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

Benjin Zhu\textsuperscript{1} \hspace{1em} Zhe Wang\textsuperscript{1} \hspace{1em} Shaoshuai Shi\textsuperscript{2} \hspace{1em} Hang Xu\textsuperscript{3} \hspace{1em} Lanqing Hong\textsuperscript{3} \hspace{1em} Hongsheng Li\textsuperscript{1,4}

\textsuperscript{1}Multimedia Laboratory, The Chinese University of Hong Kong \hspace{1em} \textsuperscript{2}Max Planck Institute for Informatics \hspace{1em} \textsuperscript{3}Huawei Noah’s Ark Lab \hspace{1em} \textsuperscript{4}Centre for Perceptual and Interactive Intelligence

\{benjin.zhu@link, hsli@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. DE- TRs usually adopt a larger number of queries than GTs (e.g., 300 queries v.s. \(\sim 40\) 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 mAP/L2 on the challenging Waymo Open Dataset validation set, outperforming previous sota methods by over 2.0 mAP/L2. Code

1. Introduction

3D object detection from point clouds has received much attention in recent years \cite{cheng2019fully, qi2017pointnet++, xu2019aggregation, yang2019citrinet, zhou2019voxelnet, zhou2021featnet} as its wide applications in autonomous driving, robots navigation, etc. State-of-the-art 3D detectors \cite{cheng2019fully, qi2017pointnet++, zhou2019voxelnet} still adopt dense predictions with post-processing (e.g., NMS \cite{bodla2017fast}) 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 optimized end-to-end to achieve optimal performance. Recently, DETR-based 2D detectors \cite{carion2020end, he2021mask, li2021deformable, yang2021deformable} show that transformers with direct sparse predictions can greatly simplify the detection pipeline, and lead to better performance. However, although many efforts \cite{carion2020end, lin2021pointed, li2021deformable} 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 \cite{carion2020end, he2021mask, li2021deformable, yang2021deformable}, and resort to the one-to-one Hungarian Matching \cite{kuhn1955hungarian} 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 \cite{carion2020end, li2021deformable} select top-300 scored query predictions to cover only \(\sim 40\) objects in each scene of Waymo Open Dataset (WOD) \cite{sun2019waymo}, while 2D DETR detectors \cite{carion2020end, he2021mask, li2021deformable} use 10\(\times\) more predictions than the average GT number of MS COCO \cite{lin2014microsoft}. 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. Each GT-query pair is 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 quantitively 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~60\% false positives and sets new records on the challenging Waymo Open Dataset (WOD) \([36]\). Contributions are summarized as bellow:

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 adopt 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].

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.
regression: classification, a smooth L1 loss and a 3D GIoU loss for box detection loss. Hungarian Matching to assign GTs to object queries. The previous decoder layer.

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 $L_{det}$ consists of a focal loss [21] for classification, a smooth L1 loss and a 3D GIoU loss for box regression:

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

where $\alpha, \beta, \gamma$ 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 oblige 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-denosing” step alone only has marginal effects to detection performance, while our multi-positive 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 \(\{b_1^i, b_2^i, \ldots, b_T^i\}\), and denote \(K\) query embeddings as \(\{q_1, q_2, \ldots, q_K\}\). 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 \(L_i^{QC}\) can be formulated as

\[
L_i^{QC} = - \sum_{t=1}^{T} \log \left( \frac{\exp\left(\cos(b_t^i, q_j) / \tau \right)}{\sum_{k=1}^{K} \exp\left(\cos(b_t^i, q_k) / \tau \right)} \right),
\]

where \(\tau\) 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 each 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, \(1 \geq 5\) and LEVEL, \(2 \geq 1\). 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., \(\geq 0.1\)) 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
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 $2 \times$ 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. $t_s$ denotes the two-stage model.

<table>
<thead>
<tr>
<th>Methods</th>
<th>mAP/mAPH L2</th>
<th>Vehicle 3D AP/APH L2</th>
<th>Pedestrian 3D AP/APH L2</th>
<th>Cyclist 3D AP/APH L2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dense Detectors</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CenterPoint$_{ts}$ [47]</td>
<td>-/67.4</td>
<td>-/67.9</td>
<td>-/65.6</td>
<td>-/68.6/-</td>
</tr>
<tr>
<td>PV-RCNN [32]</td>
<td>66.8/63.3</td>
<td>69.0/68.4</td>
<td>77.5/76.9</td>
<td>75.0/65.6</td>
</tr>
<tr>
<td>AFDetV2 [15]</td>
<td>71.0/68.8</td>
<td>70.7/69.2</td>
<td>72.7/71.1</td>
<td>70.0/67.4</td>
</tr>
<tr>
<td>SST,TS [6]</td>
<td>-/-</td>
<td>68.0/67.6</td>
<td>72.8/65.9</td>
<td>72.5/64.9</td>
</tr>
<tr>
<td>SWFormer [37]</td>
<td>-/-</td>
<td>69.2/68.8</td>
<td>72.8/65.9</td>
<td>70.0/67.4</td>
</tr>
<tr>
<td>PillarNet-34 [31]</td>
<td>71.0/68.5</td>
<td>70.9/70.5</td>
<td>72.8/65.9</td>
<td>70.0/67.4</td>
</tr>
<tr>
<td>CenterFormer [53]</td>
<td>71.2/69.0</td>
<td>70.2/69.7</td>
<td>72.8/65.9</td>
<td>70.0/67.4</td>
</tr>
<tr>
<td>PV-RCNN++ [33]</td>
<td>71.7/69.5</td>
<td>70.6/70.2</td>
<td>72.8/65.9</td>
<td>70.0/67.4</td>
</tr>
<tr>
<td><strong>Sparse Detectors</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BoxeR-3D [27]</td>
<td>-/-</td>
<td>63.9/63.7</td>
<td>70.4/70.0</td>
<td>61.5/53.7</td>
</tr>
<tr>
<td>TransFusion-L [1]</td>
<td>-/64.9</td>
<td>-/65.1</td>
<td>-/63.7</td>
<td>-/-</td>
</tr>
<tr>
<td>Voxel-DETR (ours)</td>
<td>68.8/66.1</td>
<td>67.8/67.2</td>
<td>75.4/74.9</td>
<td>69.7/63.1</td>
</tr>
<tr>
<td>ConQueR (ours)</td>
<td>70.3/67.7</td>
<td>68.7/68.2</td>
<td>76.1/75.6</td>
<td>70.9/64.7</td>
</tr>
<tr>
<td>ConQueR †(ours)</td>
<td>73.1/70.6</td>
<td>71.0/70.5</td>
<td>78.4/77.9</td>
<td>73.7/68.1</td>
</tr>
<tr>
<td>ConQueR ‡(ours)</td>
<td>74.0/71.6</td>
<td>71.0/70.5</td>
<td>78.4/77.9</td>
<td>75.8/70.1</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Methods</th>
<th>All</th>
<th>Veh.</th>
<th>Ped.</th>
<th>Cyc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CenterPoint [47]</td>
<td>69.0</td>
<td>71.9</td>
<td>67.9</td>
<td>68.2</td>
</tr>
<tr>
<td>PV-RCNN++ [33]</td>
<td>70.2</td>
<td>73.5</td>
<td>69.0</td>
<td>68.2</td>
</tr>
<tr>
<td>AFDetv2 [15]</td>
<td>70.0</td>
<td>72.6</td>
<td>68.6</td>
<td>68.7</td>
</tr>
<tr>
<td>PillarNet-34 [31]</td>
<td>69.6</td>
<td>74.7</td>
<td>68.5</td>
<td>65.5</td>
</tr>
<tr>
<td>ConQueR (Ours)</td>
<td><strong>72.0</strong></td>
<td>73.3</td>
<td><strong>70.9</strong></td>
<td><strong>71.9</strong></td>
</tr>
</tbody>
</table>

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 $\sim$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 $\sim$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 $\sim$6.0 mAPH/L2, closing the performance gap between sparse and dense 3D detectors. When compared with our baseline Voxel-DETR, the proposed Query Contrast mechanism brings over 1.6 mAPH/L2 without any extra inference cost. Besides, our baseline Voxel-DETR with only 6 epochs of training outperforms previous sparse 3D detectors, and achieves comparable performance with CenterPoint (36-epoch training) with only 1/6 GPU hours. In addition, ConQueR has an inference latency of 70ms (46ms for CenterPoint$^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 $\sim$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 $t_s$ denotes the two-stage model.

$^1$Latency is measured with batch size 1 on NVIDIA A100 GPU.
Table 3. Sparsity of final predictions. APH/L2 results are reported on the WOD validation set. The subscripts of each entry denote the way they obtain final predictions. For example, CenterPoint uses NMS to filter out duplicated boxes, and Voxel-DETR uses it uses top-N scored proposals as final predictions, while ConQueR uses score thresholding to generate final sparse predictions. \( \dagger \) denotes our best model in Table 1.

### 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 GloU cannot sufficiently measure the similarities between GTs and queries because they cannot take the appearance information into account, thus

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.

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−GloU denotes using GloU 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.
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 performance especially on categories with fewer instances (i.e., cyclists).

Temperature coefficient in Eq. (2). We shown the effects of different $\tau$ in Table 11. $\tau$ 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 1420421. Hongsheng Li is a PI of CPII under the InnoHK.
References


[15] Yihan Hu, Zhuangzheng Ding, Runzhou Ge, Wenxin Shao, Li Huang, Kun Li, and Qiang Liu. Afldetv2: 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. Detrs with hybrid matching. arXiv, 2022. 2


[27] Duy-Kien Nguyen, Jihong Ju, Olaf Booj, Martin R Oswald, and Cees GM Snoek. Boxer: Box-attention for 2d and 3d transformers. In CVPR, 2022. 1, 2, 3, 6


[34] Shaoshuai Shi, Xiaoang Wang, and Hongsheng Li. Pointr-cnn: 3d object proposal generation and detection from point cloud. In CVPR, 2019. 1


[36] Pei Sun, Henrik Kretzschmar, Xavier Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou,
[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
[41] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In CVPR, 2020. 2
[45] 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