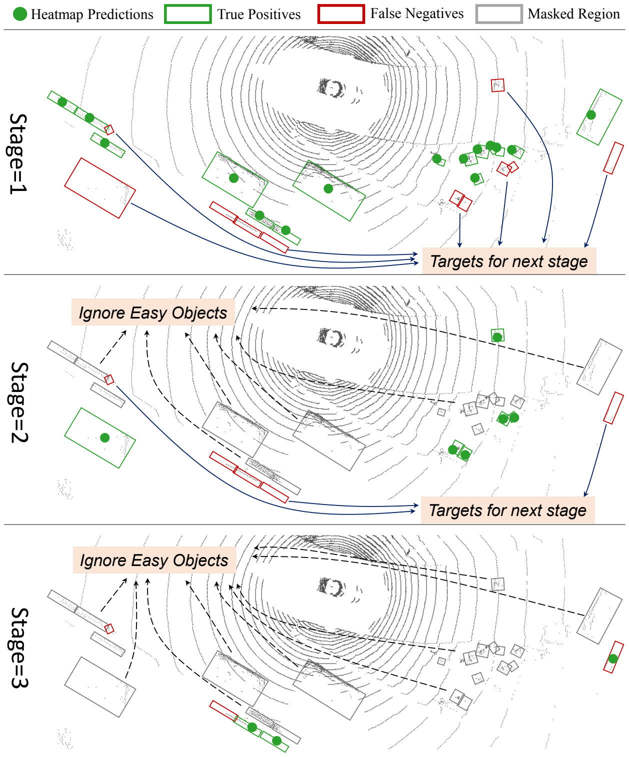

Figure 1. **Visual example for Hard Instance Probing (HIP).** By utilizing this multi-stage prediction approach, our model can progressively focus on hard instances and facilitate its ability to gradually detect them. At each stage, the model generates some *Positive* object candidates (represented by green circles). Object candidates assigned to the ground-truth objects can be classified as either *True Positives* (*TP*, represented by green boxes) and *False Negatives* (*FN*, represented by red boxes) during training. We explicitly model the unmatched ground-truth objects as the hard instances, which become the main targets for the subsequent stage. Conversely, *Positives* are considered easy samples (represented by gray boxes) and will be ignored in subsequent stages at both training and inference time. At last, all heatmap predictions across stages are collected as the initial object candidates. We ignored the *False Positives* for better visualizations.

Despite the tremendous efforts, there has been limited exploration to explicitly address *false negatives or missed*

[*] Work done during an internship at NVIDIA.
[†] Corresponding author.

objects often caused by occlusions and clutter background. False negatives are particularly concerning in autonomous driving as they cause missing information in the prediction and planning stacks. When an object or a part of an object is not detected, this can result in the autonomous vehicle being unaware of potential obstacles such as pedestrians, cyclists, or other vehicles. This is especially hazardous when the vehicle is moving at high speeds and can lead to potentially dangerous situations. Therefore, reducing false negatives is crucial to ensure the safety of autonomous driving.

To address the challenge of *False Negative*s in 3D detection, we propose and formulate a pipeline called *Hard Instance Probing* (HIP). Motivated by cascade-style decoder head for object detection [16–18], we propose a pipeline to probe false negative samples progressively, which significantly improves the recall rate Fig. 1 illustrates the pipeline in a cascade manner. In each stage, HIP suppresses the true positive candidates and focuses on the false negative candidates from the previous stages. By iterating the HIP stage, our approach can save those hard false negatives.

Based on HIP, we introduce a 3D object detector, FocalFormer3D, as shown in Fig. 2. Especially, multi-stage heatmap predictions [3, 19] are employed to excavate difficult instances. We maintain a class-aware *Accumulated Positive Mask*, indicating positive regions from prior stages. Through this masking design, the model omits the training of easy positive candidates and thereby focuses on the hard instances (*False Negatives*). Finally, our decoder collects the positive predictions from all stages to produce the object candidates. FocalFormer3D consistently demonstrates considerable gains over baselines in terms of average recall.

In addition, we also introduce a box-level refinement step to eliminate redundant object candidates. The approach employs a deformable transformer decoder [17] and represents the candidates as box-level queries using RoIAlign. This allows for box-level query interaction and iterative box refinements, binding the object queries with sufficient box context through RoIAlign [20, 21] on the bird's eye view to perform relative bounding box refinements. Finally, a rescoring strategy is adopted to select positive objects from object candidates. Our ablation study in Table 6 demonstrates the effectiveness of the local refinement approach in processing adequate object candidates.

Our contributions can be summarized as follows:

- We propose Hard Instance Probing (HIP), a learnable scheme to automatically identify *False Negatives* in a multi-stage manner.
- We present FocalFormer3D for 3D object detection that effectively harvests hard instances on the BEV and demonstrates effectiveness in terms of average recall.
- Without bells and whistles, our model achieves state-of-the-art detection performance on **both** LiDAR-based and multi-modal settings. Notably, our model ranks **1st** places on **both** nuScenes 3D LiDAR detection and tracking leaderboard at time of submission.

2. Related Work

Modern 3D object detectors, either LiDAR-based [1–3, 12, 13, 22–29], or Camera-based [30–37], or Multi-Modal [4–8, 38–45] 3D object detectors generally rely on BEV view representation [46]. These methods adopt dense feature maps or dense anchors, for conducting object prediction in a bird's eye view (BEV) space. Among these methods, VoxelNet [22] as the pioneer works discretize point clouds into voxel representation and applies dense convolution to generate BEV heatmaps. SECOND [22] accelerates VoxelNet with 3D sparse convolution [47] to extract 3D features. Some Pillar-based detectors [2, 23, 48, 49] collapse the height dimension and utilize 2D CNNs for efficient 3D detection.

Different from dense detectors, point-based 3D detectors [11, 50–52] directly process point clouds via PointNet [53, 54] and perform grouping or predictions on the sparse representations. Concerning involvement of neighborhood query on point clouds, it becomes time-consuming and unaffordable for large-scale point clouds. Concerning computation and spatial cost, another line of 3D detectors directly predicts objects on sparse point clouds to avoid dense feature construction. SST [55] applies sparse regional attention and avoids downsampling for small-object detection. FSD [56] instead further recognize instances directly on sparse representations obtained by SST [55] and SparseConv for long-range detection.

Recent multi-modal detectors [5–7, 39, 42, 57] follow the similar paradigm of BEV detectors and incorporate the multi-view image features by physical projection or learnable alignments between LiDAR and cameras. TransFusion [4] applies cross attention to obtain image features for each object query. Despite various kinds of modal-specific voxel feature encoders, these detectors finally produce dense BEV features for classification and regression at the heatmap level.

3. Methodology

We introduce Hard Instance Probing (HIP) for automated identifying hard instances (*False Negatives*) in Section 3.1. We then present the implementations for the two main components of FocalFormer3D. Section 3.2 describes our multi-stage heatmap encoder that harvests the *False Negative*s for producing high-recall initial object candidates following HIP. Section 3.3 introduces a box-level deformable decoder network that further distinguishes objects from these candidates.

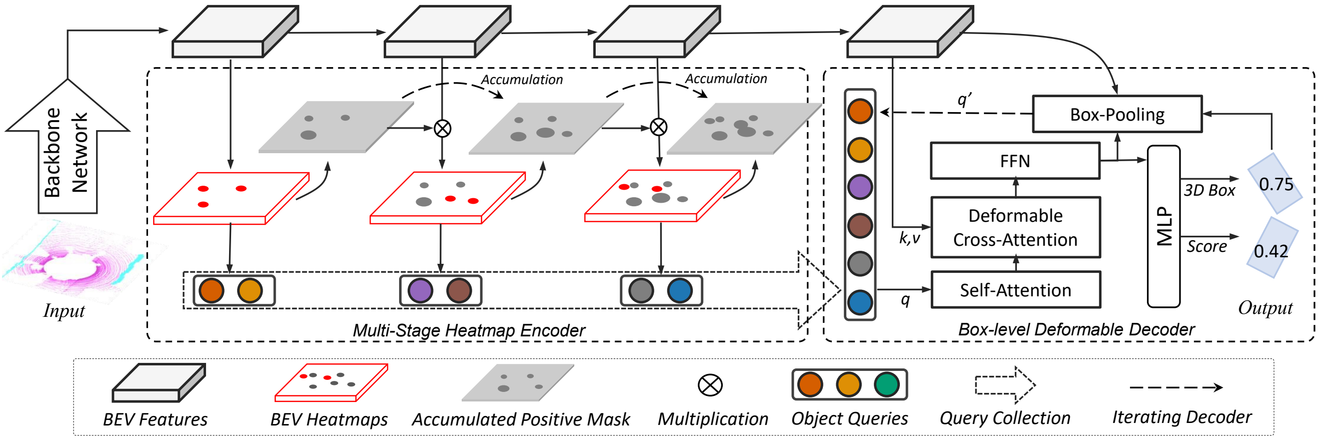

Figure 2. **Overall architecture of FocalFormer3D**. The overall framework comprises two novel components: a multi-stage heatmap encoder network that uses the Hard Instance Probing (HIP) strategy to produce high-recall object queries (candidates), and a deformable transformer decoder network with rescoring mechanism that is responsible for eliminating false positives from the large set of candidates. (a) Following feature extraction from modalities, the map-view features produce a set of multi-stage BEV features and then BEV heatmaps. The positive mask accumulates to exclude the easy positive candidates of prior stages from BEV heatmaps. The left object candidates are chosen and collected according to the response of BEV heatmap in a multi-stage process. (b) A deformable transformer decoder is adapted to effectively handle diverse object queries. The query embedding is enhanced with a box pooling module, which leverages the intermediate object supervision to identify local regions. It refines object queries in a local-scope manner, rather than at a point level. Residual connections and normalization layers have been excluded from the figure for clarity.

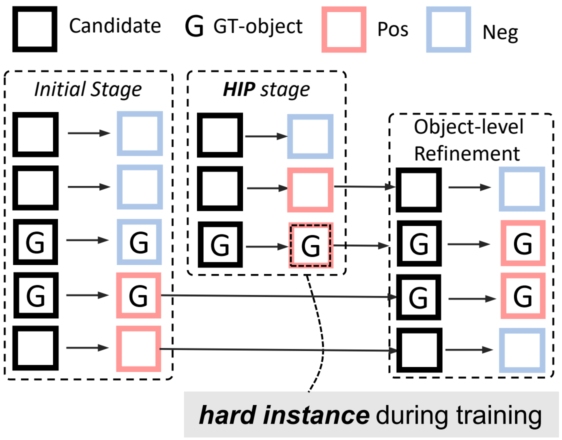

Figure 3. **Hard Instance Probing.** We use the symbol "G" to indicate the object candidates that are labeled as ground-truth objects during the target assignment process in training. To ensure clarity, we omit numerous negative predictions for detection, given that background takes up most of the images.

3.1. Hard Instance Probing (HIP)

Real-world applications, such as autonomous driving, require a high level of scene understanding to ensure safe and secure operation. In particular, false negatives in object detection can present severe risks, emphasizing the need for high recall rates. However, accurately identifying objects in complex scenes or when occlusion occurs is challenging in 3D object detection, resulting in many false negative predictions. Unfortunately, few studies have explicitly focused on addressing false negatives in the design of detection heads. Motivated by the cascade-style detectors, we formulate a training pipeline to emulate the process of identifying false negative predictions at inference time.

Formulation of Hard Instance Probing. Our strategy to identify hard instances operates stage by stage, as illustrated by a toy example in Fig. 3. Initially, we annotate the ground-truth objects as

$$\mathcal{O} = \{o_i, i = 1, 2, ...\},$$

which is the main targets for initial stages. The neural network makes *Positive* or *Negative* predictions given a set of initial object candidates $\mathcal{A} = \{a_i, i = 1, 2, ...\}$, which is not limited to anchors [58], point-based anchors [3], and object queries [59]. Suppose the detected objects (*Positive* predictions) at k-th stage are

$$\mathcal{P}_k = \{p_i, i = 1, 2, ...\}.$$

We are then allowed to classify the ground-truth objects according to their assigned candidates:

$$\mathcal{O}_k^{TP} = \{o_j | \exists p_i \in \mathcal{P}_k, \sigma(p_i, o_j) > \eta\}.$$

where an object matching metric $\sigma(\cdot, \cdot)$ (e.g. Intersection over Union [60, 61] and center distance [62]) and a predefined threshold η. Thus, the left unmatched targets can be regarded as hard instances:

$$\mathcal{O}_k^{FN} = O - \bigcup_{i=1}^{k} O_k^{TP}.$$

The training of $(k+1)$-th stages is to detect these targets $\mathcal{O}_k^{FN}$ from the object candidates while omitting all prior *Positive* object candidates.

Despite the cascade way mimicking the process of identifying false negative samples, we might collect a number of object candidates across all stages. Thus, a second-stage object-level refinement model is necessary to eliminate any potential false positives.

Relation with hard example mining. The most relevant topic close to our approach is hard example mining [63, 64], which samples hard examples during training. Recent research [65–67] has further explored soft-sampling, such as adjusting the loss distribution to mitigate foreground-background imbalance issues. In contrast, our method operates in stages. Specifically, we use *False Negative* predictions from prior stages to guide the subsequent stage of the model toward learning from these challenging objects.

3.2. Multi-stage Heatmap Encoder

The upcoming subsections outline the key implementations of FocalFormer3D as depicted in Fig. 2. We begin by detailing the implementation of hard instance probing for BEV detection. This involves using the BEV center heatmap to generate the initial object candidate in a cascade manner.

Preliminary of center heatmap in BEV perception. In common practice [3, 4, 19], the objective of the BEV heatmap head is to produce heatmap peaks at the center locations of detected objects. The BEV heatmaps are represented by a tensor $S \in \mathbb{R}^{X \times Y \times C}$, where $X \times Y$ indicates the size of BEV feature map and C is the number of object categories. The target is achieved by producing 2D Gaussians near the BEV object points, which are obtained by projecting 3D box centers onto the map view. In top views such as Fig. 4, objects are more sparsely distributed than in a 2D image. Moreover, it is assumed that objects do not have intra-class overlaps on the bird's eye view.

Based on the non-overlapping assumption, excluding prior easy positive candidates from BEV heatmap predictions can be achieved easily. In the following, we illustrate the implementation details of HIP, which utilizes an accumulated positive mask.

Positive mask accumulation. To keep track of all easy positive object candidates of prior stages, we generate a positive mask (PM) on the BEV space for each stage and accumulated them to an accumulated positive mask (APM):

$$\hat{M}_k \in \{0,1\}^{X \times Y \times C},$$

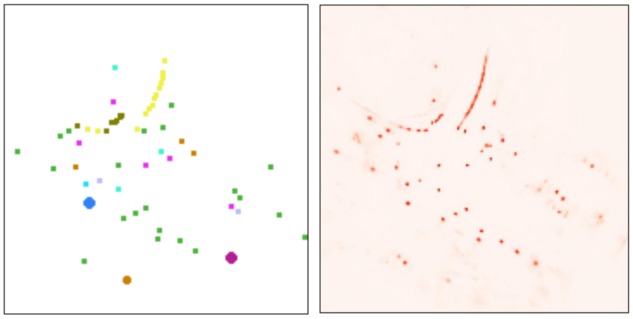

Figure 4. **Example visualization for the positive mask. (left) and predicted BEV heatmap (right)**. The positive mask is class-aware and we show different categories with different colors for visualization. The masking area for objects of different categories can differ in the pooling-based masking method.

which is initialized as all zeros.

The generation of multi-stage BEV features is accomplished in a cascade manner using a lightweight inversed residual block [68] between stages. Multi-stage BEV heatmaps are generated by adding an extra convolution layer. At each stage, we generate the positive mask according to the positive predictions. To emulate the process of identifying *False Negative*s, we use a test-time selection strategy that ranks the scores according to BEV heatmap response [3, 4]. Specifically, at the k-th stage, Top-K selection is performed on the BEV heatmap across all BEV positions and categories, producing a set of object predictions $\mathcal{P}_k$. Then the positive mask $M_k \in \{0,1\}^{X \times Y \times C}$ records the all the positions of positive predictions by setting $M_{(x,y,c)} = 1$ for each predicted object $p_i \in \mathcal{P}_k$, where (x,y) represents p_i's location and c is p_i's class. The left points are set to 0 by default.

According to the non-overlapping assumption, the ideal way to indicate the existence of a positive object candidate (represented as a point in the center heatmap) on the mask is by masking the box if there is a matched ground truth box. However, since the ground-truth boxes are not available at inference time, we propose the following masking methods during training:

- **Point Masking**. This method involves no change, where only the center point of the positive candidates is filled.

- **Pooling-based Masking**. In this method, smaller objects fill in the center points while larger objects fill in with a kernel size of 3 × 3.

- **Box Masking**. This method requires an additional box prediction branch and involves filling the internal region of the predicted BEV box.

The accumulated positive mask (APM) for the k-th stage is obtained by simply accumulating prior Positive Masks as follows:

$$\hat{M}_k = \max_{1 \le i \le k} M_i.$$

By masking the BEV heatmap S_k with

$$\hat{S}_k = S_k \cdot (1 - \hat{M}_k),$$

we omit prior easy positive regions in the current stage, thus enabling the model to focus on the false negative samples of the prior stage (hard instances). To train the multi-stage heatmap encoder, we adopt Gaussian Focal Loss [4] as the training loss function. We sum up the BEV heatmap losses across stages to obtain the final heatmap loss.

During both training and inference, we collect the positive candidates from all stages as the object candidates for the second-stage rescoring as the potential false positive predictions.

Discussion on implementation validity for HIP. Although the HIP strategy is simple, the masking way has two critical criteria that need to be met to ensure valid implementation of HIP:

- Exclusion of prior positive object candidates at the current stage.
- Avoidance of removal of potential real objects (false negatives).

Point masking satisfies both requirements based on the following facts. As the Top-K selection is based on ranking predicted BEV heatmap scores, the hottest response points are automatically excluded when a point is masked. Besides, the design of a class-aware positive mask ensures that non-overlapping assumptions at the intra-class level on the BEV are met.

However, the point masking strategy is less efficient as only one BEV object candidate is excluded for each positive prediction compared with the ideal masking with ground-truth box guidance. Therefore, there is a trade-off between the masking area and the validity of the exclusion operation. We compare all three strategies in Table 5 and pooling-based masking performs better than others.

3.3. Box-level Deformable Decoder

The object candidates obtained from the multi-stage heatmap encoder can be treated as positional object queries [4, 69]. The recall of initial candidates improves with an increase in the number of collected candidates. However, redundant candidates introduce false positives, thereby necessitating a high level of performance for the following object-level refinement blocks.

To enhance the efficiency of object query processing, we employ deformable attention [17] instead of computationally intensive modules such as cross attention [59] or box attention [70]. Unlike previous methods that used center point features as the query embedding [4, 69], we model the object candidates as box-level queries. Specifically, Specifically, we introduce object supervision between deformable decoder layers, facilitating relative box prediction.

Box-pooling module. To better model the relations between objects and local regions in the regular grid manner, we extract the box context information from the BEV features using simple RoIAlign [20] in the Box-pooling module as Fig. 2. In specific, given the intermediate predicted box, each object query extracts 7×7 feature grid points [20] from the BEV map followed by two MLP layers. The positional encoding is also applied both for queries and all BEV points for extracting positional information. This allows us to update both the content and positional information into the query embedding. This lightweight module enhances the query feature for the deformable decoder (See Table 6).

Decoder implementation. Following Deformable DETR [17], our model employs 8 heads in all attention modules, including multi-head attention and multi-head deformable attention. The deformable attention utilizes 4 sampling points across 3 scales. To generate three scales of BEV features, we apply $2\times$ and $4\times$ downsampling operations to the original BEV features. The box-pooling module extracts 7×7 feature grid points within each *rotated* BEV box followed by 2 FC layers and adds the object feature to query embedding. We expand the predicted box to $1.2\times$ size of its original size.

3.4. Model Training

The model is trained in two stages. In the first stage, we train the LiDAR backbone using a deformable transformer decoder head, which we refer to as DeformFormer3D (Table 4 (a)). After initializing the weights from DeformFormer3D, we train the FocalFormer3D detector, which consists of a multi-stage heatmap encoder and a box-level deformable decoder. However, during the training of the deformable decoder with bipartite graph matching, we encounter slow convergence issues in the early stages [18]. To address this, we generate noisy queries from ground-truth objects [18, 77, 78], enabling effective training of the model from scratch. Additionally, we improve the training process by excluding matching pairs with a center distance between the prediction and its GT object exceeding 7 meters.

4. Experiments

4.1. Experimental Setup

Dataset and metric. We evaluate our approach on *nuScenes* and *Waymo* 3D detection dataset.

nuScenes Dataset [62] is a large-scale outdoor dataset. nuScenes contains $1,000$ scenes of multi-modal data, in-

Methods	Modality	mAP	NDS	Car	Truck	C.V.	Bus	Trailer	Barrier	Motor.	Bike	Ped.	T.C.
LiDAR-based 3D Detection													
PointPillars [2]	L	30.5	45.3	68.4	23.0	4.1	28.2	23.4	38.9	27.4	1.1	59.7	30.8
CBGS [71]	L	52.8	63.3	81.1	48.5	10.5	54.9	42.9	65.7	51.5	22.3	80.1	70.9
LargeKernel3D [28]	L	65.3	70.5	85.9	55.3	26.8	66.2	60.2	74.3	72.5	46.6	85.6	80.0
TransFusion-L [4]	L	65.5	70.2	86.2	56.7	28.2	66.3	58.8	78.2	68.3	44.2	86.1	82.0
PillarNet-34 [48]	L	66.0	71.4	87.6	**57.5**	27.9	63.6	63.1	77.2	70.1	42.3	87.3	83.3
LiDARMultiNet [72]	L	67.0	71.6	86.9	57.4	31.5	64.7	61.0	73.5	75.3	47.6	87.2	**85.1**
FocalFormer3D	L	**68.7**	**72.6**	**87.2**	57.1	**34.4**	**69.6**	**64.9**	**77.8**	**76.2**	**49.6**	**88.2**	82.3
CenterPoint [3] †	L	60.3	67.3	85.2	53.5	20.0	63.6	56.0	71.1	59.5	30.7	84.6	78.4
MGTANet† [73]	L	67.5	72.7	88.5	59.8	30.6	67.2	61.5	66.3	75.8	52.5	87.3	**85.5**
LargeKernel3D‡ [28]	L	68.8	72.8	87.3	59.1	30.2	68.5	65.6	75.0	**77.8**	**53.5**	88.3	82.4
FocalFormer3D †	L	**70.5**	**73.9**	**87.8**	**59.4**	**37.8**	**73.0**	**65.7**	**77.8**	77.4	52.4	**90.0**	83.4
Multi-Modal 3D Detection													
PointPainting [74]	L+C	46.4	58.1	77.9	35.8	15.8	36.2	37.3	60.2	41.5	24.1	73.3	62.4
3D-CVF [75]	L+C	52.7	62.3	83.0	45.0	15.9	48.8	49.6	65.9	51.2	30.4	74.2	62.9
MVP [41]	L+C	66.4	70.5	86.8	58.5	26.1	67.4	57.3	74.8	70.0	49.3	89.1	85.0
FusionPainting [76]	L+C	68.1	71.6	87.1	60.8	30.0	68.5	61.7	71.8	74.7	53.5	88.3	85.0
TransFusion [4]	L+C	68.9	71.7	87.1	60.0	33.1	68.3	60.8	78.1	73.6	52.9	88.4	86.7
BEVFusion [5]	L+C	69.2	71.8	88.1	60.9	34.4	69.3	62.1	78.2	72.2	52.2	89.2	85.2
BEVFusion-MIT [6]	L+C	70.2	72.9	**88.6**	60.1	**39.3**	69.8	63.8	80.0	74.1	51.0	89.2	86.5
DeepInteraction [10]	L+C	70.8	73.4	87.9	60.2	37.5	70.8	63.8	**80.4**	75.4	54.5	**91.7**	**87.2**
FocalFormer3D	L+C	**71.6**	**73.9**	88.5	**61.4**	35.9	**71.7**	**66.4**	79.3	**80.3**	**57.1**	89.7	85.3
PointAugmenting [57] †	L+C	66.8	71.0	87.5	57.3	28.0	65.2	60.7	72.6	74.3	50.9	87.9	83.6
Focals Conv-F [27] ‡	L+C	70.1	73.6	87.5	60.0	32.6	69.9	64.0	71.8	81.1	59.2	89.0	85.5
LargeKernel3D-F [28] ‡	L+C	71.1	74.2	88.1	60.3	34.3	69.1	66.5	75.5	82.0	**60.3**	89.6	85.7
FocalFormer3D-F †	L+C	**72.9**	**75.0**	**88.8**	**63.5**	**39.0**	**73.7**	**66.9**	**79.2**	**81.0**	58.1	**91.1**	**87.1**

Table 1. **Performance comparison on the nuScenes 3D detection *test* set.** † represents using flipping test-time augmentation. ‡ means using both flipping and rotation test-time augmentation. C.V, Motor., Ped. and T.C. are short for construction vehicle, motorcycle, pedestrian, and traffic cones, respectively.

cluding 32-beams LiDAR with 20FPS and 6-view camera images. We mainly evaluate our method on both *LiDAR-only* and *LiDAR-Camera fusion* settings. The evaluation metrics follow nuScenes official metrics including mean average precision (mAP) and nuScenes detection score (NDS) defined by averaging the matching thresholds of center distance $\mathbb{D} = \{0.5, 1., 2., 4.\}$ (m). For evaluating the quality of object queries, we also introduce the Average Recall (AR) defined by center distance as well. The ablation studies in our research primarily utilize the nuScenes dataset, unless explicitly stated otherwise.

Waymo Open Dataset [61] has a wider detection range of $150m \times 150m$ compared to the nuScenes dataset. Waymo dataset comprises of 798 scenes for training and 202 scenes for validation. The official evaluation metrics used are mean Average Precision (mAP) and mean Average Precision with Heading (mAPH), where the mAP is weighted by the heading accuracy. The mAP and mAPH scores are computed with a 3D Intersection over Union (IoU) threshold of 0.7 for *Vehicle* and 0.5 for *Pedestrian* and *Cyclist*. The evaluation has two difficulty levels: Level 1, for boxes with more than five LiDAR points, and Level 2, for boxes with at least one LiDAR point. Of the two difficulty levels, Level 2 is prioritized as the primary evaluation metric for all experiments.

Implementation details. Our implementation is mainly based on the open-sourced codebase MMDetection3D [79]. For the LiDAR backbone, we use CenterPoint-Voxel as the point cloud feature extractor. For the multi-stage heatmap encoder, we apply 3 stages, generating a total of 600 queries by default. Data augmentation includes random double flipping along both X and Y axes, random global rotation between $[-\pi/4, \pi/4]$, the random scale of [0.9, 1.1], and random translation with a standard deviation of 0.5 in all axes. All models are trained with a batch size of 16 on eight V100 GPUs. More implementation details are referred to in supplementary files.

4.2. Main Results

nuScenes LiDAR-based 3D object detection. We evaluate the performance of FocalFormer3D on the nuScenes *test* set. As shown in Table 1, the results demonstrate its superiority over state-of-the-art methods on various evaluation metrics and settings. Our single-model FocalFormer3D achieved 68.7 mAP and 72.6 NDS, which surpasses the prior TransFusion-L method by +3.2 points on mAP and +2.4 points on NDS. Notably, even compared with the previous best method that was trained with segmentation-level labels, our method without extra supervision still outperformed LiDARMultiNet by +1.7 mAP and +1.0 NDS.

nuScenes multi-modal 3D object detection. We extend our approach to a simple multi-modal variant and demonstrate its generality. Following TransFusion [4], we use a pre-trained *ResNet-50* model on COCO [80] and nuImage [62] dataset as the image model and freeze its weights during training. To reduce computation costs, the input images are downscaled to 1/2 of their original size. Unlike heavy lift-splat-shot [32] camera encoders used in BEVFusion [5, 6], the multi-view camera images are projected onto a pre-defined voxel space and fused with LiDAR BEV feature. Additional details are available in the supplementary files. Without test-time augmentation, our simple multi-modal variant model outperforms all other state-of-the-art with less inference time (Table 2). With TTA, FocalFormer3D achieves 72.9 mAP and 75.0 NDS, ranking first among all single-model solutions on the nuScenes benchmark. Interestingly, our model achieves high results for some rare classes such as (*Trailer*, *Motorcycle*, *Bicycle*) compared to other methods.

nuScenes 3D object tracking. To further demonstrate the versatility, we also extend FocalFormer3D to 3D multi-object tracking (MOT) by using the tracking-by-detection algorithm SimpleTrack. Interested readers can refer to the original paper [81] for more comprehensive details. As depicted in Table 2, FocalFormer3D gets 2.9 points better than prior state-of-the-art TransFusion-L [4] in LiDAR settings and FocalFormer3D-F achieves 2.1 points over TransFusion in terms of AMOTA. Moreover, our single model FocalFormer3D-F with double-flip testing results performs even better than the BEVFusion [6] with model ensembling.

Waymo LiDAR 3D object detection. The results of our single-frame LiDAR 3D detection method on the Waymo dataset are presented in Table 3, alongside the comparison with other approaches. Employing with the same VoxelNet backbone as nuScenes, our method achieves competitive performance without any fine-tuning of the model hyperparameters specifically for the Waymo dataset. Particularly, when compared to TransFusion-L with the same backbone, our method exhibits a +1.1 mAPH improvement.

Methods	AMOTA	AMOTP	MOTA	IDS
LiDAR-based 3D Tracking				
AB3DMOT [82]	15.1	150.1	15.4	9027
CenterPoint [3]	63.8	55.5	53.7	760
CBMOT [83]	64.9	59.2	54.5	557
OGR3MOT [84]	65.6	62.0	55.4	**288**
SimpleTrack [81]	66.8	55.0	56.6	575
UVTR-L [7]	67.0	55.0	56.6	774
TransFusion-L [4]	68.6	52.9	57.1	893
FocalFormer3D	71.5	54.9	**60.1**	888
FocalFormer3D†	**72.1**	**47.0**	60.0	701
Multi-Modal 3D Tracking				
UVTR-MultiModal [7]	70.1	68.6	61.8	941
TransFusion [4]	71.8	55.1	**60.7**	944
BEVFusion-MIT [6]‡	74.1	40.3	60.3	506
FocalFormer3D-F	73.9	51.4	61.8	**824**
FocalFormer3D-F†	**74.6**	**47.3**	63.0	849

Table 2. **Performance comparison on nuScenes 3D tracking test set.** † is based on the double-flip testing results in Table 1. ‡ is based on model ensembling.

Methods	mAP	mAPH	Vel.	Ped.	Cyc.
LiDAR-based 3D Detection					
RSN* [85]	–	–	65.5	63.7	–
AFDetV2* [86]	71.0	68.8	69.2	67.0	70.1
SST* [55]	67.8	64.6	65.1	61.7	66.9
PV-RCNN* [24]	66.8	63.3	68.4	65.8	68.5
PV-RCNN++* [25]	71.7	69.5	70.2	**68.0**	70.2
PillarNet-34* [48]	71.0	68.8	**70.5**	66.2	68.7
FSD-spconv* [56]	**71.9**	**69.7**	68.5	**68.0**	72.5
CenterPoint [3]	69.8	67.6	73.4	65.8	68.5
TransFusion-L$^\wedge$ [4]	70.5	67.9	66.8	66.1	70.9
FocalFormer3D	71.5	69.0	67.6	66.8	**72.6**

Table 3. **Performance comparison on the Waymo *val* set.** All models inputs single-frame point clouds. The methods marked with * indicate the utilization of different point cloud backbones in VoxelNet. The method marked with $^\wedge$ indicates our reproduction. The evaluation metric used is the LEVEL 2 difficulty, and the results are reported on the full Waymo validation set.

4.3. Recall Analysis

To diagnose the performance improvements, we compare several recent methods in terms of AR for both stages – initial BEV heatmap predictions and final box predictions in Fig. 5. The metric of AR is computed based on center distance following the nuScenes metrics and different distance thresholds (*e.g.*, $0.5m$, $1.0m$, $2.0m$, $4.0m$), and the mean AR (mAR) are compared.

Recall comparison on initial object candidates. Figure 5 compares the recall of state-of-the-art methods that share the same SparseUNet backbone. With total 200 queries, FocalFormer3D-200P reaches 75.2 mAR, achieving considerable and consistent improvements by +4.5

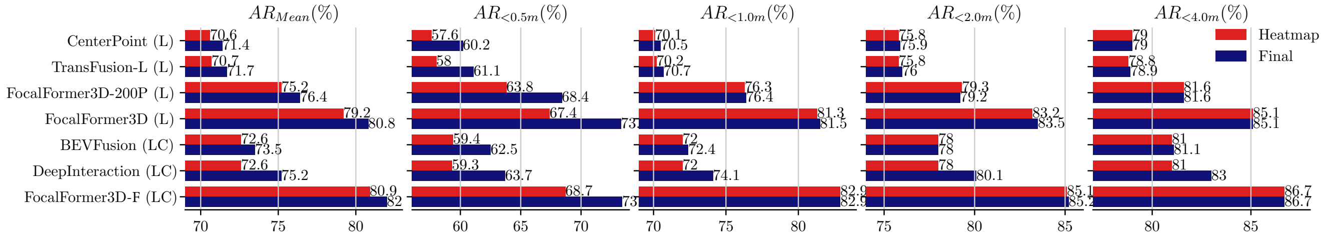

Figure 5. Average recall comparisons between initial object predictions and final object prediction centers on the nuScenes *val* set. The subfigures are shown over center distance thresholds (%) following nuScenes detection metrics.

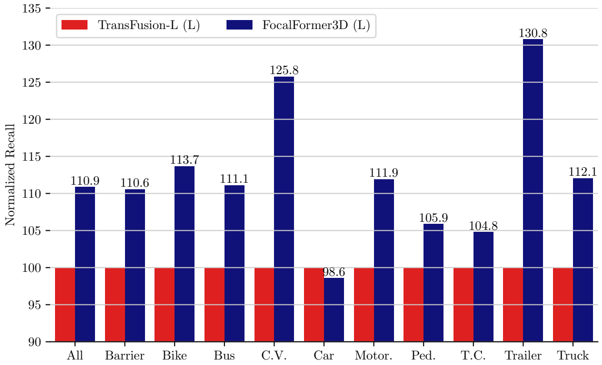

Figure 6. **Class-wise recall comparison on nuScenes val set** between TransFusion-L and FocalFormer3D in terms of recall values across nuScenes center distance (CD) threshes (0.25/0.5/1.0m) on the nuScenes *val* set. The red bars are normalized to 100%.

mAR compared with the prior state-of-the-art LiDAR approach TransFusion-L. Surprisingly, our LiDAR-based FocalFormer even achieves better results than the prior multi-modal approach DeepInteraction by 2.6 points in terms of mAR as well. As the query sizes get 600, FocalFormer3D achieves 79.2 mAR, surpassing the fusion approach DeepInteraction by 6.6 points. Further, by incorporating multi-view camera features, our multi-modal version FocalFormer-F gets improved to 80.9 mAR.

Recall comparison on final object prediction. Concerning the final predictions of 3D detectors, most LiDAR and fusion approaches obtain fewer performance improvements as the distance thresholds increase as shown in Fig. 5. This can be explained by higher distance thresholds indicating the performance for the extreme cases of missing detections. The introduction of camera features helps the model see the context in the perspective view, which leads to better performance such as DeepInteraction. However, their final prediction recall falls far behind FocalFormer-F with a large margin of 6.8 points.

Class-wise recall comparison. We compare the class-wise recall analysis for object candidates in Fig. 6 at the category level. The findings highlight the effectiveness of FocalFormer3D in improving the relative recall of initial BEV queries by a relative +10.9% improvement against TransFusion-L. Large objects such as *Construction Vehicles* and *Trailer* get the most improvements so that the predictions of their initial centers are challenging.

#	# Stages	#Total Queries	mAP	NDS
(a)	1	200	65.3	70.5
(b)	2	200	66.0	70.7
(c)	1	600	65.4	70.5
(d)	2	600	66.4	70.9
(e)	3	600	66.5	71.1

Table 4. **Effects of numbers of stages and total queries.** Here one stage stands for the baseline method without using hard instance probing.

Mask Type	mAP	NDS
None	65.3	70.4
Point-based	65.9	70.5
Box-based	66.1	70.9
Pooling-based	66.5	71.1

Table 5. **Effects of various positive mask types.** All models adopt the same network except for the masking way.

4.4. Ablation Study

HIP query sizes and generation stages. Table 4 ablates the impacts of the number of queries and stages in the multi-stage heatmap encoder. When using the same query size of rough 200, approaches (b), which uses additional one stage of HIP, demonstrates better performance than baseline (a) by a margin of +0.7 mAP. When provided with more queries (600), our approach (d) and (e) achieve over 1.1-point improvement in terms of mAP.

Positive mask type. Table 5 presents an ablation study on the effectiveness of Hard Instance Probing in terms of various mask types. Specifically, we compare the performance of our method with none masking, point-based masking,

#	M.S. Heat	Refinement Module		mAP	NDS
		BoxPool	C.A.		
(a)	✗	✗	✗	63.1	69.1
(b)	✓	✗	✗	63.3	69.3
(c)	✓	✓	✗	65.1	69.9
(d)	✓	✗	✓	65.9	70.9
(e)	✓	✓	✓	66.5	71.1
(f)	✓	Rescoring Only		66.1	68.8

Table 6. **Step-by-step improvements made by modules.** "M.S. Heat" represents the application of the multi-stage heatmap encoder for hard instance probing. "C.A." denotes using deformable cross attention for second-stage refinement. "BoxPool" represents the Box-pooling module. The term "Rescoring Only" refers to the model that directly generates box prediction from BEV feature and uses its decoder head to rescore the candidate predictions from heatmap without performing additional bounding box refinement.

and pooling-based masking. The results demonstrate that even with single-point masking, HIP improves the performance of the baseline by a gain of +0.6 points in terms of mAP. Furthermore, the pooling-based masking shows the best gain with +1.2 mAP and +0.7 NDS, outperforming the box-based masking. This can be attributed to two facts. Point or pooling-based masking can already effectively exclude positive objects as the center heatmap [3] only highlights a Gaussian peak. Second, the wrong false positive predictions or predicted boxes might lead to false masking of the ground-truth boxes, resulting in missed detection.

Step-by-step module refinement. We conduct ablation studies on the step-by-step improvements by each module, presented in Table 6, to illustrate the component effectiveness within hard instance probing (HIP) pipeline. Initially, without second-stage refinement, we used simple center-based predictions [3] (a), which estimate boxes directly from BEV feature by another convolutional layer.

Despite an improvement in the average recall by over 9 points in Fig. 5, we found little improvement of (b) over (a) in performance after using the multi-stage heatmap encoder to generate the object candidates. By applying simple object-level rescoring (c), with RoI-based refinement (using two hidden MLP layers), the performance is boosted to 65.1 mAP and 69.9 NDS. Remarkably, our complete box-level deformable decoder (e) further improves the performance by a margin of +1.4 mAP and +1.2 NDS.

To assess the effects of rescoring alone, we perform experiment (f), which excludes the effects of box regression by not using any box or position regression in the object-level refinement module. Despite this, experiment (f) still achieves high center accuracy (66.1 mAP) compared to (a). This finding highlights the limitations of the initial ranking of object candidates across stages based solely on BEV heatmap scores. Therefore, it validates the necessity for a second-stage object-level rescoring in the hard instance probing pipeline (Fig. 3).

Models/Components	Latency
TransFusion-L	93ms
FocalFormer3D	109ms
– VoxelNet backbone	78ms
– Multi-stage heatmap encoder	13ms
– Box-level deformable decoder	18ms

Table 7. **Latency analysis for model components.** Latency is measured on a V100 GPU for reference.

Latency analysis for model components. We conduct a latency analysis for FocalFormer3D on the nuScenes dataset. The runtimes are measured on the same V100 GPU machine for comparison. To ensure a fair speed comparison with CenterPoint [3], dynamic voxelization [87] is employed for speed testing of both TransFusion-L and FocalFormer3D. The computation time is mostly taken up by the sparse convolution-based backbone network (VoxelNet [1, 22]), which takes 78ms. Our multi-stage heatmap encoder takes 13ms to collect queries from the heatmaps across stages, while the box-level deformable decoder head takes 18ms. Note that, the generation of multi-stage heatmaps only takes 5ms, and additional operations such as Top-K selection takes 7ms, indicating potential optimization opportunities for future work.

5. Conclusion

In this work, we explicitly focus on the fatal problem in autonomous driving, *i.e.*, false negative detections. We present FocalFormer3D as solution. It progressively probes hard instances and improves prediction recall, via the hard instance probing (HIP). Nontrivial improvements are introduced with limited overhead upon transformer-based 3D detectors. The HIP algorithm enables FocalFormer3D to effectively reduce false negatives in 3D object detection.

Limitation. A key limitation is that FocalFormer3D's hard instance probing (HIP) relies on the assumption that object centers produce Gaussian-like peaks in the BEV heatmap, which may not hold for camera-based detectors where heatmaps tend to be fan-shaped. Additionally, few studies have explored hard instances in long-range detection, so more research is needed to evaluate HIP in this area. We leave more investigation of hard instance probing as future work.

References

[1] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, 2018. 1, 2, 9

[2] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. 2018. 1, 2, 6

[3] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *CVPR*, pages 11784–11793, 2021. 1, 2, 3, 4, 6, 7, 9

[4] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *CVPR*, pages 1080–1089, 2022. 1, 2, 4, 5, 6, 7

[5] Tingting Liang, Hongwei Xie, Kaicheng Yu, Zhongyu Xia, Zhiwei Lin, Yongtao Wang, Tao Tang, Bing Wang, and Zhi Tang. Bevfusion: A simple and robust lidar-camera fusion framework. In *Advances in Neural Information Processing Systems*. 1, 2, 6, 7

[6] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *arXiv preprint arXiv:2205.13542*, 2022. 1, 2, 6, 7

[7] Yanwei Li, Yilun Chen, Xiaojuan Qi, Zeming Li, Jian Sun, and Jiaya Jia. Unifying voxel-based representation with transformer for 3d object detection. 2022. 1, 2, 7

[8] Yanwei Li, Xiaojuan Qi, Yukang Chen, Liwei Wang, Zeming Li, Jian Sun, and Jiaya Jia. Voxel field fusion for 3d object detection. In *CVPR*, 2022. 1, 2

[9] Yang Jiao, Zequn Jie, Shaoxiang Chen, Jingjing Chen, Lin Ma, and Yu-Gang Jiang. Msmdfusion: A gated multi-scale lidar-camera fusion framework with multi-depth seeds for 3d object detection. *arXiv preprint arXiv:2209.03102*, 2022. 1

[10] Zeyu Yang, Jiaqi Chen, Zhenwei Miao, Wei Li, Xiatian Zhu, and Li Zhang. Deepinteraction: 3d object detection via modality interaction. *arXiv preprint arXiv:2208.11112*, 2022. 1, 6

[11] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, pages 770–779, 2019. 1, 2

[12] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast point r-cnn. In *ICCV*, pages 9775–9784, 2019. 1, 2

[13] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *AAAI*, volume 35, pages 1201–1209, 2021. 1, 2

[14] Zhichao Li, Feng Wang, and Naiyan Wang. Lidar r-cnn: An efficient and universal 3d object detector. *CVPR*, 2021. 1

[15] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. In *ICCV*, pages 2949–2958, 2021. 1

[16] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018. 2

[17] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 2, 5

[18] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *CVPR*, pages 13619–13627, 2022. 2, 5

[19] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019. 2, 4

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 2, 5

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 2

[22] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. 2018. 2, 9

[23] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *CVPR*, 2018. 2

[24] 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 *CVPR*, pages 10529–10538, 2020. 2, 7

[25] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. *IJCV*, pages 1–21, 2022. 2, 7

[26] Zixiang Zhou, Xiangchen Zhao, Yu Wang, Panqu Wang, and Hassan Foroosh. Centerformer: Center-based transformer for 3d object detection. In *ECCV*, 2022. 2

[27] Yukang Chen, Yanwei Li, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Focal sparse convolutional networks for 3d object detection. In *CVPR*, pages 5428–5437, 2022. 2, 6

[28] Yukang Chen, Jianhui Liu, Xiaojuan Qi, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Scaling up kernels in 3d cnns. *arXiv preprint arXiv:2206.10555*, 2022. 2, 6

[29] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[30] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. In *British Machine Vision Conference*, 2019. 2

[31] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Dsgn: Deep stereo geometry network for 3d object detection. In *CVPR*, pages 12536–12545, 2020. 2

[32] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, pages 194–210. Springer, 2020. 2, 7

[33] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022. 2

[34] Yunpeng Zhang, Zheng Zhu, Wenzhao Zheng, Junjie Huang, Guan Huang, Jie Zhou, and Jiwen Lu. Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving. *arXiv preprint arXiv:2205.09743*, 2022. 2

[35] Junjie Huang, Guan Huang, Zheng Zhu, Ye Yun, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 2

[36] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, pages 531–548. Springer, 2022. 2

[37] Yilun Chen, Shijia Huang, Shu Liu, Bei Yu, and Jiaya Jia. Dsgn++: Exploiting visual-spatial relation for stereo-based 3d detectors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2

[38] AJ Piergiovanni, Vincent Casser, Michael S Ryoo, and Anelia Angelova. 4d-net for learned multi-modal alignment. In *ICCV*, pages 15435–15445, 2021. 2

[39] Yingwei Li, Adams Wei Yu, Tianjian Meng, Ben Caine, Jiquan Ngiam, Daiyi Peng, Junyang Shen, Yifeng Lu, Denny Zhou, Quoc V Le, et al. Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In *CVPR*, pages 17182–17191, 2022. 2

[40] Zehui Chen, Zhenyu Li, Shiquan Zhang, Liangji Fang, Qinhong Jiang, and Feng Zhao. Autoalignv2: Deformable feature aggregation for dynamic multi-modal 3d object detection. *ECCV*, 2022. 2

[41] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Multimodal virtual point 3d detection. *NeurIPS*, 34:16494–16507, 2021. 2, 6

[42] Xuanyao Chen, Tianyuan Zhang, Yue Wang, Yilun Wang, and Hang Zhao. Futr3d: A unified sensor fusion framework for 3d detection. *arXiv preprint arXiv:2203.10642*, 2022. 2

[43] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *CVPR*, 2017. 2

[44] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *CVPR*, pages 7345–7353, 2019. 2

[45] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *ECCV*, 2018. 2

[46] Hongyang Li, Chonghao Sima, Jifeng Dai, Wenhai Wang, Lewei Lu, Huijie Wang, Enze Xie, Zhiqi Li, Hanming Deng, Hao Tian, et al. Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe. *arXiv preprint arXiv:2209.05324*, 2022. 2

[47] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. 2018. 2

[48] Guangsheng Shi, Ruifeng Li, and Chao Ma. Pillarnet: Real-time and high-performance pillar-based 3d object detection. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*, pages 35–52. Springer, 2022. 2, 6, 7

[49] Runzhou Ge, Zhuangzhuang Ding, Yihan Hu, Yu Wang, Sijia Chen, Li Huang, and Yuan Li. Afdet: Anchor free one stage 3d object detection. *arXiv preprint arXiv:2006.12671*, 2020. 2

[50] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector, 2020. 2

[51] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *ICCV*, 2019. 2

[52] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *ICCV*, pages 9277–9286, 2019. 2

[53] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. 2017. 2

[54] Charles R. Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 2

[55] Lue Fan, Ziqi Pang, Tianyuan Zhang, Yu-Xiong Wang, Hang Zhao, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Embracing Single Stride 3D Object Detector with Sparse Transformer. In *CVPR*, 2022. 2, 7

[56] Lue Fan, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Fully Sparse 3D Object Detection. *arXiv preprint arXiv:2207.10035*, 2022. 2, 7

[57] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *CVPR*, pages 11794–11803, 2021. 2, 6

[58] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. 3

[59] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020. 3, 5

[60] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 3

[61] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, pages 2446–2454, 2020. 3, 6

[62] 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 multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020. 3, 5, 7

[63] Kah-Kay Sung. Learning and example selection for object and pattern detection. 1996. 4

[64] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, 2016. 4

[65] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *ICCV*, 2017. 4

[66] Buyu Li, Yu Liu, and Xiaogang Wang. Gradient harmonized single-stage detector. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8577–8584, 2019. 4

[67] Yuhang Cao, Kai Chen, Chen Change Loy, and Dahua Lin. Prime sample attention in object detection. In *CVPR*, pages 11583–11591, 2020. 4

[68] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018. 4

[69] Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient detr: improving end-to-end object detector with dense prior. *arXiv preprint arXiv:2104.01318*, 2021. 5

[70] Duy-Kien Nguyen, Jihong Ju, Olaf Booij, Martin R Oswald, and Cees GM Snoek. Boxer: Box-attention for 2d and 3d transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4773–4782, 2022. 5

[71] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492*, 2019. 6

[72] Dongqiangzi Ye, Zixiang Zhou, Weijia Chen, Yufei Xie, Yu Wang, Panqu Wang, and Hassan Foroosh. Lidarmultinet: Towards a unified multi-task network for lidar perception. *arXiv preprint arXiv:2209.09385*, 2022. 6

[73] Junho Koh, Junhyung Lee, Youngwoo Lee, Jaekyum Kim, and Jun Won Choi. Mgtanet: Encoding sequential lidar points using long short-term motion-guided temporal attention for 3d object detection. *arXiv preprint arXiv:2212.00442*, 2022. 6

[74] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *CVPR*, pages 4604–4612, 2020. 6

[75] Jin Hyeok Yoo, Yecheol Kim, Jisong Kim, and Jun Won Choi. 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. In *European Conference on Computer Vision*, pages 720–736. Springer, 2020. 6

[76] Shaoqing Xu, Dingfu Zhou, Jin Fang, Junbo Yin, Zhou Bin, and Liangjun Zhang. Fusionpainting: Multimodal fusion with adaptive attention for 3d object detection. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3047–3054. IEEE, 2021. 6

[77] Junjie Yan, Yingfei Liu, Jianjian Sun, Fan Jia, Shuailin Li, Tiancai Wang, and Xiangyu Zhang. Cross modal transformer via coordinates encoding for 3d object dectection. *arXiv preprint arXiv:2301.01283*, 2023. 5

[78] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Harry Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *ICLR*, 2022. 5

[79] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d, 2020. 6

[80] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 7

[81] Ziqi Pang, Zhichao Li, and Naiyan Wang. Simpletrack: Understanding and rethinking 3d multi-object tracking. *arXiv preprint arXiv:2111.09621*, 2021. 7

[82] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3d multi-object tracking: A baseline and new evaluation metrics. In *IROS*, pages 10359–10366, 2020. 7

[83] Nuri Benbarka, Jona Schröder, and Andreas Zell. Score refinement for confidence-based 3d multi-object tracking. In *IROS*, pages 8083–8090, 2021. 7

[84] Jan-Nico Zaech, Dengxin Dai, Alexander Liniger, Martin Danelljan, and Luc Van Gool. Learnable online graph representations for 3d multi-object tracking. *CoRR*, abs/2104.11747, 2021. 7

[85] Pei Sun, Weiyue Wang, Yuning Chai, Gamaleldin Elsayed, Alex Bewley, Xiao Zhang, Cristian Sminchisescu, and Dragomir Anguelov. Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In *CVPR*, pages 5725–5734, 2021. 7

[86] Yihan Hu, Zhuangzhuang Ding, Runzhou Ge, Wenxin Shao, Li Huang, Kun Li, and Qiang Liu. Afdetv2: Rethinking the necessity of the second stage for object detection from point clouds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 969–979, 2022. 7

[87] Yin Zhou, Pei Sun, Yu Zhang, Dragomir Anguelov, Jiyang Gao, Tom Ouyang, James Guo, Jiquan Ngiam, and Vijay Vasudevan. End-to-end multi-view fusion for 3d object detection in lidar point clouds. In *Conference on Robot Learning*, pages 923–932. PMLR, 2020. 9