

ConQueR: Query Contrast Voxel-DETR for 3D Object Detection

Benjin Zhu¹ Zhe Wang¹ Shaoshuai Shi² Hang Xu³ Lanqing Hong³ Hongsheng Li^{1,4}

¹Multimedia Laboratory, The Chinese University of Hong Kong

²Max Planck Institute for Informatics

³Huawei Noah's Ark Lab

⁴Centre for Perceptual and Interactive Intelligence

{benjinzhu@link, hsl@ee}.cuhk.edu.hk

Abstract

Although DETR-based 3D detectors simplify the detection pipeline and achieve **direct** sparse predictions, their performance still lags behind dense detectors with post-processing for 3D object detection from point clouds. DETRs usually adopt a larger number of queries than GTs (e.g., 300 queries v.s. ~ 40 objects in Waymo) in a scene, which inevitably incur many false positives during inference. In this paper, we propose a simple yet effective sparse 3D detector, named **Query Contrast Voxel-DETR (ConQueR)**, to eliminate the challenging false positives, and achieve more accurate and sparser predictions. We observe that most false positives are highly overlapping in local regions, caused by the lack of explicit supervision to discriminate locally similar queries. We thus propose a Query Contrast mechanism to explicitly enhance queries towards their best-matched GTs over all unmatched query predictions. This is achieved by the construction of positive and negative GT-query pairs for each GT, and a contrastive loss to enhance positive GT-query pairs against negative ones based on feature similarities. ConQueR closes the gap of sparse and dense 3D detectors, and reduces $\sim 60\%$ false positives. Our single-frame ConQueR achieves 71.6 mAPH/L2 on the challenging Waymo Open Dataset validation set, outperforming previous sota methods by over **2.0** mAPH/L2. **Code**

1. Introduction

3D object detection from point clouds has received much attention in recent years [7, 32, 34, 47, 52] as its wide applications in autonomous driving, robots navigation, etc. State-of-the-art 3D detectors [7, 31, 33, 53] still adopt dense predictions with post-processing (e.g., NMS [2]) to obtain final sparse detections. This *indirect* pipeline usually involves many hand-crafted components (e.g., anchors, center masks) based on human experience, which involves much effort for tuning, and prevents dense detectors from being

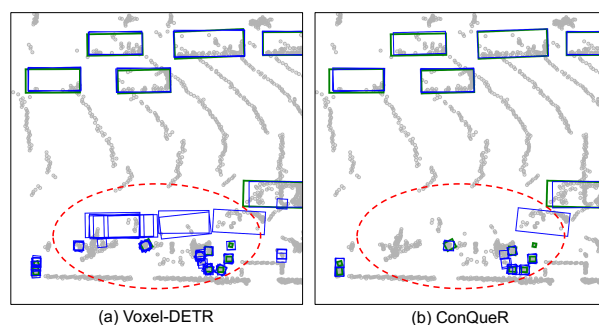


Figure 1. Comparison of our baseline Voxel-DETR and ConQueR. GTs (green) and predictions (blue) of an example scene in the WOD is visualized. Sparse predictions of Voxel-DETR still contain many highly overlapped false positives (in the red dashed circle), while ConQueR can generate much sparser predictions.

optimized end-to-end to achieve optimal performance. Recently, DETR-based 2D detectors [3, 39, 49, 57] show that transformers with *direct* sparse predictions can greatly simplify the detection pipeline, and lead to better performance. However, although many efforts [1, 26, 27] have been made towards *direct* sparse predictions for 3D object detection, because of the different characteristics of images and point clouds (i.e., dense and ordered images v.s. sparse and irregular points clouds), performance of sparse 3D object detectors still largely lags behind state-of-the-art dense detectors.

To achieve direct sparse predictions, DETRs usually adopt a set of object queries [1, 3, 27, 39, 49, 57], and resort to the one-to-one Hungarian Matching [17] to assign ground-truths (GTs) to object queries. However, to guarantee a high recall rate, those detectors need to impose much more queries than the actual number of objects in a scene. For example, recent works [1, 27] select top-300 scored query predictions to cover only ~ 40 objects in each scene of Waymo Open Dataset (WOD) [36], while 2D DETR detectors [3, 39, 49, 57] use $10\times$ more predictions than the average GT number of MS COCO [22]. As shown in Fig. 1(a), we visualize an example scene by a baseline DETR-based

3D detector, named Voxel-DETR, which shows its top-300 scored predictions. Objects are generally small and densely populated in autonomous driving scenes, while 3D DETRs adopt the same fixed top- N scored predictions as 2D DETRs, and lack a mechanism to handle such small and dense objects. Consequently, they tend to generate densely overlapped false positives (in the red-dashed circle), harming both the accuracy and *sparsity* [29, 39] of final predictions.

We argue the key reason is that the Hungarian Matching in existing 3D DETRs only assigns each GT to its best matched query, while all other unmatched queries near this GT are not effectively suppressed. For each GT, the one-to-one matching loss solely forces all unmatched queries to predict the same “no-object” label, and the best matched query are supervised *without* considering its relative ranking to its surrounding unmatched queries. This design causes the detectors to be insufficiently supervised in discriminating similar query predictions for each GT, leading to duplicated false positives for scenes with densely populated objects.

To overcome the limitations of current supervision, we introduce a simple yet novel Query Contrast strategy to explicitly suppress predictions of all unmatched queries for each GT, and simultaneously enhance the best matched query to generate more accurate predictions in a contrastive manner. The Query Contrast strategy is integrated into our baseline Voxel-DETR, which consists of a sparse 3D convolution backbone to extract features from voxel grids, and a transformer encoder-decoder architecture with a bipartite matching loss to directly generate sparse predictions. Our Query Contrast mechanism involves the construction of positive and negative GT-query pairs, and the contrastive learning on all GT-query pairs to supervise both matched and unmatched queries with knowledge of the states of their surrounding queries. Such GT-query pairs are directly created by reusing the Hungarian Matching results: each GT and its best matched query form the positive pair, and all other unmatched queries of the same GT then form negative pairs. To quantitatively measure the similarities of the GT-query pairs, we formulate the object queries to be the same as GT boxes (*i.e.*, using only box categories, locations, sizes and orientations), such that GTs and object queries can be processed by the same transformer decoder, and embedded into a unified feature space to properly calculate their similarities. Given the GT-query similarities, we adopt the contrastive learning loss [5, 12, 54] to effectively enhance the positive (matched) query’s prediction for each GT, and suppress those of all its negative queries at the same time. Moreover, to further improve the contrastive supervision, we construct multiple positive GT-query pairs for each GT by adding small random noises to the original GTs, which greatly boost the training efficiency and effectiveness. The resulting sparse 3D detector **Query Contrast**

Voxel-DETR (**ConQueR**) significantly improves the detection performance and sparsity of predictions, as shown in Fig. 1(b). Besides, ConQueR abandons the fixed top- N prediction scheme and achieves dynamic prediction numbers across scenes. ConQueR reduces $\sim 60\%$ false positives and sets new records on the challenging Waymo Open Dataset (WOD) [36]. Contributions are summarized as below:

1. We introduce a novel Query Contrast strategy into DETR-based 3D detectors to effectively eliminate densely overlapped false positives and achieve more accurate predictions.
2. We propose to construct multi-positive contrastive training, which greatly improve the effectiveness and efficiency of our Query Contrast mechanism.
3. Our proposed sparse 3D detector ConQueR closes the gap between sparse and dense 3D detectors, and sets new records on the challenging WOD benchmark.

2. Related Works

End-to-End 2D Object Detection. End-to-end object detection aims to generate final sparse predictions without non-differentiable components like NMS. RelationNet [14] proposes an object relation module and DETR [3] greatly simplifies the detection pipeline by removing many hand-crafted components like anchors, NMS, etc. DETR introduce a set of object queries and resorts to the Hungarian Matching to associate each GT with the query predictions of minimal matching cost, and selects top- N scored predictions for inference. [39, 42] also reveal that one-to-one matching is the key to achieve sparse predictions. Following works [16, 19, 19, 25, 43, 57] improves DETR in many aspects including query design, convergence speed, and performance, surpassing CNN-based dense detectors [8, 51, 56] by a large margin. However, they still need to select a fixed number of predictions as final results, no matter how many objects are there in an image. Recently, DINO-DETR [49] introduces a “contrastive” denoising training strategy. It creates positive and negative GTs conceptually, and supervise these GTs with different targets separately, which has no relation with contrastive learning.

3D Object Detection from Point Clouds. State-of-the-art 3D detectors usually adopts voxel-based [31–33, 47], range-view [38, 40] or point-based [7, 45] paradigms to convert raw point clouds into dense feature representations, followed by detection heads to generate dense predictions and resort to NMS to filter out low-quality predictions. Many attempts have also been made to incorporate transformer architectures [24, 30, 37, 53] into 3D object detection, but they still rely on post-processing. Others [1, 27] make a step

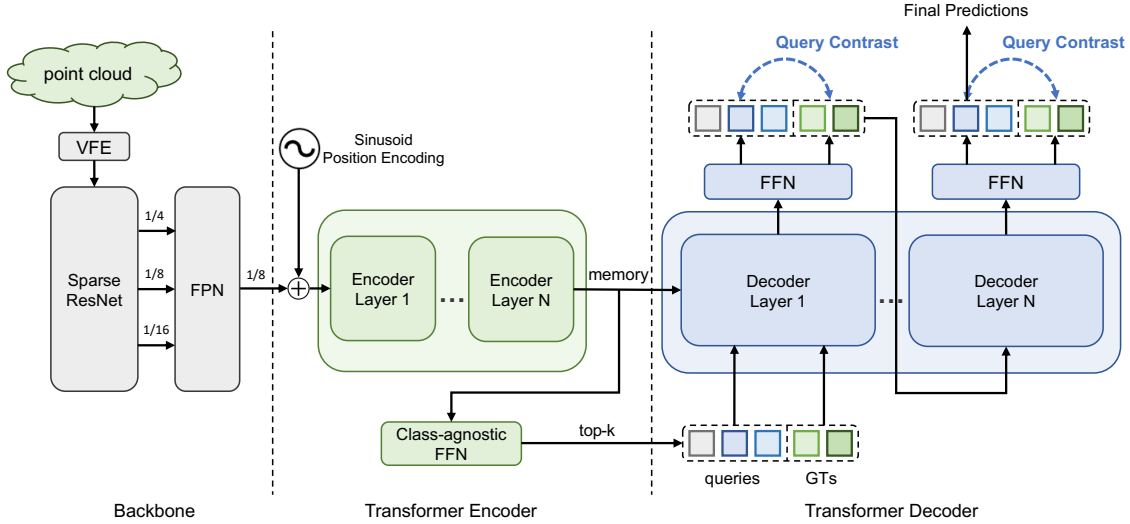


Figure 2. Overall pipeline of the proposed ConQueR. It consists of a 3D Sparse ResNet-FPN backbone to extract dense BEV features, and a transformer encoder-decoder architecture with one-to-one matching. Top- k scored object proposals from a class-agnostic FFN form the object queries to input to the transformer decoder. During training, GTs (noised) are concatenated with object queries to input to the transformer decoder to obtain unified embeddings, which are then used for Query Contrast at each decoder layer. During inference, Top-scored predictions from the last decoder layer are kept as final sparse predictions. “VFE” denotes the voxel feature extractor in [44,47,55].

further to use the one-to-one matching loss to achieve direct sparse 3D predictions. [27] proposes Box-Attention, a variant of deformable attention to better capture local informations and applies it to 3D object detection. [1] introduce image features into a decoder-only architecture to enhance query features. However, their performance still largely lags behind state-of-the-art dense 3D detectors.

Contrastive Learning for Object Detection. Contrastive learning aims to learn an embedding space such that similar data pairs stay close while dissimilar ones are far apart. [10] proposes to learn representations by contrasting positive pairs against negative ones. The popular InfoNCE loss [28] uses categorical cross-entropy loss to learn such an embedding space. Following works [4, 5, 12] demonstrate the superiority of contrastive learning on providing pre-trained weights for downstream tasks (*e.g.*, 2D detection). Few works explore the use of contrastive loss in object detection. [18] introduces semantically structured embeddings from knowledge graphs to alleviate misclassifications. [46] conducts contrastive distillation between different feature regions to better capture teacher’s information. As far as we know, we are the first to introduce the contrastive learning process into DETR-based detectors.

3. Query Contrast Voxel-DETR (ConQueR)

State-of-the-art 3D detectors usually generate dense object predictions, which require many hand-designed components (*e.g.*, anchors, box masks) based on prior knowledge, and resort to post-processing to filter out low-quality

and duplicated boxes. This *indirect* pipeline hinders the detectors from being optimized end-to-end and achieving optimal performance. 3D DETRs aim at streamlining these hand-crafted modules, and *directly* generating sparse predictions via the transformer architecture and one-to-one matching loss, but they still cannot compete with state-of-the-art dense 3D detectors and face the problem of highly overlapped false positives, as shown in Fig. 1(a). To solve these challenges, we first introduce our competitive DETR-based 3D framework, named Voxel-DETR in Sec. 3.1, and present the Query Contrast strategy to tackle with the duplicated false positives and further improve the detection performance in Sec. 3.2.

3.1. Voxel-DETR

As illustrated in Fig. 2, Voxel-DETR consists of a 3D backbone, an encoder-decoder transformer architecture, and a set-matching loss to achieve direct sparse predictions. **Backbone.** Point cloud is rasterized into sparse voxel grids and fed into a 3D Sparse ResNet [13] backbone network to extract sparse 3D features. These features are transformed into dense Bird Eye View (BEV) feature maps, followed by an FPN [20] to extract multi-scale features.

Transformer. The encoder-decoder transformer is similar to the two-stage Deformable-DETR [57]. The $8 \times$ down-scaled BEV features from the FPN are input to the transformer encoder, which consists of 3 encoder layers. Considering the characteristics of 3D detection from point clouds (*i.e.*, all objects are relatively small and densely distributed), we adopt BoxAttention [27], which applies spatial in-box constraints to Deformable Attention [57], to perform lo-

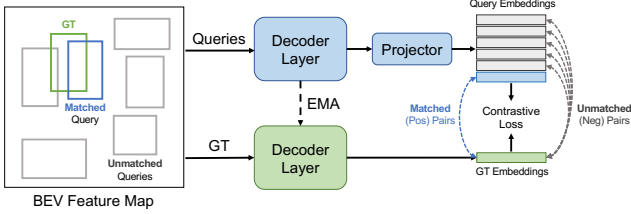


Figure 3. Illustration of Query Contrast. Given the GT (green), Hungarian Matching gives its best matched (blue) and all other unmatched (gray) object queries. Query embeddings are projected by an extra MLP to align with GT embeddings. The contrastive loss is applied to all positive and negative GT-query pairs based on their feature similarities.

cal self-attention. A class-agnostic feed-forward network (FFN) head is used to generate initial object proposals from encoder features. Top- k scored box proposals are selected as object queries to input to the 3-layer transformer decoder. Decoder layers conduct inter-query self-attention and cross-attention between query and encoder features, followed by prediction heads to perform iterative box refinement [57]. Predicted query boxes from the previous decoder layer’s FFN head are transformed by a 3-layer MLP and added with the updated query features (initialized as zero) from the previous decoder layer.

Losses. During training, all FFN prediction heads use the Hungarian Matching to assign GTs to object queries. The detection loss \mathcal{L}_{det} consists of a focal loss [21] for classification, a smooth L1 loss and a 3D GIoU loss for box regression:

$$\mathcal{L}_{det} = \alpha \mathcal{L}_{focal} + \beta \mathcal{L}_{l1} + \gamma \mathcal{L}_{GIoU}, \quad (1)$$

where α, β, γ are hyper-parameters to balance the loss terms. During inference, top- N scored predictions from the last decoder layer are kept as the final sparse detections.

3.2. Query Contrast

Although Voxel-DETR already achieves satisfactory performance, its top- N scored predictions still suffer from densely overlapped false positives (as shown in Fig. 1(a)). To tackle this problem, we present a novel Query Contrast mechanism (depicted in Fig. 3) to explicitly enhance each GT’s best matched query over unmatched ones. We first construct positive and negative GT-query pairs for each GT, which are then processed by each decoder layer to generate aligned GT and query embeddings. To promote the positive queries’ similarity towards a GT against negative ones, the contrastive loss is applied at each decoder layer.

Construction of positive/negative GT-query pairs. To determine queries to be enhanced or suppressed for each GT, we first construct positive and negative GT-query pairs by reusing the Hungarian Matching results (used for

Eq.(1)), which is naturally compatible with our Voxel-DETR framework. Given a GT, the query with the minimal matching cost forms a positive pair with the GT, all other queries and this GT then form negative GT-query pairs. These GT-query pairs help to identify the object queries that need to be further enhanced or suppressed in our Voxel-DETR. Motivated by the SwAV [4] that incorporates multiple image crops to form multiple positive pairs to boost the training process, we further add small noises of different magnitudes on each GT to generate multiple noised GT copies. The multiple noised GT copies then form additional GT-query pairs with the same positive/negative query partitions as original GTs.

In practice, if a noised copy deviates too much from its original GT, the noised GT-query pairs would harm the contrastive training process. However, finding proper noise magnitudes is rather laboursome and cannot generalise well across scenarios. We thus add an auxiliary GT de-noising loss similar to that in DN-DETR [19] to obligate the detector to recover the original GT from its noised versions, which ensures that the noised GT copies would not diverge. Note that the “noising-denoising” step alone only has marginal effects to detection performance, while our multipositive Query Contrast based on the noised GT copies leads to superior detection performance, as shown in our ablation studies.

Contrast positive pairs against negative pairs. Before applying supervisions to the positive and negative GT-query pairs, we need to quantitatively measure the similarities of these pairs. However, simple geometric metrics (*e.g.*, IoU) cannot sufficiently model the similarities between GTs and queries (*i.e.*, category, appearance, location, size, etc.). We thus propose to embed GTs and queries into a latent space for comprehensive similarity measurement. In our Voxel-DETR, object queries are formulated as proposal boxes (*i.e.* object category, box location, size, and orientation). Therefore, the transformer decoder can naturally be used to encode both GTs and queries into feature embeddings at a chosen layer. We simply select the output layer of the FFN prediction head after each decoder layer (as shown in Fig. 2), followed by a shared MLP for similarity estimation.

However, we observe that the distributions of GT objects and query boxes can be quite different: GTs have no overlap with each other and generally distribute following the roadmap layouts, while queries might correspond to densely overlapped boxes and show up at random locations. As the transformer decoder utilizes self-attention to capture inter-box relations, the different distributions of GTs and query boxes would greatly affect estimation of their similarities. To mitigate the distribution gap, we adopt an extra MLP to project query features to align with GTs’ latent space (the “Projector” in Fig. 3). With the aligned GT and query embeddings, we estimate all positive and negative GT-query

pairs’ similarities with cosine similarity metric, and adopt the InfoNCE loss [28] to encourage the best matched query to generate more accurate predictions towards its assigned GT, and force all other unmatched queries to deviate away. Moreover, to obtain more stable GT representations for supervising queries, we adopt an exponential moving average (EMA) copy for each decoder layer to embed GTs, which is shown to be effective in our ablations.

Assume that for the i -th GT in a point cloud scene, we add T different noises and denote the noised GT embeddings as $\{b_i^1, b_i^2, \dots, b_i^T\}$, and denote K query embeddings as $\{q_1, q_2, \dots, q_K\}$. Suppose that the Hungarian Matching assigns the i -th GT to the j -th query, then our Query Contrast loss for the i -th GT $\mathcal{L}_i^{\text{QC}}$ can be formulated as

$$\mathcal{L}_i^{\text{QC}} = - \sum_{t=1}^T \log \left(\frac{\exp(\cos(b_i^t, g(q_j))/\tau)}{\sum_{k=1}^K \exp(\cos(b_i^t, g(q_k))/\tau)} \right), \quad (2)$$

where τ is the temperature coefficient, and $g(\cdot)$ denotes the extra MLP projector to align query features to GTs’. As shown in Fig. 2, the Query Contrast loss is adopted at every decoder layer.

During inference, we abandon the widely adopted top- N scored prediction strategy and use a score threshold (*e.g.*, 0.1) to filter out low-quality query predictions. Query Contrast works quite well on suppressing similar query predictions in local neighborhoods, as shown in Fig. 1(b). ConQueR greatly boosts the detection accuracy, and reduces up to $\sim 60\%$ false positives.

Discussion: Why does Query Contrast improve DETR-based 3D detectors? As discussed in Sec. 1, current detection losses (*i.e.*, focal loss for classification, smooth L1 and GIoU loss for regression) supervise each query without considering its surrounding queries, which lack supervision to train detectors to discriminate similar object queries especially in local regions. The proposed Query Contrast strategy tackles this issue by constructing a contrastive objective to supervise *all* queries simultaneously. As suggested in Eq.(2), for each GT object, the detector is instructed to identify the best matched query, and is forced to learn to differentiate it from all other unmatched counterparts, even if some of them are highly overlapping with the best matched query. As a result, all unmatched queries are trained to deviate from the GT, thus the duplicated false positives in our baseline Voxel-DETR can be effectively suppressed.

Another core design of our Query Contrast is to encode the GTs and queries into a unified learnable latent space. GT objects are encoded to provide better forms of supervision for both matched and unmatched queries. Previous works [11, 50] in 2D object detection also show that encoding labels into feature embeddings to serve as extra supervision can perform better than the common hand-designed

learning targets (*i.e.*, classification logits and regression offsets), but they generally work in a knowledge distillation (KD) manner, which cannot be utilized to supervise negative queries. In contrast, our contrastive loss does not force matched queries to approach GTs directly, but encourages them to be “closer” to their corresponding GT embeddings than other close-by duplicated queries. Note that in our Query Contrast mechanism, GT embeddings are processed in an off-line manner and encoded into a unified space as queries’, which serve as a type of supervision and force the detector to generate more similar query features as GTs’.

According to our experiments, the proposed Query Contrast strategy can not only suppress those duplicated false positives, but also contribute to better detection performance, which are consistent with the above discussions.

4. Experiments

ConQueR is mainly evaluated on the Waymo Open Dataset [36] (WOD) benchmark using the official detection metrics: mAP and mAPH (mAP weighted by heading) for Vehicle (Veh.), Pedestrian (Ped.), and Cyclist (Cyc.). The metrics are further splitted into two difficulty levels according to the point numbers in GT boxes: LEVEL_1 (>5) and LEVEL_2 (≥ 1). We conduct ablation studies on the *validation* set, and compare with state-of-the-art detectors on both *validation* and *test* set.

4.1. Implementation Details

Training. We follow common practice as previous voxel-based methods [31–33, 47] to use point cloud range of $[-75.2m, 75.2m] \times [-75.2m, 75.2m] \times [-2.0m, 4.0m]$ with voxel size $[0.1m, 0.1m, 0.15m]$ in x , y , and z -axes respectively. The same set of augmentations (*i.e.*, GT-Aug, flip, rotation, scaling) are adopted following the previous works [47]. We follow [1, 41] to use the “fade-strategy” to drop GT-Aug at the last epoch to avoid overfitting. Both our baseline Voxel-DETR and ConQueR are trained for 6 epochs unless otherwise specified. We use the OneCycle [35] learning rate scheduler and AdamW [23] optimizer with maximal learning rate 0.001.

Network. For the 3D backbone in Fig. 2, we use the same architecture as ResNet-18 [13] but use sparse 3D convolutions [9] to replace the 2D ones. No pre-trained weights are used. The same FPN structure as RetinaNet [21] is used to obtain multi-scale BEV features. For simplicity, we only use the $8 \times$ downscaled features as input to the transformer, which adopts 3 encoder layers and 3 decoder layers for computation efficiency. We select top-1000 scored query predictions from the encoder’s class-agnostic prediction head as object queries. We adopt top- N (*e.g.*, 300) scored predictions, or score threshold (*e.g.*, ≥ 0.1) during inference. We set $\alpha = 1$, $\beta = 4$, $\gamma = 2$ in Eq. (1). For the proposed Query Contrast, we use $\tau = 0.7$ in Eq. (2), and

| Methods | mAP/mAPH | Vehicle 3D AP/APH | | Pedestrian 3D AP/APH | | Cyclist 3D AP/APH | |
|--------------------------------|------------------|-------------------|------------------|----------------------|------------------|-------------------|------------------|
| | L2 | L2 | L1 | L2 | L1 | L2 | L1 |
| Dense Detectors | | | | | | | |
| CenterPoint _{ts} [47] | -/67.4 | -/67.9 | -/- | -/65.6 | -/- | -/68.6-/- | -/- |
| PV-RCNN [32] | 66.8/63.3 | 69.0/68.4 | 77.5/76.9 | 66.0/57.6 | 75.0/65.6 | 65.4/64.0 | 67.8/66.4 |
| AFDetV2 [15] | 71.0/68.8 | 69.7/69.2 | 77.6/77.1 | 72.2/67.0 | 80.2/74.6 | 71.0/70.1 | 73.7/72.7 |
| SST_TS [6] | -/- | 68.0/67.6 | 76.2/75.8 | 72.8/65.9 | 81.4/74.1 | -/- | -/- |
| SWFormer [37] | -/- | 69.2/68.8 | 77.8/77.3 | 72.5/64.9 | 80.9/72.7 | -/- | -/- |
| PillarNet-34 [31] | 71.0/68.5 | <u>70.9/70.5</u> | <u>79.1/78.6</u> | 72.3/66.2 | 80.6/74.0 | 69.7/68.7 | 72.3/71.2 |
| CenterFormer [53] | 71.2/69.0 | 70.2/69.7 | 75.2/74.7 | 73.6/68.3 | 78.6/73.0 | 69.8/68.8 | 72.3/71.3 |
| PV-RCNN++ [33] | 71.7/69.5 | 70.6/ <u>70.2</u> | 79.3/78.8 | 73.2/68.0 | <u>81.3/76.3</u> | 71.2/70.2 | 73.7/72.7 |
| Sparse Detectors | | | | | | | |
| BoxeR-3D [27] | -/- | 63.9/63.7 | 70.4/70.0 | 61.5/53.7 | 64.7/53.5 | -/- | 50.2/48.9 |
| TransFusion-L [1] | -/64.9 | -/65.1 | -/- | -/63.7 | -/- | -/65.9 | -/- |
| Voxel-DETR (ours) | 68.8/66.1 | 67.8/67.2 | 75.4/74.9 | 69.7/63.1 | 77.6/70.5 | 69.0/67.9 | 71.7/70.5 |
| ConQueR (ours) | 70.3/67.7 | 68.7/68.2 | 76.1/75.6 | 70.9/64.7 | 79.0/72.3 | 71.4/70.1 | 73.9/72.5 |
| ConQueR †(ours) | <u>73.1/70.6</u> | 71.0/70.5 | 78.4/77.9 | <u>73.7/68.1</u> | 80.9/75.2 | <u>74.5/73.3</u> | <u>77.3/76.1</u> |
| ConQueR ‡(ours) | 74.0/71.6 | 71.0/70.5 | 78.4/77.9 | 75.8/70.1 | 82.4/76.6 | 75.2/74.1 | 77.5/76.4 |

Table 1. Performances on the WOD *validation* split. All models take single-frame input with the same range, no pre-training or ensembling is required. † denotes using the 2× wider ResNet [48] with 1/4 downscaled BEV feature map in our backbone. ‡ denotes conducting NMS on pedestrians and cyclists. **Bold** denotes the best entries, and underline denotes the second-best entries. _{ts} denotes the two-stage model.

adopt $T = 3$ noising groups with a maximal box noise ratio of 0.4 [19], and label noise ratio of 0.5 [19]. Category labels are simply encoded as one-hot embeddings rather than the learnable embeddings in DN-DETR [19].

4.2. Main Results

For fair comparison, all methods included use the same point cloud input range, do not use any pre-trained weights, test-time augmentation or model ensembling.

Performance. As shown in Table 1, state-of-the-art 3D detectors are divided into dense and sparse categories according to whether they can *directly* generate sparse detections. Our sparse detector ConQueR sets new records on *all categories* of the WOD *validation* set. ConQueR with direct sparse predictions (the second-last entry) achieves ~ 1.0 mAPH/L2 higher than the previous best single-frame model PV-RCNN++ [33], and is over 3.0 mAPH/L2 higher than the popular anchor-free CenterPoint [47]. Notably, ConQueR demonstrates overwhelming performance on pedestrians and cyclists, outperforming previous best methods by ~ 2.0 APH/L2, which shows the effectiveness of our Query Contrast strategy especially for densely populated categories. The significant performance improvements can also be validated on the WOD *test* set in Table 2. Moreover, ConQueR surpasses previous best sparse detectors TransFusion-L by ~ 6.0 mAPH/L2, closing the performance gap between sparse and dense 3D detectors. When compared with our baseline Voxel-DETR, the proposed Query Contrast mechanism brings over 1.6 mAPH/L2 without any extra inference cost. Besides, our baseline Voxel-DETR with only 6 epochs of training outperforms previous sparse 3D detectors, and achieves comparable performance with

| Methods | All | Veh. | Ped. | Cyc. |
|-------------------|-------------|-------------|-------------|-------------|
| CenterPoint [47] | 69.0 | 71.9 | 67.0 | 68.2 |
| PV-RCNN++ [33] | 70.2 | 73.5 | 69.0 | 68.2 |
| AFDetv2 [15] | 70.0 | 72.6 | 68.6 | 68.7 |
| PillarNet-34 [31] | 69.6 | 74.7 | 68.5 | 65.5 |
| ConQueR (Ours) | 72.0 | 73.3 | 70.9 | 71.9 |

Table 2. Single-frame performance comparisons on the WOD *test* set. APH/L2 results are reported.

CenterPoint (36-epoch training) with only 1/6 GPU hours. In addition, ConQueR has an inference latency of 70ms (46ms for CenterPoint)¹.

Although ConQueR with *direct* sparse predictions already achieves state-of-the-art performance, we find that applying NMS onto ConQueR’s sparse predictions can further improve small and densely populated categories such as pedestrians, while NMS causes ~ 1.2 APH/L2 performance drop on the well-trained vehicles (as shown in Appendix. A). This is also the case with our baseline Voxel-DETR. We speculate this is caused by the learning difficulties inherent in the data for extremely similar queries (as shown in Fig. 1(b)). We thus report ConQueR’s performance after conducting NMS on pedestrians and cyclists (the last entry of Table 1).

Sparsity. Apart from the performance improvements on the WOD official metrics, ConQueR shows great potential in reducing false positives and improving the sparsity of final predictions. We list the average number of predictions per scene for different 3D detectors in Table 3. For the baseline Voxel-DETR, thresholding according to scores helps to reduce $\sim 25\%$ predictions per sample with slightly

¹Latency is measured with batch size 1 on NVIDIA A100 GPU.

| Methods | Preds/Scene | Veh. | Ped. | Cyc. |
|---------------------------------------|-------------|-------------|-------------|-------------|
| CenterPoint _{nms} | 192 | 66.4 | 62.9 | 67.9 |
| Transfusion _{topN} | 300 | 65.1 | 63.7 | 65.9 |
| Voxel-DETR _{topN} | 300 | 67.1 | 63.0 | 67.8 |
| Voxel-DETR _{score} | 222 | 67.2 | 63.1 | 67.9 |
| ConQueR _{topN} | 300 | 68.0 | 64.6 | 70.0 |
| ConQueR _{score} | 131 | 68.2 | 64.7 | 70.1 |
| ConQueR _{score} [†] | 122 | 70.5 | 68.1 | 73.3 |

Table 3. Sparsity of final predictions. APH/L2 results are reported on the WOD *validation* set. The subscripts of each entry denotes the way they obtain final predictions. For example, CenterPoint_{nms} uses NMS to filter out duplicated boxes, and Voxel-DETR_{topN} denotes it uses top- N scored proposals as final predictions, while ConQueR_{score} denotes that using score thresholding to generate final sparse predictions. † denotes our best model in Table 1.

better performance. With the help of Query Contrast, ConQueR further reduces the number of predictions substantially by $\sim 60\%$. Besides, as the performance of ConQueR continually improves (the last two lines), the sparsity of final predictions steadily improve as well. When we adopt the same top-300 predictions as baseline Voxel-DETR_{topN} for evaluation, ConQueR_{topN} still improves the detection performance significantly. This indicates the Query Contrast mechanism contributes to generating more accurate predictions from best matched queries. Furthermore, our ConQueR can achieve much sparser predictions even compared with NMS-based dense detectors such as CenterPoint.

4.3. Ablation Study

Components of Query Contrast. We deduce the components of ConQueR to baseline Voxel-DETR by gradually removing multi-positive pairs, auxiliary de-noising loss, and contrastive loss in Table 4. Compared to ConQueR (the first row), removing the multiple noised copies of GTs from contrastive learning (the second row) causes over 0.6 mAPH/L2 performance drop. If we further remove the auxiliary denoising loss (the third row), performances of vehicles and pedestrians classes even become slightly better, indicating that the auxiliary denoising loss alone is not the key for performance improvements. Moreover, we can find that Query Contrast with only original GTs (the second last entry) already improves over the baseline (the last entry) dramatically especially on pedestrians and cyclists. Overall, the Query Contrast scheme brings **1.1**, **1.7**, **2.3** APH/L2 improvements for vehicles, pedestrians and cyclists.

Effects of different supervisions or similarity metrics for GT-query pairs. We demonstrate the effects of different type of supervision or similarity metrics applied to GT-query pairs in Table 5. As discussed in Sec. 3.2, simple geometric relations like GIoU cannot sufficiently measure the similarities between GTs and queries because they cannot take the appearance information into account, thus

| InfoNCE Loss | Aux DN | Multi Pos | APH/L2 | | |
|--------------|--------|-----------|-------------|-------------|-------------|
| | | | Veh. | Ped. | Cyc. |
| ✓ | ✓ | ✓ | 68.2 | 64.7 | 70.1 |
| ✓ | ✓ | | 67.4 (-0.8) | 64.1 (-0.6) | 69.6 (-0.5) |
| ✓ | | | 67.5 (+0.1) | 64.2 (+0.1) | 69.3 (-0.3) |
| | | | 67.1 (-0.4) | 63.0 (-1.2) | 67.8 (-1.5) |

Table 4. Effects of components in Query Contrast. The numbers in brackets denotes the performance drop (red) or increase (blue) for each component. Both the multi-positive contrastive loss (Multi-Pos) and the InfoNCE loss (Eq. (2)) from only original GTs have deep impact on performance, while the auxiliary denoising loss (Aux-DN) only has marginal effects.

| Methods | Veh. | Ped. | Cyc. |
|----------------------------|-------------|-------------|-------------|
| Voxel-DETR | 67.1 | 63.0 | 67.8 |
| ConQueR _{KD-MSE} | 68.1 | 63.4 | 68.2 |
| ConQueR _{QC-GIoU} | 66.6 | 63.6 | 68.4 |
| ConQueR _{QC-Cos} | 68.2 | 64.7 | 70.1 |

Table 5. Effects of different supervisions or similarity metrics applied to GT-query pairs. APH/L2 results are reported. _{QC-Cos} denotes our default Query Contrast with the cosine similarity metric, while _{QC-GIoU} denotes using GIoU as the similarity measurement of GT-query pairs. _{KD-MSE} indicates replacing Query Contrast with Knowledge Distillation MSE loss to supervise positive GT-query pairs only.

only have marginal effects compared to our baseline Voxel-DETR. If we replace Query Contrast with the MSE loss in knowledge distillation (KD) to supervise positive GT-query pairs, performance of vehicles is still comparable with our Query Contrast strategy (the last entry), but it cannot handle densely populated categories like pedestrians and cyclists, indicating the importance of suppressing negative GT-query pairs in our Query Contrast strategy.

Number of positive pairs. We present the results of using different numbers of noised GT copies in Table 6. We observe that using 3 groups of noised copies without original GTs (default setting) achieves the best performance. Moreover, incorporating original GT into the multi-positive contrastive loss harms the performance. The first two entries show that using single noised copies of GTs is better than using the original GTs. We conjecture this is caused by the lack of training for original GT boxes. The detector is only trained to recover from noised GTs, while having no idea how to deal with perfectly located original GTs.

Query-GT feature alignment. We demonstrate the importance of aligning query embeddings to GTs’ with an extra MLP in Table 7. Removing the MLP for query embeddings alignment (the first row) or applying the MLP alignment for both GT and query embeddings (the last row) causes ~ 1 APH/L2 performance drop, indicating the importance of the asymmetric alignment design to mitigate the distribution gap between GT and query embeddings.

| Original GTs | # Noised GT Groups | Veh | Ped | Cyc |
|--------------|--------------------|-------------|-------------|-------------|
| ✓ | 0 | 67.5 | 64.2 | 69.3 |
| | 1 | 67.9 | 64.4 | 69.6 |
| | 2 | 68.2 | 64.3 | 69.9 |
| ✓ | 2 | 67.8 | 64.4 | 68.8 |
| | 3 | 68.2 | 64.7 | 70.1 |
| ✓ | 3 | 68.0 | 64.3 | 69.9 |
| | 4 | 67.7 | 64.4 | 70.1 |

Table 6. Number of positive pairs in the contrastive loss. APH/L2 results are reported on the WOD *validation* split. ✓ denotes including the original GT group into Eq. (2).

| Projection | Veh. | Ped. | Cyc. |
|------------|-------------|-------------|-------------|
| | 67.2 | 64.2 | 69.3 |
| Q | 68.2 | 64.7 | 70.1 |
| G&Q | 67.3 | 64.1 | 68.9 |

Table 7. Design choices of the asymmetric feature alignment. APH/L2 results are reported. ‘G’ and ‘Q’ denotes GT and query embeddings respectively from the selected layer in detector or prediction heads.

| Layer to Contrast | Veh. | Ped. | Cyc. |
|---------------------------|-------------|-------------|-------------|
| Last _{decoder} | 68.1 | 63.9 | 69.7 |
| Last _{FFN} | 68.2 | 64.7 | 70.1 |
| SecondLast _{FFN} | 67.4 | 64.6 | 69.6 |

Table 8. Layers to conduct Query Contrast. Results are the APH/L2 reported on the WOD *validation* split. Last_{decoder} and Last_{FFN} denotes the output layer of each decoder layer and FFN prediction head respectively, while SecondLast_{FFN} indicates the second-last layer of each FFN prediction head is chosen to conduct Query Contrast.

Neural Layers for conducting Query Contrast. We compare 3 layer alternatives to conduct Query Contrast in Table 8: the output layer of each decoder layer, the output layer of each FFN prediction head, and the second-last layer of each FFN prediction head. The Query Contrast scheme can bring consistent improvements for all layer choices, and the features from the last layer of FFN prediction head performs the best, indicating that directly regulate the detection outputs via the contrastive loss can achieve the “enhance-suppress” effects onto queries to the utmost.

Generalisation ability w.r.t. query numbers. We verify the generalization ability of Query Contrast by varying query numbers in Table 9. By default we adopt top-1000 scored proposals as initial queries to input to the transformer decoder. The performance gain of Query Contrast is relatively stable when we gradually reduce query numbers to 500 and 300.

EMA coefficients for generating GT embeddings. Here we show results of different momentums of our EMA decoder, which is used to embed GT boxes, in Table 10. The performance of using the same decoder as queries (the first line) already achieves satisfactory results, while introducing a more stable decoder for GT boxes can further improve the

| Methods | #Query | Veh. | Ped. | Cyc. |
|------------|--------|--------------------|--------------------|--------------------|
| Voxel-DETR | 300 | 66.3 | 62.0 | 66.5 |
| ConQueR | 300 | 67.0 (+0.7) | 63.6 (+1.6) | 68.9 (+2.4) |
| Voxel-DETR | 500 | 66.9 | 62.8 | 67.3 |
| ConQueR | 500 | 67.8 (+0.9) | 64.4 (+1.6) | 69.0 (+1.7) |
| Voxel-DETR | 1000 | 67.1 | 63.0 | 67.8 |
| ConQueR | 1000 | 68.2 (+1.1) | 64.7 (+1.7) | 70.1 (+2.3) |

Table 9. Improvements of Query Contrast under different query numbers. APH/L2 results are reported. The blue numbers in brackets indicates the performance gains.

| Momentum | Veh. | Ped. | Cyc. |
|----------|-------------|-------------|-------------|
| 0.000 | 67.9 | 64.4 | 69.0 |
| 0.900 | 67.6 | 64.3 | 69.1 |
| 0.990 | 68.0 | 64.5 | 69.2 |
| 0.999 | 68.2 | 64.7 | 70.1 |

Table 10. Effects of EMA momentum coefficient.

| τ | Veh. | Ped. | Cyc. |
|--------|-------------|-------------|-------------|
| 1.0 | 67.9 | 64.2 | 69.8 |
| 0.7 | 68.2 | 64.7 | 70.1 |
| 0.5 | 67.6 | 64.5 | 69.7 |

Table 11. Effects of τ . APH/L2 results are reported.

performance especially on categories with fewer instances (*i.e.*, cyclists).

Temperature coefficient in Eq. (2). We shown the effects of different τ in Table 11. τ controls the contrastive learning difficulty of the GT-query similarities, and we find $\tau = 0.7$ leads to the best performance.

5. Conclusion

DETR-based sparse 3D detectors faces the problem of duplicated false positives caused by dense similar queries, and lags in detection performance. In this paper, we solve these challenges with our simple yet effective Query Contrast Voxel-DETR (ConQueR). Based on our sparse 3D detection framework Voxel-DETR, we propose a Query Contrast strategy to explicitly suppress densely overlapping false positives, and simultaneously promote the best matched queries towards their assigned GTs in a contrastive manner. ConQueR reduces $\sim 60\%$ false positives in the final sparse predictions, closes the gap between sparse and dense 3D detectors, and surpasses previous state-of-the-art 3D detectors by a large margin on the challenging WOD benchmark.

Acknowledgement This project is funded in part by National Key RD Program of China Project 2022ZD0161100, by the Centre for Perceptual and Interactive Intelligence (CPII) Ltd under the Innovation and Technology Commission (ITC)’s InnoHK, by General Research Fund of Hong Kong RGC Project 14204021. Hongsheng Li is a PI of CPII under the InnoHK.

References

- [1] 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*, 2022. 1, 2, 3, 5, 6
- [2] John Canny. A computational approach to edge detection. *TPAMI*, 1986. 1
- [3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 2
- [4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 2020. 3, 4
- [5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2, 3
- [6] 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. 6
- [7] Lue Fan, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Fully sparse 3d object detection. *NeurIPS*, 2022. 1, 2
- [8] Zheng Ge, Songtao Liu, Zeming Li, Osamu Yoshie, and Jian Sun. Ota: Optimal transport assignment for object detection. In *CVPR*, 2021. 2
- [9] Benjamin Graham and Laurens van der Maaten. Submanifold sparse convolutional networks. *arXiv*, 2017. 5
- [10] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006. 3
- [11] Miao Hao, Yitao Liu, Xiangyu Zhang, and Jian Sun. Labelenc: A new intermediate supervision method for object detection. In *ECCV*, 2020. 5
- [12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 2, 3
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 5
- [14] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *CVPR*, 2018. 2
- [15] 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 *AAAI*, 2022. 6
- [16] Ding Jia, Yuhui Yuan, Haodi He, Xiaopei Wu, Haojun Yu, Weihong Lin, Lei Sun, Chao Zhang, and Han Hu. Dets with hybrid matching. *arXiv*, 2022. 2
- [17] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 1955. 1
- [18] Christopher Lang, Alexander Braun, and Abhinav Valada. Contrastive object detection using knowledge graph embeddings. *arXiv*, 2021. 3
- [19] 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*, 2022. 2, 4, 6
- [20] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 3
- [21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 4, 5
- [22] 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*, 2014. 1
- [23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2019. 5
- [24] Jiageng Mao, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiashi Feng, Xiaodan Liang, Hang Xu, and Chunjing Xu. Voxel transformer for 3d object detection. In *ICCV*, 2021. 2
- [25] Depu Meng, Xiaokang Chen, Zejjia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *ICCV*, 2021. 2
- [26] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *ICCV*, 2021. 1
- [27] Duy-Kien Nguyen, Jihong Ju, Olaf Booij, Martin R Oswald, and Cees GM Snoek. Boxer: Box-attention for 2d and 3d transformers. In *CVPR*, 2022. 1, 2, 3, 6
- [28] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv*, 2018. 3, 5
- [29] Byungseok Roh, JaeWoong Shin, Wuhyun Shin, and Saehoon Kim. Sparse detr: Efficient end-to-end object detection with learnable sparsity. *ICLR*, 2022. 2
- [30] Hualian Sheng, Sijia Cai, Yuan Liu, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, and Min-Jian Zhao. Improving 3d object detection with channel-wise transformer. In *ICCV*, 2021. 2
- [31] Guangsheng Shi, Ruifeng Li, and Chao Ma. Pillarnet: High-performance pillar-based 3d object detection. *ECCV*, 2022. 1, 2, 5, 6
- [32] 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*, 2020. 1, 2, 5, 6
- [33] 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*, 2021. 1, 2, 5, 6
- [34] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, 2019. 1
- [35] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *AIMLMDO*, 2019. 5
- [36] 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*, 2020. 1, 2, 5
- [37] Pei Sun, Mingxing Tan, Weiyue Wang, Chenxi Liu, Fei Xia, Zhaoqi Leng, and Drago Anguelov. Swformer: Sparse window transformer for 3d object detection in point clouds. In *ECCV*, 2022. 2, 6
- [38] 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*, 2021. 2
- [39] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *CVPR*, 2021. 1, 2
- [40] Zhi Tian, Xiangxiang Chu, Xiaoming Wang, Xiaolin Wei, and Chunhua Shen. Fully convolutional one-stage 3d object detection on lidar range images. *NeurIPS*, 2022. 2
- [41] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *CVPR*, 2021. 5
- [42] Jianfeng Wang, Lin Song, Zeming Li, Hongbin Sun, Jian Sun, and Nanning Zheng. End-to-end object detection with fully convolutional network. In *CVPR*, 2021. 2
- [43] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *AAAI*, 2022. 2
- [44] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 2018. 3
- [45] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *CVPR*, 2020. 2
- [46] Lewei Yao, Renjie Pi, Hang Xu, Wei Zhang, Zhenguo Li, and Tong Zhang. G-detkd: Towards general distillation framework for object detectors via contrastive and semantic-guided feature imitation. In *ICCV*, 2021. 3
- [47] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *CVPR*, 2021. 1, 2, 3, 5, 6
- [48] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv*, 2016. 6
- [49] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv*, 2022. 1, 2
- [50] Peizhen Zhang, Zijian Kang, Tong Yang, Xiangyu Zhang, Nanning Zheng, and Jian Sun. Lgd: Label-guided self-distillation for object detection. In *AAAI*, 2022. 5
- [51] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *CVPR*, 2020. 2
- [52] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, 2018. 1
- [53] Zixiang Zhou, Xiangchen Zhao, Yu Wang, Panqu Wang, and Hassan Foroosh. Centerformer: Center-based transformer for 3d object detection. *ECCV*, 2022. 1, 2, 6
- [54] Benjin Zhu, Junqiang Huang, Zeming Li, Xiangyu Zhang, and Jian Sun. Eqco: Equivalent rules for self-supervised contrastive learning. *arXiv*, 2020. 2
- [55] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv*, 2019. 3
- [56] Benjin Zhu, Jianfeng Wang, Zhengkai Jiang, Fuhang Zong, Songtao Liu, Zeming Li, and Jian Sun. Autoassign: Differentiable label assignment for dense object detection. *arXiv*, 2020. 2
- [57] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *ICLR*, 2020. 1, 2, 3, 4