Enhancing 3D Object Detection with 2D Detection-Guided Query Anchors

Haoxuanye Ji[2,*,♯] Pengpeng Liang[1,*,†] Erkang Cheng[2,‡]
[1]School of Computer and Artificial Intelligence, Zhengzhou University [2]Nullmax
jihaoxuanye@163.com, {liangpcs, twokang.cheng}@gmail.com

Abstract

*Multi-camera-based 3D object detection has made notable progress in the past several years. However, we observe that there are cases (e.g. faraway regions) in which popular 2D object detectors are more reliable than state-of-the-art 3D detectors. In this paper, to improve the performance of query-based 3D object detectors, we present a novel query generating approach termed QAF2D, which infers 3D **q**uery **a**nchors **f**rom **2D** detection results. A 2D bounding box of an object in an image is lifted to a set of 3D anchors by associating each sampled point within the box with depth, yaw angle, and size candidates. Then, the validity of each 3D anchor is verified by comparing its projection in the image with its corresponding 2D box, and only valid anchors are kept and used to construct queries. The class information of the 2D bounding box associated with each query is also utilized to match the predicted boxes with ground truth for the set-based loss. The image feature extraction backbone is shared between the 3D detector and 2D detector by adding a small number of prompt parameters. We integrate QAF2D into three popular query-based 3D object detectors and carry out comprehensive evaluations on the nuScenes dataset. The largest improvement that QAF2D can bring about on the nuScenes validation subset is 2.3% NDS and 2.7% mAP. Code is available at* https://github.com/nullmax-vision/QAF2D.

1. Introduction

3D object detection with multi-view images captured by surrounding cameras plays an important role in autonomous driving systems, and camera-based approaches have the benefit of low deployment cost in comparison to LIDAR-based approaches [4, 7, 32]. Though notable progress has been made in the past several years [12, 18, 23, 29, 33, 40], multi-camera-based 3D object detection is still a challenging task due to the lack of depth information and the small object size in faraway regions.

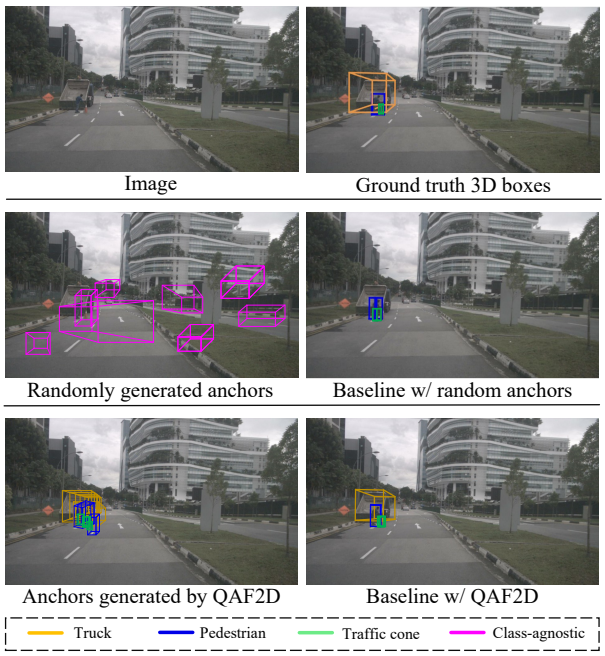


Figure 1. Comparison between randomly generated anchors and anchors generated by our QAF2D and comparison between their corresponding detection results. We use StreamPETR [33] as the baseline. Note that for illustration purpose, we just draw part of the anchors to alleviate clutter.

Inspired by the promising performance of query-based 2D object detectors [14, 21, 26, 27, 41], query-based strategy [27] has been explored by several recent works for multi-camera-based 3D object detection [18, 19, 23, 24, 33, 35]. DETR3D [35] projects a set of sparse 3D object queries to the 2D images for image feature aggregation. PETR [23, 24] constructs queries based on 3D points to interact with 3D position-aware image features. SparseBEV [19] initializes a set of sparse queries based on pillars in bird's-eye-view (BEV) space which is used to sample multi-view image features of several frames. StreamPETR [33] is built upon [23] with query propagation for temporal information modeling. BEVFormer [18] adopts a 3D detection head based on Deformable DETR [41] after constructing the BEV feature

*Equal contribution. ♯Work done during an internship at Nullmax.
†Project lead. ‡Corresponding author.

maps. Despite these approaches bringing about meaningful performance improvement, we observe there are some cases for which popular 2D detectors can handle successfully but 3D detectors fail.

There are a few approaches [11, 37, 40] trying to boost the 3D detection performance with the aid of 2D detectors. To generate 3D proposals, Far3D [11] lifts the 2D detection bounding boxes to 3D with the depth estimated by a separate network. MV2D [37] infers 3D reference points for object query generation by first transforming each 2D bounding box to a 2.5D point with RoI image features of the 2D box. A problem of [11, 37] is that their approaches to lift 2D results involve inferring depth information from images, which itself is a challenging task. SimMOD [40] uses 2D bounding box detection as an auxiliary task during training to improve the perception of fine-grained structures, but it cannot use 2D detection results to directly provide guidance for 3D object detectors.

Considering the above problems, in this paper, we propose an approach named QAF2D to generate 3D query anchors from 2D detection bounding boxes to improve the performance of query-based 3D object detectors. More specifically, to lift the 2D bounding box of an object in an image to a set of 3D anchors, as the projection of the object's 3D center is within the 2D box, we first uniformly sample a set of projected centers inside the 2D box, and for each center, we associate it with depth, 3D size, and yaw angle candidates to generate 3D anchors. The set of 3D sizes chosen for each 2D detected box depends on the class of the box. Then, each 3D anchor in the initial set is projected back to the image, and the IoU between the projected box and the corresponding 2D box is calculated, and only anchors having IoUs larger than a threshold are used to construct 3D queries. To make use of the class information of the 2D bounding boxes, for the calculation of the set-to-set loss based on DETR [27], we associate each query with the class of its corresponding 2D box, and each predicted 3D bounding box can only be matched with ground truth boxes that have the same class. Fig. 1 illustrates the advantage of the proposed anchor generation method.

To reduce the computation cost while keeping the performance of the 3D detector as intact as possible, the image feature extraction backbone is shared by the 3D detector and 2D detector via prompt tuning [1, 10], and the network is trained in two stages. In the first stage, the 3D detector is trained, and the 2D detection results are obtained by projecting the 3D ground truth to the multi-view images. In the second stage, only the prompt parameters and 2D detection head are trained with all other parameters frozen. We integrate the proposed QAF2D into three query-based 3D detectors (StreamPETR [33], SparseBEV [19], and BEVFormer [18]) and carry out comprehensive experiments on the nuScenes dataset [2]. The performance of all three query-based 3D detectors can be improved, an average improvement of 1.18% NDS and 1.74% mAP is achieved on the nuScenes validation subset, and the largest improvement is 2.3% NDS and 2.7% mAP.

The contributions of our paper are summarized as follows:

- We propose to generate 3D query anchors from 2D bounding boxes so that the results of the more reliable 2D detector can be directly used to improve the 3D detection performance.
- We share the image feature extraction backbone between the 3D and 2D detectors by visual prompts for efficiency and successfully train the network in two stages.
- Consistent performance improvement is achieved on the nuScenes dataset when the proposed QAF2D is integrated into three query-based 3D object detectors, and it shows the effectiveness and generalization ability of our proposed approach.

2. Related Work

2.1. Camera-based 3D Object Detection

Camera-based 3D detectors aim to predict the 3D bounding boxes of objects in camera images. Some approaches [3, 25, 34, 39] focus on the monocular setting. FCOS3D [34] utilizes a fully convolutional single-stage network to regress 3D object information directly without using any 2D-3D correspondence priors. [25] identifies the localization error as a key factor that constrains the detection performances and proposes strategies to alleviate it. MonoPair [3] leverages spatial constraints between paired objects to deal with occlusion. MonoDETR [39] enhances the vanilla Transformer with the contextual depth cues to guide the 3D detection process.

Recently, 3D object detection under the surrounding multiple-camera setting has attracted a lot of research efforts. One line of approaches [12, 15–18, 29] transforms the image features to the BEV space with the help of depth estimation before applying a detection head. LSS [29] utilizes an estimated depth distribution of each pixel to lift the features of each image individually into a frustum and converts the frustums of all images into a BEV grid. BEVDepth [17] uses the intrinsic camera parameters as one of the inputs of the depth estimation module with supervision from point cloud to predict depth of images for BEV feature construction. BEVStereo [16] designs an effective depth estimation method based on temporal stereo to build BEV features. BEVFormer [18] proposes to aggregate features from both the spatial and temporal spaces to the current BEV space with learnable queries. PolarFormer [12] builds the BEV features in the polar coordinate system to consider the wedge shape of the physical world under the ego car's perspective. DFA3D [15] proposes to use 3D deformable attention to aggregate the lifted features in 3D space so that the depth

ambiguity problem can be mitigated.

Another line of approaches [24, 33, 35, 37, 38] directly samples image features with queries and uses a decoder network to detect objects. DETR3D [35] proposes to use a set of sparse object queries to implicitly transform features from 2D to 3D without estimating dense 3D scene geometry. PETR [23] generates 3D position-aware image features and then uses a set of queries to interact with the features and predict 3D bounding boxes. PETRv2 [24] constructs the 3D position embedding in a data-dependent way, and temporal information is exploited by transforming the coordinates in the previous frame to the current coordinate system. StreamPETR [33] makes use of temporal information by propagating selected queries from a memory queue to the current frame, and a motion-aware layer normalization is designed as well. SparseBEV [19] lets the queries interact with image features in a sparse manner with promising performance by designing scale-adaptive self-attention and adaptive spatio-temporal sampling modules. MV2D [37] proposes to learn queries based on 2D detection results, which later interact with RoI image features. CAPE [38] adopts a local camera-view coordinate system instead of a global one to form 3D position embeddings so that variances caused by changes of camera extrinsic parameters can be eliminated.

2.2. Query-based 2D Object Detection

DETR [27] is the first Transformer-based [6] object detection approach, and it uses a set of object queries to interact with images features and constructs loss via bipartite matching. Many subsequent works [14, 21, 22, 26, 36, 41] have proposed to improve DETR. Deformable-DETR [41] proposes deformable attention that only attends to a small number of key sampling points of a reference. To accelerate the convergence, Conditional-DETR [26] disentangles the content and spatial queries and predicts conditional spatial queries from the decoder embedding. Anchor-DETR [36] uses anchor points to build queries so that each query can focus on a specific region. DAB-DETR [21] proposes to construct queries with dynamic learnable anchor boxes which are updated layer-by-layer. DN-DETR [14] reduces the instability of bipartite graph matching by reconstructing ground truth boxes from noisy queries. [22] proposes to only use positional metrics to stabilize the matching process of the DETR loss.

2.3. Visual Prompt Tuning

Prompting is initially proposed to modify the input text string so that a pre-trained large language model can be adapted to new tasks with few or no labeled data [20]. CLIP [30] uses prompts to transfer visual models trained with natural language supervision to downstream tasks under the zero-shot setting. Recently, some works [1, 5, 8, 10] propose to adapt a pre-trained visual model to different domains by adding a small amount of prompt parameters instead of fine-tuning the entire model. VPT [10] utilizes a small amount of trainable parameters to adapt large-scale Transformer models to downstream tasks instead of full fine-tuning. LPT [5] introduces a shared prompt and group-specific prompts into a frozen pre-trained model to adapt to long-tailed data. [1] transforms the input image with prompts so that a frozen pre-trained model can perform new tasks. E^{2}VPT [8] proposes to use learnable key-value prompts and visual prompts with a prompt pruning procedure for effective and efficient fine-tuning.

3. Method

3.1. Overall Architecture

As shown in Fig. 2, at timestamp t, the captured multi-camera images $I_t = \{I_c^t\}_{c=1}^{N_{\text{cam}}}$ are input into an feature extraction backbone (e.g. ResNet [9] or VovNet [13]) to extract image features $F_t = \{F_c^t\}_{c=1}^{N_{\text{cam}}}$, where N_{cam} is the number of cameras. We first feed F_t to the 2D detection branch to obtain 2D bounding boxes. The 3D anchor generator generates a set of 3D anchors for each 2D bounding box with its corresponding camera's intrinsic and extrinsic parameters and its class information. Then, the query-based 3D detector takes F_t and the generated 3D anchors as input to predict 3D bounding boxes. We train the network in two stages.

3.2. 3D Query Anchor Generation

Given the 2D detection bounding boxes $B = \{(\mathbf{b}_i, g_i)\}_{i=1}^{N}$ of image I, where $\mathbf{b}_i = (x_i, y_i, w_i, h_i)$ represents a bounding box with its center coordinate, width, and height, and g_i is its class information, we associate each sampled point in $\mathbf{b}_i$ with 3D size, depth and yaw angle candidates, and generate a set of 3D query anchors for $\mathbf{b}_i$ following the steps described below. The process is shown on the right side of Fig. 2.

3D anchor center candidates. Since the projection of the center of an object to an image plane is always within the minimum unrotated rectangle that contains the entire object, given a 2D bounding box $\mathbf{b}_i = (x_i, y_i, w_i, h_i)$, we first sample a set of 2D object centers in the following way:

$$\begin{aligned} C_{\text{2D}} = \{(x_s, y_s) | x_s &= x_{\min} + s_x \times i_x, \\ y_s &= y_{\min} + s_y \times i_y; \\ i_x, i_y &\in \mathbb{Z}_{\geq 0}, x_s \leq x_{\max}, y_s \leq y_{\max}\}, \end{aligned} \tag{1}$$

where s_x and s_y are step sizes, $x_{\min} = \lfloor x_i - w_i/2 \rfloor$, $x_{\max} = \lfloor x_i + w_i/2 \rfloor$, $y_{\min} = \lfloor y_i - h_i/2 \rfloor$, and $y_{\max} = \lfloor y_i + h_i/2 \rfloor$. Then, we define a set of depth candidates $D = \{d_0, ..., d_{N_D - 1}\}$ of size N_D and associate each point in C_{2D} with the depth candidates to generate $C'_{\text{2D}} = \{(x_s, y_s, d_s) | (x_s, y_s) \in C_{\text{2D}}, d_s \in D\}$. Next, we transform each point in C'_{2D} to the 3D coordinate system and

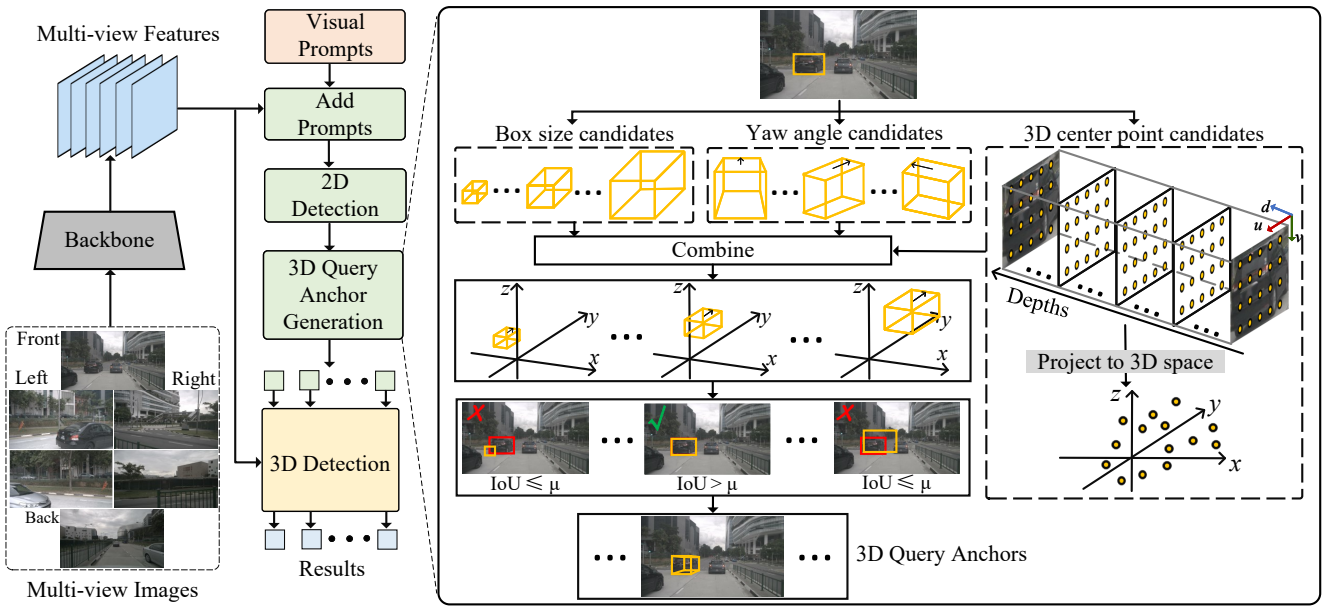


Figure 2. Overview of the 3D detection pipeline with our proposed 3D query anchor generation approach. The image backbone network extracts features of the input multi-view images, and the features are shared between the 3D detector and 2D detector with visual prompts. 2D detection results are used to generate 3D query anchors. Our 3D anchor generation method first generates box size candidates, yaw angle candidates, and 3D center point candidates, and then combines them to construct an initial set of anchors, which is refined with IoU check to form the final set of 3D query anchors. The entire network is optimized in two stages.

obtain a 3D object center set $C_{3D} = \{(x_e, y_e, z_e)\}$. The transformation is carried out with the intrinsic parameters $\mathbf{K}$ and extrinsic parameters $\mathbf{R}$ of the corresponding camera of (x_s, y_s, z_s) using the formula below:

$$\begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} = \mathbf{R}^{-1}\mathbf{K}^{-1} \begin{bmatrix} x_s \cdot d_s \\ y_s \cdot d_s \\ d_s \end{bmatrix} \tag{2}$$

3D anchor size candidates. As the 3D object sizes of different classes can vary considerably while the change of object size of the same class is usually small, we select a bounding box $\mathbf{b}_i$'s 3D size candidates based on its class g_i. For each class g, we decide its width range $(w_{\mathrm{g}}^{\min}, w_{\mathrm{g}}^{\max})$, height range $(h_{\mathrm{g}}^{\min}, h_{\mathrm{g}}^{\max})$, and length range $(l_{\mathrm{g}}^{\min}, l_{\mathrm{g}}^{\max})$ by traversing all objects of class g in the training data and choosing the maximum and minimum values. Based on the above ranges, we construct the sets of width, height, and length as follows:

$$\begin{aligned} W_g &= \{w | w = w_{\mathrm{g}}^{\min} + s_w \times i; i \in \mathbb{Z}_{\geq 0}, w \leq w_{\mathrm{g}}^{\max}\}, \\ H_g &= \{h | h = h_{\mathrm{g}}^{\min} + s_h \times i; i \in \mathbb{Z}_{\geq 0}, h \leq h_{\mathrm{g}}^{\max}\}, \\ L_g &= \{l | l = l_{\mathrm{g}}^{\min} + s_l \times i; i \in \mathbb{Z}_{\geq 0}, l \leq l_{\mathrm{g}}^{\max}\} \end{aligned} \tag{3}$$

where s_w, s_h, s_l are the step sizes. Then, the set of 3D object size candidates S_g for class g are generated by combining the above three sets:

$$S_g = \{(w, h, l) | w \in W_g, h \in H_g, l \in L_g\} \tag{4}$$

Yaw angle candidates. We construct the set of yaw angle candidates by uniformly sampling in $[0, 2\pi)$ with interval π / N_θ, and the set Θ is defined below:

$$\Theta = \{\theta | \theta = \frac{n_\theta}{N_\theta}\pi, n_\theta = 0, 1, 2, ..., 2N_\theta - 1\} \tag{5}$$

Generating 3D query anchors. Given the 3D center candidate set C_{3D}, size candidate set S_{g}, and yaw angle candidate set Θ of a 2D bounding box $\mathbf{b}$ of class g, we generate an initial set of 3D anchors via the Cartesian product of the three candidate sets, i.e. $P_{\mathrm{Init}} = \{\mathbf{p}_i | \mathbf{p}_i = (x_i, y_i, z_i, w_i, h_i, l_i, \theta_i)\} = C_{3D} \times Sg \times \Theta$. To remove the anchors that are not compatible with $\mathbf{b}$, we project each $\mathbf{p}_i$ to the image plane of the 2D bounding box $\mathbf{b}$ and get $\mathbf{p}_i^{2D} = (x_i^{2D}, y_i^{2D}, w_i^{2D}, h_i^{2D})$. Then, the IoU between $\mathbf{p}_i^{2D}$ and $\mathbf{b}$ is calculated, and only $\mathbf{p}_i$ with IoU larger than the threshold μ is retained. The final set of query anchors is $P = \{\mathbf{p}_i | \mathrm{IoU}(\mathbf{p}_i^{2D}, \mathbf{b})) > \mu\}$. We use P as the decoder input of three selected query-based 3D object detectors to show its effectiveness.

3.3. Two-stage Optimization with Visual Prompts

To enable the 2D detection branch (DAB-DETR [21] is used as the default 2D detector) to share the image feature extraction backbone with the 3D detection branch without compromising the performance of the 3D detector, we train the network in two stages with visual prompts [1] added to the 2D detection branch.

Stage 1: Training 3D detection branch. For training, instead of using the output of the 2D detection branch, we project the ground truth 3D bounding boxes to the images to get 2D bounding boxes. Then, the proposed 3D query anchor generation method is used to generate 3D query anchors which are used as the input to the decoder of the query-based 3D object detector (based on StreamPETR [33], SparseBEV [19], or BEVFormer [18]). To take advantage of the class information of 2D bounding boxes, when calculating the set-to-set loss [27] of the query-based 3D detector, the predicted 3D bounding box of a query is only matched with a ground truth box that has the same class as the query's corresponding 2D box. More specifically, we divide the set of predicted bounding boxes $\hat{B}$ and the set of ground truth bounding boxes B into $\{\hat{B}_g\}_{g=1}^{G}$ and $\{B_g\}_{g=1}^{G}$, respectively, where G is the number of classes. Given a set pair $(\hat{B}_g, B_g)$, boxes in $\hat{B}_g$ are matched with boxes in B_g by Hungarian algorithm following [27], and the cost matrix is calculated based on the predicted probability of the ground truth class and the L_1 loss between the predicted and ground truth 3D bounding boxes.

Stage 2: Training 2D detection branch with visual prompts. After training the image feature backbone with the 3D detection task in Stage 1, the backbone is frozen, and the 2D detection branch uses the same image features as the 3D detector. To adapt the image features to the 2D detection task, we add visual prompts designed in [1] to the feature maps. It is worth noting that [1] adds prompts to the input images instead of feature maps as do we. We adopt the padding prompt design of [1]. More specifically, given a feature map of size $C \times H \times W$, where C, H, and W are the number of channels, height, and width, respectively. Two prompt patches of size $C \times (\tau \times H) \times W$ are added to the top and bottom of the feature map, respectively. Another two prompt patches of size $C \times (H - 2\tau \times H) \times (\tau \times W)$ are added to the left and right sides of the feature map, respectively. The total number of prompt parameters varies with the change of τ. For the training of the 2D detection branch, only the prompt parameters and the head of the 2D detector are updated.

In the test phase, the 2D bounding boxes predicted by the 2D detection branch are used to generate the 3D query anchors for the query-based 3D detector.

3.4. Integrating into Query-based 3D Detectors

We integrate our approach into three selected query-based 3D object detectors (StreamPETR [33], SparseBEV [19], BEVFormer [18]) by replacing the randomly initialized anchors (or learnable queries) with our proposed 3D query anchors inferred from 2D bounding boxes.

StreamPETR. StreamPETR [33] is built upon the query-based 3D object detector PETR [23]. To make use of temporal information efficiently, it maintains a memory queue of historical object queries. The queries of the current frame consist of selected queries from the memory queue and newly added queries. The new queries depend on a set of learnable 3D anchor points initialized with uniform distribution between 0 and 1. We integrate the proposed 3D query anchors into StreamPETR by simply substituting a set of anchors $\{(x_i, y_i, z_i, w_i, l_i, h_i, \sin\theta_i, \cos\theta_i)\}$ inferred from 2D bounding boxes for the original learnable 3D anchor points.

SparseBEV. Besides adopting sparse queries, SparseBEV [19] removes the dense global attention between queries and image features of [23], and it proposes an adaptive spatio-temporal sampling method to aggregate image features. SparseBEV defines a set of learnable queries, and each query represents an object's translation, dimension, rotation, and velocity. To integrate our 3D query anchors, we replace the learnble queries with $\{(x_i, y_i, z_i, w_i, l_i, h_i, \sin\theta_i, \cos\theta_i)\}$ generated from 2D detection results.

BEVFormer. In [18], BEV features are constructed by a spatial cross-attention between predefined BEV queries and multi-camera image features, which are used as input to a modified Deformable DETR [41] for 3D object detection. To apply the 3D query anchors, we design a 3D detection head based on DAB-Deformable-DETR [21], and the learnable dynamic anchors in the form of (x, y, h, w) are replaced with 3D query anchors in the form of $(x, y, z, w, l, h, \sin\theta, \cos\theta)$, and the decoder predicts 3D bounding boxes and velocity rather than 2D bounding boxes.

4. Experiments

4.1. Dataset and Metrics

We conduct experiments on the nuScene dataset [2]. It consists of 1000 multi-modal videos each of which is about 20s, and keyframes are annotated every 0.5s. The sensors include camera, LIDAR, and RADAR, and we use the images captured by the six surrounding cameras for our experiments. The videos are split into three subsets of 750, 150, and 150 for training, validation, and test, respectively. There are 1.4M annotated 3D bounding boxes of 10 classes in total.

We use the official evaluation metrics of nuScenes. Along with mean average precision (mAP), the following true positive errors are reported: average translation error (ATE), average scale error (ASE), average orientation error (AOE), average velocity error (AVE), and average attribute error (AAE). In addition, a more comprehensive nuScenes detection score (NDS) is derived from the above metrics.

4.2. Implementation Details

We implement our method with PyTorch [28]. After integrating our QAF2D into a base detector, we use the base detector's data augmentation strategy and training setting (e.g. learning rate, batch size, number of epochs) for training.

Methods	Backbone	Image Size	NDS	mAP	mATE	mASE	mAOE	mAVE	mAAE
StreamPETR [33]	V2-99	320×800	57.1	48.2	61.0	25.6	37.5	26.3	19.4
StreamPETR-8DQuery		320×800	57.6	48.6	58.9	25.7	37.7	25.5	19.5
StreamPETR-QAF2D (Ours)		320×800	**58.6**	**50.0**	56.1	26.1	36.9	25.1	19.6
StreamPETR [33]	ResNet50	320×800	54.0	43.2	58.1	27.2	41.3	29.5	19.5
StreamPETR-8DQuery		320×800	54.2	44.0	62.1	27.1	41.1	26.5	21.0
StreamPETR-QAF2D (Ours)		320×800	**54.6**	**44.7**	62.3	26.9	41.0	27.7	19.5
StreamPETR*‡ [33]	ResNet50	320×800	55.0	45.0	61.3	26.7	41.3	26.5	19.6
StreamPETR*‡-8DQuery		320×800	55.2	45.5	61.0	27.1	40.1	27.6	20.1
StreamPETR*‡-QAF2D (Ours)		320×800	**56.2**	**46.5**	61.5	26.2	36.0	26.5	19.9
SparseBEV [19]	ResNet50	256×704	55.8	44.8	58.1	27.1	37.3	24.7	19.0
SparseBEV-QAF2D (Ours)		256×704	**56.1**	**46.0**	57.3	26.3	38.7	27.6	19.1
BEVFormer-small [18]	ResNet101-DCN	736×1280	47.9	37.0	72.1	28.0	40.7	43.6	22.0
BEVFormer-small-DAB3D		736×1280	49.2	39.0	71.7	27.5	41.6	42.2	19.7
BEVFormer-small-QAF2D (Ours)		736×1280	**50.2**	**39.7**	70.3	27.4	36.9	40.4	21.3

Table 1. Comparison of the base detectors and their QAF2D enhanced version on the nuScenes validation split. $*$ indicates the use of perspective-view pre-training. ‡ represents the use of 300 randomly initialized queries (irrelevant to QAF2D) and 128 propagation queries. Please refer to the corresponding text in Sec. 4.3 for the meaning of "8DQuery" and "DAB3D". The best is in **bold**.

Method	Backbone	Image Size	NDS	mAP	mATE	mASE	mAOE	mAVE	mAAE
DETR3D [35]	V2-99	900×1600	47.9	41.2	64.1	25.5	39.4	84.5	13.3
BEVFormer [18]	V2-99	900×1600	56.9	48.1	58.2	25.6	37.5	37.8	12.6
PolarFormer [12]	V2-99	900×1600	57.2	49.3	55.6	25.6	36.4	43.9	12.7
PETRv2 [24]	V2-99	640×1600	58.2	49.0	56.1	24.3	36.1	34.3	12.0
CAPE [38]	V2-99	640×1600	61.0	52.5	50.3	24.2	36.1	30.6	11.4
MV2D [37]	V2-99	–	59.6	51.1	52.5	24.3	35.7	35.7	12.0
SparseBEV [19]	V2-99	640×1600	62.7	54.3	50.2	24.4	32.4	25.1	12.6
SparseBEV (dual-branch) [19]	V2-99	640×1600	63.6	55.6	48.5	24.4	33.2	24.6	11.7
StreamPETR [33]	V2-99	640×1600	63.6	55.0	47.9	23.9	31.7	24.1	11.9
StreamPETR-8DQuery	V2-99	640×1600	63.6	55.5	47.1	23.6	32.2	26.8	11.8
StreamPETR-QAF2D (Ours)	V2-99	640×1600	**64.2**	**56.6**	46.1	24.0	32.6	26.1	12.1

Table 2. Comparison with the state-of-the-art approaches on the nuScene test split. The best is in **bold**.

All models are trained with 8 NVIDIA GeForce RTX 3090 GPUs. For the parameters regarding 3D anchors generation in Section 3.2, s_x and s_y that control the sampling intervals of 2D object centers are set to 10. The depth candidates in D are sampled between 3 meters and 103 meters with an interval of 1.5 meters. The width range, height range and length range of each class are given in the appendix. We set the sampling intervals s_w, s_h and s_l to 5, and N_θ that controls the sampling interval of yaw angle is set to 12. The IOU threshold μ for anchor validation is 0.99, i.e. only 3D anchors, the projections of which have high overlap with their corresponding 2D boxes, are kept. The parameter τ related to the number of prompt parameters in Section 3.3 is set to 0.2 based on the ablation study.

4.3. Effectiveness of QAF2D

Validation on nuScenes val split. To verify the effectiveness of the 3D query anchors generated by the proposed QAF2D, we compare StreamPETR [33], SparseBEV [19], and BEVFormer [18] with their QAF2D-enhanced version on the nuScenes validation split. For fair comparison of StreamPETR, besides its original results using three dimensional random queries, we also report the results of StreamPETR with eight dimensional random queries (8DQuery), which have the same dimension as queries of QAF2D. For fair comparison of BEVFormer-small, along with its original result based on modified Deformable DETR [41], we report the result of BEVFormer-small with our modified DAB-Deformable-DETR [21] (BEVFormer-small-DAB3D) as well. BEVFormer-small-DAB3D uses eight dimensional randomly initialized queries. For SparseBEV, as its queries are nine dimensional containing velocity information, we do not change queries to eight dimensional ones. The results in Table 1 demonstrate that our QAF2D can bring about consistent improvement for all three base detectors. With regard to StreamPETR (V2-99 backbone [13]), after improving the performance of StreamPETR by 0.5% NDS and 0.4% mAP with eight dimensional queries, our proposed QAF2D can obtain additional improvement of 1.0% NDS and 1.4% mAP in comparison with StreamPETR-8DQuery,

	Backbone	2D detection	3D anchor generation	3D detection	Total	Speed
StreamPETR	45ms	–	–	8ms	53ms	18.9 FPS
StreamPETR-QAF2D	47ms	12ms	1ms	5ms	65ms	15.4 FPS

Table 3. Component time consumption and speed comparison between StreamPETR and StreamPETR-QAF2D with V2-99 backbone on an NVIDIA 3090 GPU.

and the entire improvement is 1.5% NDS and 1.8% mAP. We also evaluate StreamPETR and its enhanced version with ResNet50 backbone [9], and our QAF2D can gain an improvement of 0.6% NDS and 1.5% mAP. In addition, we incorporate QAF2D into StreamPETR*‡ (benefits from perspective-view pre-training and uses different number of queries [33]), and QAF2D can improve its performance by 1.2% NDS and 1.5% mAP. For SparseBEV, our proposed QAF2D can improve its performance by 0.3% NDS and 1.2% mAP. As to BEVFormer, it can be first improved with our modified DAB-Deformable-DETR by 1.3% NDS and 2.0% mAP, and the proposed OAF2D can gain another improvement of 1.0% NDS and 0.7% mAP compared with BEVFormer-small-DAB3D, and the total improvement is 2.3% NDS and 2.7% mAP.

Comparison with state-of-the-art on nuScenes test split. We compare StreamPETR-QAF2D with the state-of-the-art approaches on the nuScenes test split. The results in Table 2 show that while StreamPETR-8DQuery has the same performance as StreamPETR in terms of NDS and mAP, our OAF2D still can improve StreamPETR by 0.6% NDS and 1.6% mAP, which further validates the effectiveness of our approach. Meanwhile, QAF2D-enhanced StreamPETR achieves the best performance on the nuScenes test split.

The performance improvement can be attributed to two aspects: (1) 3D query anchors inferred from 2D detection results can provide better initial 3D box positions and sizes than random anchors, and this can ease the optimization of the network and predict the results more accurately, (2) the state-of-the-art 2D detectors are more reliable than the 3D detectors, and some missed detections of the base 3D detectors can be recovered with the help of QAF2D.

Component time consumption and speed. We report the time consumption of each component and the speed of StreamPETR [33] and StreamPETR-QAF2D in Table 3. Note that the two approaches use the same backbone, the slight difference in time between two separate runs should be inevitable for the hardware. Though StreamPETR-QAF2D is somewhat slower than StreamPETR (15.4 FPS vs 18.9 FPS), we think that the overall efficiency of StreamPETR-QAF2D is acceptable. Meanwhile, the increase of the complexity mainly comes from the 2D detection head of DAB-DETR [21] and the 3D query anchor generation component is very fast (1ms). As QAF2D is not sensitive to the choice of the 2D detector (please refer to the ablation study in Section 4.4), the efficiency of QAF2D can be improved by using lighter 2D detectors.

Method	NDS	mAP
BEVFormer-small-DAB3D	49.2	39.0
BEVFormer-small-QAF2D (Faster-RCNN [31])	50.0	39.5
BEVFormer-small-QAF2D (DAB-DETR [21])	**50.2**	**39.7**

Table 4. Comparison of QAF2D with different 2D detectors.

Method	NDS	mAP
BEVFormer-small-QAF2D	50.2	39.7
BEVFormer-small-QAF2D (w/ RA)	**50.3**	**40.1**

Table 5. Impact of additional random anchors (RA).

	NDS	mAP	# of prompt params.
No sharing	50.3	40.2	–
Sharing w/o prompt	49.7	39.2	–
$\tau = 0.1$	50.0	39.5	0.08M
$\tau = 0.2$	**50.2**	**39.7**	0.15M
$\tau = 0.3$	50.1	**39.7**	0.20M
$\tau = 0.4$	49.8	39.4	0.23M
$\tau = 0.5$	**50.2**	39.6	0.26M

Table 6. Effect of visual prompts in feature sharing and comparison of different τs.

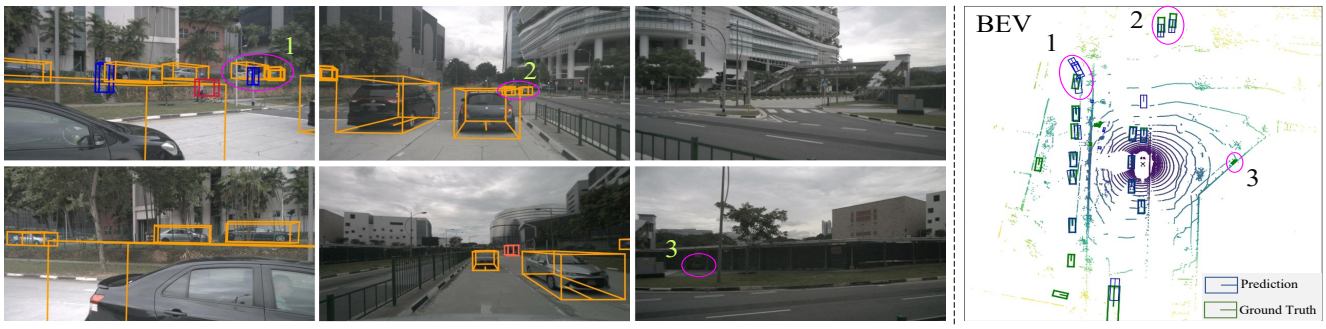
4.4. Ablation Study

We carry out ablation study on the nuScenes validation split and use BEVFormer [18] as the base detector.

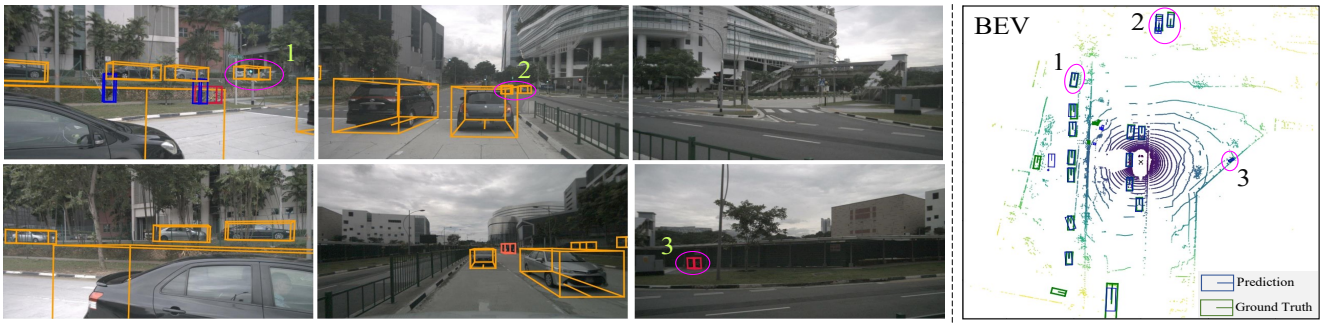
Generalization to different 2D detectors. To study the generalization ability of QAF2D to different 2D detectors, besides the default 2D detector DAB-DETR [21] , we also combine QAF2D with the popular Faster-RCNN [31] and applies it to BEVFormer [18]. Based on the results in Table 4, we can see that the performances of DAB-DETR and Faster-RCNN are very close to each other, which can demonstrate that the proposed QAF2D is not sensitive to the selection of the 2D detector. As the result of DAB-DETR is slightly better, we use DAB-DETR as the default 2D detector to provide 2D detection boxes. The insensitivity should be because that QAF2D does not need very precise 2D boxes, as long as the 2D detector does not miss objects, QAF2D can use rough 2D boxes to generate meaningful 3D query anchors that are better than random ones.

Impact of additional random 3D query anchors. We also combine an additional set of randomly initialized 3D query anchors with anchors generated by QAF2D to explore its impact. The number of random 3D query anchors is 900, which is the number of queries used by BEVFormer-small. The results are shown in Table 5. The additional random anchors can bring about improvement. But as the improvement of 0.1% NDS and 0.4% mAP is small, we prefer not to use additional random anchors in our default setting.

Effect of visual prompts for feature sharing. To investigate the effects of adding a small number of prompt parameters to the feature maps of the backbone for feature



(a) Results of StreamPETR-8DQuery



(b) Results of StreamPETR-QAF2D

Figure 3. Visualization results of StreamPETR-8DQuery and StreamPETR-QAF2D. The results in multi-camera images are shown on the left, and the corresponding results in bird's-eye-view are shown on the right. Three typical cases where StreamPETR-8DQuery fails but its QAF2D-enhanced version succeeds are in purple ellipses with numbers.

sharing, we train a separate 2D detector that has its own image feature extraction backbone (denoted by "No sharing"). We also train another 2D detector that directly uses the image features of the backbone trained by the 3D detector, and only the 2D detection head is fine-tuned (denoted by "Sharing w/o prompt"). From the results in Table 6, we can see that "No sharing" and "Sharing w/o prompt" serve as the upper bound and lower bound of performance, respectively.

The parameter τ in "Stage 2" of Section 3.3 controls the number of prompt parameters. We vary τ from 0.1 to 0.5 to study how the choice of it affects the performance. The results in Table 6 show that all choices of τ can close the gap between "No sharing" and "Sharing w/o prompt", and $\tau = 0.2$ is the best value. With $\tau = 0.2$, the NDS difference between "No sharing" and "Sharing w/o prompt" is reduced from 0.6% to 0.1%, and the mAP difference is reduced from 1.0% to 0.5%.

4.5. Visualization Results

Visual comparison between StreamPETR-8DQuery and StreamPETR-QAF2D are shown in Fig. 3. We draw 3D detection boxes in multi-camera images and their projections in BEV space. Three typical cases where the proposed QAF2D helps are given. Case 1 shows that QAF2D can remove the false positive (blue box in the purple ellipse in the top-left image of Fig. 3 (a)) and make the true positive more accurate (see the alignment between the prediction and ground truth in BEV). Case 2 demonstrates that when the objects are faraway, the prediction of QAF2D is more accurate as well. Case 3 shows that when the object is small and difficult to distinguish from the background, QAF2D can help to alleviate miss detection. Please refer to the appendix for more visualization results.

5. Conclusion and Limitation

In this paper, we propose to generate 3D query anchors from 2D boxes so that the more reliable 2D detection results can be used to boost the performance of 3D detectors. To share the image feature backbone between 2D and 3D detectors while keeping the performance of the 3D detector uncompromised, we design a two-stage optimization approach with visual prompts. We integrate the proposed approach into three query-based 3D object detectors, and comprehensive experiments are carried out on the nuScenes dataset to verify its effectiveness.

A limitation of our approach is that 3D detection results depend on the quality of 2D detectors (though not sensitive to it). If the 2D detector misses an object, it should be difficult for a query-based 3D detector to recover the missed object. Meanwhile, combining the 3D anchors generated by our approach with the random anchors in a straightforward manner does not produce notable improvement. We will investigate how to achieve synergy between the two kinds of anchors in our future work.

Acknowledgement. P. Liang was supported in part by a Fundamental Research Cultivation Fund of ZZU.

References

[1] Hyojin Bahng, Jahanian Ali, Sankaranarayanan Swami, and Isola Phillip. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022. 2, 3, 4, 5

[2] 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 11618–11628, 2020. 2, 5

[3] Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li. Monopair: Monocular 3d object detection using pairwise spatial relationships. In *CVPR*, pages 12093–12102, 2020. 2

[4] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *CVPR*, pages 21674–21683, 2023. 1

[5] Bowen Dong, Pan Zhou, Shuicheng Yan, and Wangmeng Zuo. Lpt: Long-tailed prompt tuning for image classification. In *ICLR*, 2023. 3

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 3

[7] 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*, pages 8458–8468, 2022. 1

[8] Cheng Han, Qifan Wang, Yiming Cui, Zhiwen Cao, Wenguan Wang, Siyuan Qi, and Dongfang Liu. E2vpt- an effective and efficient approach for visual prompt tuning. In *ICCV*, pages 17491–17502, 2023. 3

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3, 7

[10] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, pages 709—727, 2022. 2, 3

[11] Xiaohui Jiang, Shuailin Li, Yingfei Liu, Shihao Wang, Fan Jia, Tiancai Wang, Lijin Han, and Xiangyu Zhang. Far3d: Expanding the horizon for surround-view 3d object detection. *arXiv preprint arXiv:2308.09616*, 2023. 2

[12] Yanqin Jiang, Li Zhang, Zhenwei Miao, Xiatian Zhu, Jin Gao, Weiming Hu, and Yu-Gang Jiang. Polarformer: Multi-camera 3d object detection with polar transformers. In *AAAI*, pages 1042–1050, 2023. 1, 2, 6

[13] Youngwan Lee, Joong-won Hwang, Sangrok Lee, Yuseok Bae, and Jongyoul Park. An energy and gpu-computation efficient backbone network for real-time object detection. In *CVPRW*, pages 752–760, 2019. 3, 6

[14] 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. 1, 3

[15] Hongyang Li, Hao Zhang, Zhaoyang Zeng, Shilong Liu, Feng Li, Tianhe Ren, and Lei Zhang. Dfa3d: 3d deformable attention for 2d-to-3d feature lifting. In *ICCV*, pages 6684–6693, 2023. 2

[16] Yinhao Li, Han Bao, Zheng Ge, Jinrong Yang, Jianjian Sun, and Zeming Li. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with temporal stereo. In *AAAI*, pages 1486–1494, 2023. 2

[17] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In *AAAI*, pages 1477–1485, 2023. 2

[18] 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. In *ECCV*, pages 1–18, 2022. 1, 2, 5, 6, 7

[19] Haisong Liu, Yao Teng, Tao Lu, Haiguang Wang, and Limin Wang. Sparsebev: High-performance sparse 3d object detection from multi-camera videos. In *ICCV*, pages 18580–18590, 2023. 1, 2, 3, 5, 6

[20] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, pages 1–35, 2023. 3

[21] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. In *ICLR*, 2022. 1, 3, 4, 5, 6, 7

[22] Shilong Liu, Tianhe Ren, Jiayu Chen, Zhaoyang Zeng, Hao Zhang, Feng Li, Hongyang Li, Jun Huang, Hang Su, Jun Zhu, and Zhang Lei. Detection transformer with stable matching. In *ICCV*, pages 6491–6500, 2023. 3

[23] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *ECCV*, pages 531–548, 2022. 1, 3, 5

[24] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Qi Gao, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petrv2: A unified framework for 3d perception from multi-camera images. *arXiv preprint arXiv:2206.01256*, 2022. 1, 3, 6

[25] Xinzhu Ma, Yinmin Zhang, Dan Xu, Dongzhan Zhou, Shuai Yi, Haojie Li, and Wanli Ouyang. Delving into localization errors for monocular 3d object detection. In *CVPR*, pages 4721–4730, 2021. 2

[26] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *ICCV*, pages 3651–3660, 2021. 1, 3

[27] Carion Nicolas, Massa Francisco, Synnaeve Gabriel, Usunier Nicolas, Kirillov Alexander, and Zagoruyko Sergey. End-to-end object detection with transformers. In *ECCV*, page 213–229, 2020. 1, 2, 3, 5

[28] Adam Paszke, S. Gross, Soumith Chintala, G. Chanan, E. Yang, Zachary Devito, Zeming Lin, Alban Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *NeurIPS*, 2017. 5

[29] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, pages 194–210, 2020. 1, 2

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021. 3

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE TPAMI*, pages 1137–1149, 2017. 7

[32] Pei Sun, Mingxing Tan, Weiyue Wang, Chenxi Liu, Fei Xia, Zhaoqi Leng, and Dragomir Anguelov. Swformer: Sparse window transformer for 3d object detection in point clouds. In *ECCV*, pages 426–442, 2022. 1

[33] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In *ICCV*, pages 3621–3631, 2023. 1, 2, 3, 5, 6, 7

[34] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. FCOS3D: Fully convolutional one-stage monocular 3d object detection. In *ICCVW*, pages 913–922, 2021. 2

[35] Yue Wang, Vitor Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *COLR*, pages 180–191, 2021. 1, 3, 6

[36] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *AAAI*, pages 2567–2575, 2022. 3

[37] Zitian Wang, Zehao Huang, Jiahui Fu, Naiyan Wang, and Si Liu. Object as query: Equipping any 2d object detector with 3d detection ability. In *ICCV*, pages 3791–3800, 2023. 2, 3, 6

[38] Kaixin Xiong, Shi Gong, Xiaoqing Ye, Xiao Tan, Ji Wan, Errui Ding, Jingdong Wang, and Xiang Bai. Cape: Camera view position embedding for multi-view 3d object detection. In *CVPR*, pages 21570–21579, 2023. 3, 6

[39] Renrui Zhang, Han Qiu, Tai Wang, Ziyu Guo, Ziteng Cui, Yu Qiao, Hongsheng Li, and Peng Gao. Monodetr: Depth-guided transformer for monocular 3d object detection. In *CVPR*, pages 9155–9166, 2023. 2

[40] Yunpeng Zhang, Wenzhao Zheng, and Zheng Zhu. A simple baseline for multi-camera 3d object detection. In *AAAI*, page 3507–3515, 2023. 1, 2

[41] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2020. 1, 3, 5, 6