Viewpoint Equivariance for Multi-View 3D Object Detection

Dian Chen   Jie Li   Vitor Guizilini   Rareș Ambruș   Adrien Gaidon

Toyota Research Institute (TRI), Los Altos, CA

{firstname.lastname}@tri.global

Abstract

3D object detection from visual sensors is a cornerstone capability of robotic systems. State-of-the-art methods focus on reasoning and decoding object bounding boxes from multi-view camera input. In this work we gain intuition from the integral role of multi-view consistency in 3D scene understanding and geometric learning. To this end, we introduce VEDet, a novel 3D object detection framework that exploits 3D multi-view geometry to improve localization through viewpoint awareness and equivariance. VEDet leverages a query-based transformer architecture and encodes the 3D scene by augmenting image features with positional encodings from their 3D perspective geometry. We design view-conditioned queries at the output level, which enables the generation of multiple virtual frames during training to learn viewpoint equivariance by enforcing multi-view consistency. The multi-view geometry injected at the input level as positional encodings and regularized at the loss level provides rich geometric cues for 3D object detection, leading to state-of-the-art performance on the nuScenes benchmark. The code and model are made available at https://github.com/TRI-ML/VEDet.

1. Introduction

Camera-based 3D object detection is a critical research topic, with important applications in areas such as autonomous driving and robotics due to the semantic-rich input and low cost compared to range sensors. In the past few years, monocular 3D detection has seen significant progress, from relying on predicting pseudo point clouds as intermediate representation [33,39,42] to end-to-end learning [31,34,38]. However, monocular 3D detectors are inherently ambiguous in terms of depth, which motivated some recent exploration in multi-view and multi-sweep 3D object detection [22,25,26,40].

In a conventional monocular setting, given multiple cameras on a sensor rig, single-view detections are merged to the global frame through rule-based processing such as Non-Maximum Suppression (NMS). Recent advances in
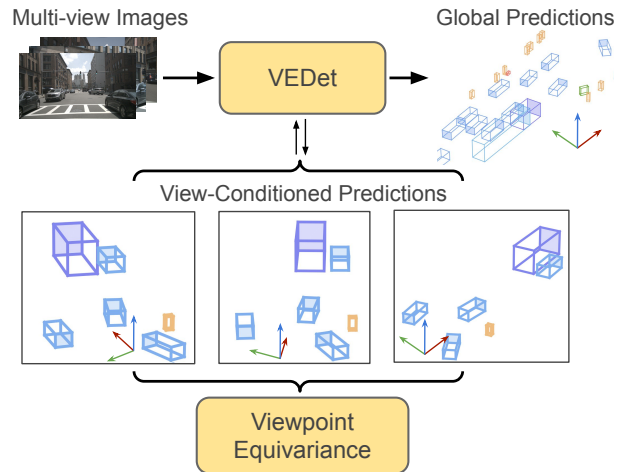

Figure 1. **Our proposed VEDet** encodes the 3D scene from multi-view images, and decodes objects with view-conditioned queries. The predicted 3D bounding boxes are expressed in the underlying views of the queries, which enables us to enforce viewpoint equivariance among predictions from multiple views. Virtual query views are generated during training and together with the viewpoint equivariance regularization bring richer geometric learning signals to guide the model to better understand the 3D structure in the scene. During inference, the global predictions can be obtained by simply choosing the global frame as the query view.

multi-view camera-based 3D algorithms [25,40] proposed to jointly aggregate multi-view information at the feature level, and directly predict a single set of detections in the global frame. These algorithms demonstrate a giant leap in 3D detection performance on multi-camera datasets (E.g., Nuscenes [5]). To aggregate information from different views, one line of query-based detectors adopt transformers to query image features [25,26,40] or bird's-eye-view (BEV) features [15,22] via an attention mechanism. In contrast, another line of works "lift-splat-shoot" [32] image features from each view into the shared BEV features to be processed by convolutional detection heads [21]. To further mitigate the depth ambiguity, some concurrent works have started extending multi-view to "multi-sweep" across timestamps and observe a promising performance boost [22,26].

While the works mentioned above demonstrate a strong potential for multi-view 3D detection, progress has concentrated on input aggregation and information interplay across frames and less on learning objectives. We argue that the learning objective can play a crucial role in ingesting the core knowledge in a multi-view setting: 3D geometry.

This paper proposes to encourage 3D geometry learning for multi-view 3D detection models through viewpoint awareness and equivariance. We obtain our intuition from traditional structure-from-motion works [1], where multi-view geometry is modeled through multi-view consistency. To this end, we propose viewpoint-awareness on the object queries, as well as a multi-view consistency learning objective as a 3D regularizer that enforces the model to reason about geometry. Compared to existing methods that make 3D predictions in the default egocentric view, our proposed multi-view predictions and viewpoint equivariance effectively bring stronger geometric signals conducive to the 3D reasoning. More specifically, in our query-based framework, the geometry information of image features and object queries is injected completely via implicit geometric encodings, and the transformer decoder is expected to learn better correspondence and 3D localization under the viewpoint equivariance objective. We demonstrate that our proposed framework can make the best of available geometry information with extensive experiments and establish the new state-of-the-art in multi-view 3D object detection. In summary, our contributions are:

- We propose a novel **Viewpoint Equivariance (VE)** learning objective that encourages multi-view consistency in 3D detection models, leading to improved 3D object detection performance.

- We propose a new multi-view 3D object detection framework, **VEDet**, which employs a query-based transformer architecture with perspective geometry and viewpoint awareness injected both at the encoding and decoding stages. VEDet fully enables our proposed VE learning objective, facilitating geometry learning with implicit inductive biases.

- VEDet achieves **state-of-the-art on large-scale benchmark**, reaching **45.1%mAP on NuScenes val set and 50.5% mAP on test set**. We provide a comprehensive analysis of our components, and share insights based on empirical observations.

2. Related Work

2.1. Monocular 3D object detection

Early works tackled camera-based 3D object detection in a monocular setting by adopting a two-stage pseudo-LiDAR paradigm [33, 39, 42, 47] or directly building upon 2D detection frameworks to predict extra 3D properties [4, 8, 30, 31, 34, 38, 44]. Due to the inherent scale ambiguity in depth estimation, one standard approach was to lift from 2D to 3D by aligning 3D properties and their 2D projections on the image plane [3, 12, 16, 19], while others leveraged additional object or scene priors such as shapes [12], CAD models [7], or ground planes [2]. In the line of representation learning, DD3D [31] exploited large-scale pre-training to learn a depth-aware representation that can universally benefit 3D detection algorithms. However, all these methods still struggle with the two major drawbacks in monocular 3D detection: inherent depth ambiguity and insufficient context to infer objects across images. As a multi-view method, our work addresses both of these issues by leveraging the ample 3D geometric cues in multi-camera setups.

2.2. Multi-view 3D object detection

Recent advances in camera-based 3D object detection have started to leverage multi-view context, which can improve the detection of objects that appear in more than one image. One line of works extends the DETR framework [6], which decodes 3D bounding boxes with a set of queries [15, 20, 22, 25, 26, 40]. Using camera parameters, DETR3D [40] directly projects 3D queries to 2D image planes to update query features, while PETR [25] constructs 3D position embeddings from point frustums to implicitly guide query updates. BEVFormer [22] and UVTR [20] first build an intermediate voxelized feature space around the vehicle/robot's ego coordinate frame, before feeding the features to a DETR-style decoder. Another line of work follows LSS [32] and constructs the voxelized feature space, before applying a detection head on the features to predict bounding boxes. Typically, a depth head is also trained to predict a depth bin distribution in order to lift-splat-shoot the features, as in BEVDepth [21]. Our work falls in the first line of research and exploits multi-view geometric consistency to improve bounding box localization in 3D space.

2.3. Implicit geometric encoding

The Transformer architecture [9, 10, 37] introduced the use of positional encodings for input features. This brought upon a paradigm shift in how to model the relative position of elements, from *explicitly*, i.e. by recurrent operations or convolutional filters, to *implicitly*, i.e. learned from data. Inspired by this new paradigm, some works started to investigate how to use positional encodings constructed from geometric priors as input-level inductive biases [25, 26, 45]. ILIB [45] utilizes multi-view geometry, including camera and epipolar cues, to generate position encodings, to be processed by a generalist Perceiver IO [14] architecture to produce a set of latent vectors. From this latent space, ILIB constructs queries from 3D viewing rays to decode depth maps. PETR [25, 26] similarly constructs

position encodings from generated point frustums at predefined depth values, and decodes 3D bounding boxes using queries constructed from 3D anchor points [41] in the ego vehicle space. Positional encoding has also been used extensively in the context of neural fields (i.e. coordinate-based multi-layer perceptron) to process the appearance, radiance, or occupancy of a scene [29, 35, 43]. Our work improves the design of 3D geometric position encoding and introduces a new optimization objective to guide the detection model toward learning better object localization.

3. Viewpoint Equivariant 3D Detection

The multi-view 3D object detection task aims at detecting 3D bounding boxes in the scene with class labels, given a set of images from N cameras with poses and intrinsics $\{\mathbf{I}_i \in \mathbb{R}^{3 \times H \times W}, \mathbf{T}_i \in SE(3), \mathbf{K}_i \in \mathbb{R}^{3 \times 3}, i = 1, 2, \ldots, N\}$. In this section, we will first introduce the overall VEDet framework in Sec. 3.1. Sec. 3.2 describes how we use geometric positional encoding to inject geometry information for image features and object queries implicitly. In Sec. 3.3 we propose making object queries view-conditioned so that 3D boxes are predicted in the specified view. Lastly, we present the novel viewpoint equivariance learning objective in Sec. 3.4, which exploits the viewpoint-awareness of object queries and produces stronger geometric signals to improve 3D detection.

3.1. Overall framework

The workflow of our proposed VEDet builds upon a transformer-based architecture, as depicted in Fig. 2. We first employ a backbone network that extracts image features $\{\mathbf{F}_i \in \mathbb{R}^{C \times H' \times W'}\}$ from multi-view images. For each 2D location on the feature map grid, we calculate a geometric positional encoding that jointly considers pixel location, camera pose, and intrinsics. The image features, with their associated positional encoding, are flattened and processed by a transformer decoder [6] with a set of object queries $\{\mathbf{q}_j\}$. The queries are constructed from a set of learnable 3D *query points* $\{\mathbf{c}_j\}$ combined with a given *query view* $\{\mathbf{T}^v\}$. A series of self- and cross-attention layers then aggregate and update the 3D scene information into the queries, after which a feed-forward detection head maps the updated queries to box predictions $\{\hat{\mathbf{b}}_j\}$. The box predictions are conditioned and expressed in the query views $\mathbf{T}^v_j$ associated with the queries, as detailed in Sec. 3.3. Finally, we optimize the network by applying a viewpoint equivariance (VE) loss on the view-conditioned box predictions, as detailed in Sec 3.4.

3.2. Geometric positional encoding

Positional encodings provide location information of feature embeddings in Transformer architectures [37]. In this work, we encode the 3D geometric attributes associated

with the image features as well as the object queries, when they are processed by the decoder. Inspired by [25, 45], for the image features we propose to encode the camera pose and the 3D inverse projection ray that combines pixel position and camera perspective geometry; for object queries the learnable 3D query point and the selected query view are encoded (more in Sec. 3.3).

Specifically, given the extracted image features $\{\mathbf{F}_i\}$, we construct a triplet of geometric attributes, $[\mathbf{r}_{(u_i, v_i)}, \bar{\mathbf{q}}_i, \mathbf{t}_i]$ for *each feature location* (u_i, v_i). $[\bar{\mathbf{q}}_i, \mathbf{t}_i]$ denote the quaternion vector and translation of the camera pose, and $\mathbf{r}_{(u_i, v_i)}$ denotes a unit-length inverse perspective projection ray originating from the pixel location given by:

$$\mathbf{r}'_{(u_i, v_i)} = (\mathbf{K}_i \mathbf{R}_i^T)^{-1}[\alpha u_i, \alpha v_i, 1]^T, \mathbf{r} = \frac{\mathbf{r}'}{||\mathbf{r}'||_2}, \quad (1)$$

where α is the downsample factor of $\mathbf{F}_i$ compared to image $\mathbf{I}_i$, $\mathbf{K}_i$ and $\mathbf{R}_i$ are instrinsic and rotation matrix of camera i. The triplet $[\mathbf{r}_{(u_i, v_i)}, \bar{\mathbf{q}}_i, \mathbf{t}_i]$ fully describes the perspective geometry for a given image feature $\mathbf{F}_i(u_i, v_i)$. Compared to PETR [25, 26] which model the positional information of image features by manually sampling a set of 3D point locations along the ray at pre-defined depth frustums, VEDet employs a simpler design[1] and chooses not to assume the discretized depth prior, as we believe $[\mathbf{r}_{(u_i, v_i)}, \bar{\mathbf{q}}_i, \mathbf{t}_i]$ keeps the full geometry information with which the model can learn 3D localization better.

Learnable geometry mapping We encode the geometric attributes into high-dimensional embeddings via Fourier transform followed by a learnable mapping. Inspired by advances in NeRF [29, 36], we first apply a Fourier transform to capture the fine-grained changes in the geometric attributes.

$$\gamma(x|[f_1, \ldots, f_k]) = [\sin(f_1 \pi x), \cos(f_1 \pi x), \ldots] \quad (2)$$

The k frequencies $[f_1, \ldots, f_k]$ are sampled evenly between $[0, f_{\max}]$. Afterward, an MLP is used to project the output to dimension C as our final geometric positional encoding:

$$\mathbf{p}^e_{(u_i, v_i)} = \mathrm{MLP}_{\mathrm{enc}}(\gamma([\mathbf{r}_{(u_i, v_i)}, \bar{\mathbf{q}}_i, \mathbf{t}_i]) \quad (3)$$

As a result, even without explicitly projecting the image features $\{\mathbf{F}_i\}$ back to 3D space, they become 3D geometry-aware when augmented with the 3D geometric positional encodings $\{\mathbf{P}^e_i \in \mathbb{R}^{C \times H' \times W'}\}$. Hence, we implicitly encode the multi-view perception of the scene at an input level, which will work jointly with our proposed VE learning objective to enforce 3D geometric modeling.

Temporal modeling In the context of a multi-sweep setting, we follow [26] and transform the camera pose from

---

[1] PETR also combines a few other components with the 3D PE, namely 2D grid PE and view number PE, which we do not use.
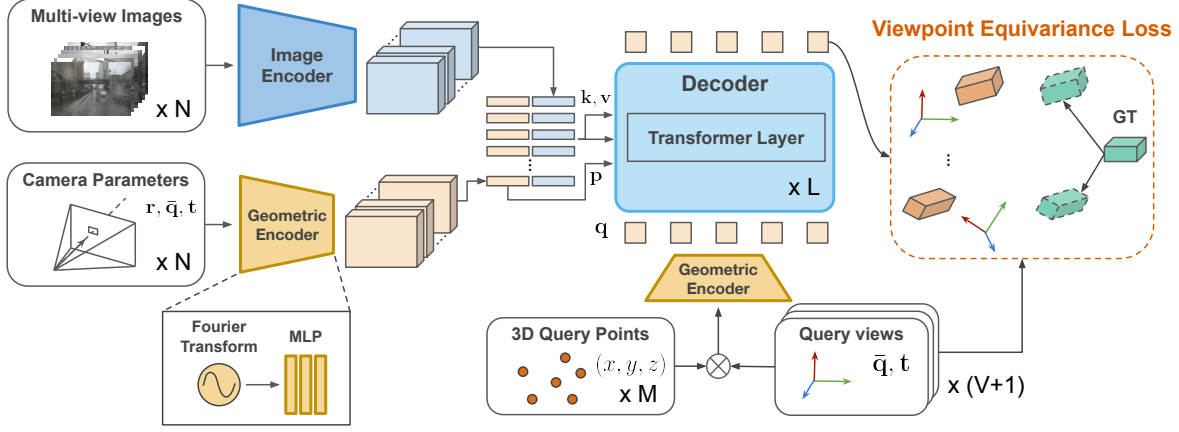
Figure 2. The framework of our proposed VEDet: Given N multi-view input cameras, an image encoder first extracts image features. For each feature embedding, we provide geometric positional encoding (PE) based on pixel location as well as camera geometrics (Sec. 3.2). At the decoding stage, we apply a view-conditioned query constructed by 3D query points and Query Views to predict view-conditioned predictions (Sec. 3.3). Finally, we optimize the network through a novel viewpoint equivariance loss (Sec. 3.4).

previous frames into the current global frame via ego-motion compensation. The multi-sweep features are concatenated at the token dimension.

3.3. View-conditioned query

VEDet adopts a DETR-style [6] decoder that consists of L transformer layers, as shown in Fig. 2. Each layer performs self-attention among a set of M queries $\{\mathbf{q}_j \in \mathbb{R}^C, j = 1, 2, \ldots, M\}$, and cross-attention between the queries and the 3D geometry-aware image features $\{(\mathbf{F}_i, \mathbf{P}_i^e)\}$. The updated queries $\{\mathbf{q}_j\}$ will serve as input to the next layer:

$$\{\mathbf{q}_j\}_l = \psi_{l-1}(\{\mathbf{F}_i\}, \{\mathbf{P}_i^e\}, \{\mathbf{q}_j\}_{l-1}), l = 1, 2, \ldots, L, \quad (4)$$

where L is the number of attention layers. A classification and regression MLP heads map the queries from each layer into class logits and bounding box predictions, respectively.

$$\hat{\mathbf{s}}_j = \mathrm{MLP}_{\mathrm{cls}}(\mathbf{q}_j), \hat{\mathbf{b}}_j = \mathrm{MLP}_{\mathrm{reg}}(\mathbf{q}_j) \quad (5)$$

We propose to ground the queries with multi-view geometry. Concretely, a query $\mathbf{q}_j$ is constructed from two parts: a *3D query point* and a *query view*. We initialize a set of M learnable 3D query points $\{\mathbf{c}_j \in \mathbb{R}^3, j = 1, 2, \ldots, M\}$ in the *global frame* $\mathbf{T}^0$, similarly to PETR [25], which is optimized during training.

Query views For the second part of query geometry, a query view $\mathbf{T}^v = [\bar{\mathbf{q}}^v, \mathbf{t}^v]$ is selected relative to the global frame. To construct the query, the 3D query points are first transformed into the query view via $\mathbf{c}_j^v = (\mathbf{T}^v)^{-1}\mathbf{c}_j$, and together with the query view $[\mathbf{c}_j^v, \bar{\mathbf{q}}^v, \mathbf{t}^v]$ compose the query geometries. As described in Sec. 3.2, the query geometries

are similarly mapped by a Fourier transform followed by an MLP, into **view-conditioned queries**:

$$\mathbf{q}_j^v = \mathrm{MLP}_{\mathrm{dec}}(\gamma([\mathbf{c}_j^v, \bar{\mathbf{q}}^v, \mathbf{t}^v])). \quad (6)$$

For query views, we refer the *global frame* $\mathbf{T}^0 = [[1, 0, 0, 0], \mathbf{0}]$ as a default query view.[2] Additionally, we generate V *virtual query views* to provide variation to the decoding views and encourage viewpoint awareness in the model. Concretely, we randomly sample Euler angles $\Theta^v \in \mathbb{R}^3$ and translation $\mathbf{t}^v \in \mathbb{R}^3$ from uniform distributions $[\Theta_{\min}, \Theta_{\max}]$ and $[\mathbf{t}_{\min}, \mathbf{t}_{\max}]$, after which the Euler angles will be converted to the equivalent quaternion $\bar{\mathbf{q}}^v \in SO(3)$, giving $\{\mathbf{T}^v = [\bar{\mathbf{q}}^v, \mathbf{t}^v]\}$. In total, there are $V + 1$ query views consisting of the default view and V virtual views $\{\mathbf{T}^v, v = 0, 1, \ldots, V\}$. Therefore, given the M 3D query points $\{\mathbf{c}_j\}$ and $V + 1$ query views $\{\mathbf{T}^v\}$, we construct object queries from $\{\mathbf{c}_j\} \times \{\mathbf{T}^v\}$, resulting in total $M \times (V+1)$ individual object queries.

View-conditioned predictions The query view specifies the coordinate system in which boxes (groundtruth, predicted) are defined. Specifically, given a *view-conditioned* query $\mathbf{q}_j^v$, the box predictions $\hat{\mathbf{b}}_j^v$ are local to the underlying query view $\mathbf{T}^v$, parameterized as:

$$\hat{\mathbf{b}}_j^v = [\Delta\hat{\mathbf{c}}_j^v, \hat{\mathbf{d}}_j, \cos(\phi), \sin(\phi), \hat{\mathbf{v}}_j^v], \quad (7)$$

where $\Delta\hat{\mathbf{c}}_j^v \in \mathbb{R}^3$ is the offset from the 3D query point $\mathbf{c}_j^v$ to the bounding box center, $\hat{\mathbf{d}}_j \in \mathbb{R}^3$ is the box dimensions, ϕ is the yaw angle of the box, and $\hat{\mathbf{v}}_j^v \in \mathbb{R}^3$ is the box velocity. As for the object classification score, we simply decode one

---

[2]This view is also the egocentric coordinate frame for box prediction and evaluation as done in other multi-view detection works [25, 26, 40].
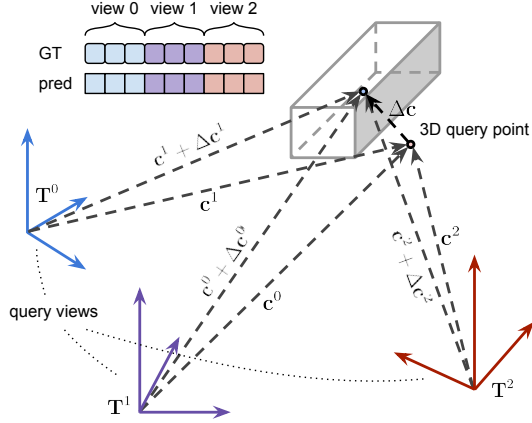
Figure 3. **Multi-view consistency enforced at training time.** According to multi-view geometry, the observations of the same box in different views should only differ by the relative transformation. Therefore, when a 3D query point is paired with multiple query views (3 in this illustration) to construct queries, the predictions in each respective view are *combined* to match and regress the ground truth counterparts *jointly*.

from the global frame for each query point $\mathbf{c}_j$, as it is simple and decoding from virtual views did not show advantage in our experiments. We predict a binary score for each class normalized by a sigmoid function.

The view-conditioned queries and their local predictions serve as a form of data augmentation during training and, more importantly, enable viewpoint equivariance regularization as discussed in Sec. 3.4. More design choices are also ablated in Sec. 4.3. We only use the global frame $\mathbf{T}^0$ as the query view at inference time.

3.4. Viewpoint equivariance loss

As described in Sec. 3.3, given $V + 1$ query views, there are $V + 1$ versions of bounding box predictions $\{\hat{\mathbf{b}}_j^v\}$ coming from a single query point $\mathbf{c}_j$. The $V + 1$ bounding boxes are expressed in different coordinate frames but of the same underlying ground truth object. According to multi-view geometry, the observations of the *same* object from different frames should be geometrically consistent and only differ by the relative transformation as shown in Fig. 3. Therefore, we propose a viewpoint equivariance objective that considers the multi-view predictions coming from the same query point c_j and box target from all query views *jointly*.

Concretely, we first ensure that the $V + 1$ versions predictions from query point $\mathbf{c}_j$ are assigned to the same ground truth object. To achieve this goal, we create a **super box** by concatenating the predictions from different query views:

$$\hat{\mathbf{B}}_j = [\hat{\mathbf{b}}_j^0, \hat{\mathbf{b}}_j^1, \ldots, \hat{\mathbf{b}}_j^V]. \tag{8}$$

Similarly, we extend all the ground truth bounding boxes

into super boxes:

$$\mathbf{B}_m = [\mathbf{g}_m^0, \mathbf{g}_m^1, \ldots, \mathbf{g}_m^V], \tag{9}$$

where $\mathbf{g}_m^v$ is the ground truth bounding box in the expressed in the query view $\{\mathbf{T}^v\}$.

Next, we perform Hungarian matching [17] to decide the optimal assignment between $\{\hat{\mathbf{B}}_j\}$ and $\{\mathbf{B}_m\}$, using the following cost function similar to DETR [6]:

$$\sigma = -\mathbb{1}_{\{c_m \neq \emptyset\}} \log(\mathbf{s}_j(c_m)) + \mathbb{1}_{\{c_m \neq \emptyset\}} L_{reg}(\mathbf{B}_m, \hat{\mathbf{B}}_j), \tag{10}$$

where c_m is the ground truth class label and $L_{reg}()$ is a weighted L1 loss, given by:

$$L_{reg}(\mathbf{B}_m, \hat{\mathbf{B}}_j) = ||\hat{\mathbf{b}}_j^0 - \mathbf{g}_m^0||_1 + \Sigma_1^V \lambda_v ||\hat{\mathbf{b}}_j^v - \mathbf{g}_m^v||_1. \tag{11}$$

We use λ_v to weigh the virtual views. Once we identify the optimal assignment, we calculate the loss on the super boxes:

$$L_{VE} = \lambda_{cls} L_{cls}(\mathbf{s}, c) + \lambda_{reg} L_{reg}(\mathbf{B}, \hat{\mathbf{B}}), \tag{12}$$

for each paired prediction and ground truth. We adopt focal loss [24] for classification loss L_{cls}, and the same form of regression loss L_{reg} as in matching. λ_{cls} and λ_{reg} are loss weights. For each 3D query point, by considering $V + 1$ versions of predictions *jointly* during both matching and optimization, the model learns viewpoint equivariance through multi-view consistency, leading to better 3D detection.

4. Experiments

4.1. Experimental setup

Dataset and metrics We evaluate our method on the large-scale benchmark NuScenes. NuScenes has 1000 scenes split into 700/150/150 as train/val/test subsets. Each scene contains 20-second videos collected by 6 surround-view cameras at 10Hz, with synchronized 3D box annotations at 2Hz. We report on metrics defined by NuScenes: mean Average Precision (mAP), mean Average Translation Error (mATE), mean Average Scale Error (mASE), mean Average Orientation Error, mean Average Velocity Error (mAVE), mean Average Attribution Error (mAAE), and NuScenes Detection Score (NDS) which is a comprehensive score that aggregates the above five sub-metrics.

Implementation details We adopt ResNet [11] and VoVNetV2 [18] with FPN [23] as the backbone network, and use the P4 features (1/16 image size) from the pyramid for all our experiments. We use the AdamW optimizer [28] with cosine annealing [27] to train VEDet with learning rate starting at 2×10^{-4} and ending at 2×10^{-7}, on 8 Tesla A100 GPUs with a total batch size 8. For image

Method	Backbone	Image size	CBGS	mAP↑	NDS↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
PETR	Res-50	384×1056	✓	0.313	0.381	0.768	**0.278**	0.564	0.923	0.225
VEDet	Res-50	384×1056	✓	**0.347**	**0.443**	**0.726**	0.282	**0.542**	**0.555**	**0.198**
FCOS3D	Res-101-DCN	900×1600		0.295	0.372	0.806	0.268	0.511	1.131	**0.170**
DETR3D†	Res-101-DCN	900×1600	✓	0.349	0.434	0.716	0.268	0.379	0.842	0.200
BEVFormer†	Res-101-DCN	900×1600		0.416	0.517	0.673	0.274	0.372	**0.394**	0.198
UVTR†	Res-101-DCN	900×1600		0.379	0.483	0.731	**0.267**	**0.350**	0.510	0.200
PETR	Res-101	512×1408	✓	0.357	0.421	0.710	0.270	0.490	0.885	0.224
VEDet	Res-101	512×1408	✓	**0.432**	**0.520**	**0.638**	0.275	0.362	0.498	0.191
PETR‡	V2-99	640×1600		0.404	0.447	0.739	0.271	0.452	0.876	0.208
PETRv2‡	V2-99	640×1600		0.431	0.517	0.730	0.264	0.399	**0.404**	**0.190**
VEDet-SF‡	V2-99	640×1600		0.433	0.486	0.683	**0.263**	0.352	0.808	0.201
VEDet ‡	V2-99	640×1600		**0.451**	**0.527**	**0.670**	0.263	0.347	0.510	0.192

Table 1. **NuScenes val set performance.** Our VEDet outperforms existing baselines consistently cross various backbone and resolution choices. † means initializing from FCOS3D backbone. ‡ means initializing from the depth-pretrained backbone provided by DD3D [31].

Method	Backbone	Image size	TTA	mAP↑	NDS↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
DD3D‡	V2-99	900×1600	✓	0.418	0.477	0.572	0.249	0.368	1.014	0.124
DETR3D‡	V2-99	900×1600	✓	0.412	0.479	0.641	0.255	0.394	0.845	0.133
PETR‡	V2-99	640×1600		0.434	0.481	0.641	0.248	0.437	0.894	0.143
UVTR‡	V2-99	900×1600		0.472	0.551	0.577	0.253	0.391	0.508	0.123
BEVFormer‡	V2-99	900×1600		0.481	0.569	0.582	0.256	0.375	0.378	0.126
BEVDet4D	Swin-B	900×1600	✓	0.451	0.569	**0.511**	0.241	0.386	**0.301**	0.121
PolarFormer‡	V2-99	900×1600		0.493	0.572	0.556	0.256	0.364	0.439	0.127
PETRv2‡	V2-99	640×1600		0.490	0.582	0.561	**0.243**	0.361	0.343	**0.120**
VEDet ‡	V2-99	640×1600		**0.505**	**0.585**	0.545	0.244	**0.346**	0.421	0.123

Table 2. **NuScenes test set performance.** Our VEDet achieves state-of-the-art performance compared to existing publications. We also note that other baselines are trained with CBGS which is a more advantegeous sampling strategy. ‡ means initializing from the depth-pretrained backbone provided by DD3D [31].

data augmentation, we apply random resize, horizontal flip, and crop; and for the 3D space we apply random scaling and rotation following CenterPoint [46]. Importantly, when flipping the input image, we flip the 3D box annotations and camera extrinsics accordingly. The algorithm-specific hyper-parameters are ablated in Sec. 4.3 and fixed for all experiments. Please see the supplemental material's full list of hyper-parameters and more training details. All experiments are trained for 24 epochs except for test submission, which is trained for 96 epochs without CBGS [48].

4.2. Comparison to state of the art

Our VEDet achieves state-of-the-art detection performance on NuScenes val set across a range of model setups as shown in Tab. 1, compared to previous works and some concurrent preprints [15, 26]. We use ImageNet-pretrained models for setups with ResNet-50/101 backbones to process image resolutions of 384×1056 and 512×1408, and outperform existing baselines. As the full-fledged setup, we adopt a V2-99 backbone initialized with depth-pretrained weights [31], operating on 640×1600 images. In this high-performance regime we compare two closely related works, PETR [25], and PETRv2 [26], as shown in the third

group. VEDet surpasses PETRv2 by 2.0% mAP and 1.0% NDS, excelling at 3 sub-metrics. Our *single-frame* version VEDet-SF also achieves substantial gain over the single-frame baseline PETR, by 2.9% mAP and 3.9% NDS; it even outperforms the two-frame PETRv2 by 0.2% mAP and at 3 sub-metrics. The noticeably lower mATE scores of VEDet and VEDet-SF further verify the strong localization capability. For the test submission we adopt the depth-pretrained V2-99 backbone from [31] with 640×1600 images. Without using the more advantageous data sampling strategy CBGS [48] as all the other baselines do, VEDet still outperforms PETRv2 and achieves state-of-the-art performance with 50.5% mAP and 58.5% NDS.

4.3. Ablation studies

We first ablate the choices of some important hyper-parameters used in VEDet shown in Tabs. 3a to 3d, and then analyze the most critical components proposed in VEDet by adding one component at a time as shown in Tab. 3e or making variations of specific components shown in Tab. 3f.

Hyper-parameter selection. We ablate the maximum Fourier frequency f_{max} Tab. 3a and number of Fourier bands k Tab. 3b used in Eqs. (2) and (6), the number of vir-

$f_{\max}$	mAP↑	NDS↑
4	0.446	0.521
8	**0.451**	**0.527**
16	0.447	0.521
32	0.442	0.517

(a) **Fourier frequency.**

k	mAP↑	NDS↑
32	0.446	0.522
64	**0.451**	**0.527**
96	0.445	0.520
128	0.438	0.512

(b) **# Fourier bands.**

V	mAP↑	NDS↑
0	0.432	0.495
1	0.441	0.518
2	**0.451**	**0.527**
4	0.448	0.526
6	0.437	0.511

(c) **# virtual views**

λ_{v}	mAP↑	NDS↑
0.1	0.442	0.522
0.2	**0.451**	**0.527**
0.4	0.439	0.517
0.6	0.431	0.499

(d) **virtual view weights.**

#	Fourier+MLP GPE.	2 virtual views	2-frame	mAP↑	NDS↑
1				0.404	0.447
2	✓			0.420	0.464
3	✓	✓		0.434	0.488
4	✓		✓	0.432	0.495
5	✓	✓	✓	**0.451**	**0.527**

(e) **components addition.**

method	mAP↑	NDS↑
VEDet	**0.451**	**0.527**
no Fourier	0.382	0.476
no $\bar{\mathbf{q}}$	0.436	0.515
no $\mathbf{t}$	0.353	0.453
no joint match	0.425	0.483

(f) **components variation**

Table 3. **Ablation studies and analyses.** In Tabs. 3a to 3d we first analyze important hyper-parameters relevant to the algorithm, and choose the ones giving the best performance. In Tab. 3e we ablate and show the importance of the proposed geometric embeddings and virtual views. They not only monotonically bring improvements to the 3D detection performance, but also mutually benefit adding time frames. In Tab. 3f we ablate some variations in the components, showing the critical usage of Fourier encoding and some optimal settings for queries during training and inference.

tual query views V Tab. 3c and their weighting λ_{v} Tab. 3d used in box regression loss. The table shows that VEDet is robust across a wide range of hyper-parameter choices with competitive performance. Importantly, in Tab. 3c starting from not using any virtual query views ($V = 0$), adding more views gradually improves performance until too many views might have caused the optimization to be more difficult. We choose $f_{\max} = 8, k = 64, V = 2, \lambda_{\mathrm{v}} = 0.2$ according to the best results and fix them for all experiments.

Geometric positional encodings and object queries. In Tab. 3e, we note that #1 is effectively PETR [25]. The "Fourier+MLP GPE" column means we switch the position embeddings and queries in PETR with ours introduced in Secs. 3.2 and 3.3. From #1 to #2 we can see significant improvements in mAP by 1.6% and NDS by 1.7%, which demonstrates that *our proposed implicit geometric mapping better captures the 3D geometries* thanks to its Fourier component and the use of geometric attributes $[\mathbf{r}, \bar{\mathbf{q}}, \mathbf{t}]$.

Virtual query views. In Tab. 3e, we then add $V = 2$ virtual views during training on top of adopting our proposed geometric position embeddings and object queries, for both single-frame and 2-frame ("2-frame" column) settings. From #2 to #3 , we see further jumps in mAP by 1.4% and NDS by 2.4%; from #4 to #5, mAP increases by 1.9% and NDS by 3.2%. This shows *our proposed multi-view consistency loss applied on virtual view decoding effectively guides the model to improve 3D detection*.

Mutual benefit between VEDet and multi-sweep. Based on the above two comparisons, another observation is that

our proposed VEDet benefits in the 2-frame setting more than the single-frame setting, indicating that more geometric cues can be exploited when the input images contain richer multi-view context. Similarly, when looking at #2 to #4 (+1.2% mAP, +3.1% NDS) and #3 to #5 (+1.7% mAP, +3.9% NDS), we can see that adding more frames becomes more helpful after we add in the viewpoint equivariance on $V = 2$ views. These two observations further consolidate the effectiveness of our geometric position embeddings, object queries, and virtual views.

Fourier encoding. In Tab. 3f "no Fourier" we show the importance of the Fourier encoding before the MLP by simply dropping it for both position embeddings and object queries, such that the MLPs map the geometric primitives directly to the 256-dim vectors. This leads to a drastic decrease in the detection performance (-6.9% mAP and -5.1% NDS), showing the critical role of the Fourier encoding to capture fine-grained changes in the geometries, which can be considered as high-frequency signals [36].

Partial camera geometry. In Tab. 3f "no $\bar{\mathbf{q}}$" and "no $\mathbf{t}$" we show that removing partial information from the input cameras' poses leads to noticeably degraded performance. Specifically, removing the rotation information leads to drops in mAP by 1.5% and NDS by 1.2%, since the rotation indicates how the perspective projection plane is facing, which the rays are insufficient to describe; removing the translation leads to catastrophic drops in mAP by 9.2% and NDS by 7.4%, justifying the importance of translation information which is too difficult to infer from data implicitly if missing.
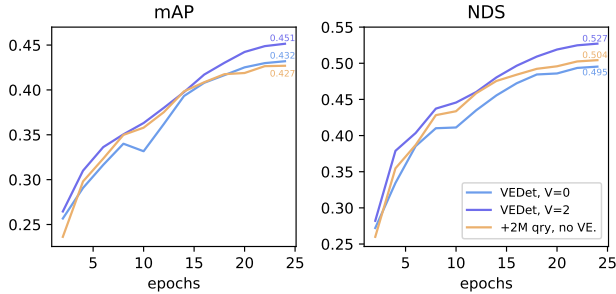
Figure 4. **Effectiveness of the view equivariant regularization.** Simply adding more queries and duplicating box targets without introducing view equivariance only helps model optimization at an early stage and does not help the final detection performance as shown by the orange curves compared to the blue curves. VEDet is able to leverage richer geometric signals brought by the view equivariance objective with the help of view-conditioned queries, which leads to a performance boost shown by purple curves.

Multi-view consistency. In Sec. 3.4, we constrain the different views of a query point to be considered simultaneously by concatenating the box predictions for matching and calculating loss. Instead, in Tab. 3f "no joint match" we treat them as individual objects. This leads to the possibility that a query point with different query views can be matched to different boxes, which leads to 2.6% drop in mAP and 3.9% drop in NDS. This demonstrates the multi-view consistency of the *same* query point is meaningful, whereas simply augmenting the queries with views and treating all queries individually does not exploit the geometric signals enough. Without multi-view consistency, the excessive number of queries might be even harmful to the optimization, as indicated by the lower performance (42.5% mAP, 48.3% NDS) than the "$V = 0$" VEDet (43.2% mAP, 49.5% NDS) in Tab. 3c.

4.4. More analysis on viewpoint equivariance

Given our design of view-conditioned queries and utilizing multiple virtual views during training, a natural question arises: does the performance gain come from the viewpoint equivariance regularization, or simply because more queries participate in the training? To show the effectiveness of the viewpoint equivariance, we compare to an intuitive baseline described as follows. We start from a plain version where we do not apply virtual views in queries (i.e. setting $V = 0$). Under this setting, we add an additional set of $V \times M$ query points during training while duplicating the box targets by V more times, and only use the original M queries during inference. This setting *matches the number of participating queries and box targets as VEDet but does not contain any viewpoint equivariance regularization*, denoted by "$+2M$ qry, no VE.." in Fig. 4, for NuScenes val set performances.

As shown in Fig. 4, the extra queries and box targets generate more learning signals at the early stage as reflected by the superior mAP and NDS (orange curves) compared to our "$V = 0$" VEDet (blue curves). However, their effects diminish as the curves plateau when reaching the end, and eventually "$+2M$ qry, no VE." underperforms "VEDet-0 views", as shown in both plots of Fig. 4. In contrast, Our counterpart VEDet with 2 virtual views during training, denoted by "VEDet, $V = 2$" (purple curve), outperforms "VEDet, $V = 0$" consistently throughout the training and noticeably boosts the performance by +1.9% mAP and +3.2% NDS to 45.1% mAP to 52.7% NDS. This justifies the viewpoint equivariance regularization is more than just increasing the number of queries, and that it brings richer geometric learning signals for the model.

5. Limitations

Camera parameter robustness. VEDet leverage implicit geometric encodings to learn 3D geometry in a data-driven way, therefore the robustness of VEDet against camera parameters is critical and worth investigating in future works.

Depth information. This work mainly leverages geometric signals generated from the ground truth 3D bounding boxes and generic pose information of both cameras and object queries, while some concurrent works [13, 21] explicitly use depth to guide the 3D feature construction and hence the interaction between the heads and features. Incorporating depth information into the framework, such as enhancing the geometric positional encodings or guiding spatial attention, will be explored for future works.

Temporal modeling. The current VEDet follows existing works to simply concatenate features from multi-sweeps along the token dimension. While this is effective, the concatenation has two main issues: it throws away the temporal ordering, and has difficulty in modeling long sequences due to memory and computation constraints. Investigation on a better temporal modeling such as recurrent processing will be valuable for future works.

6. Conclusion

In this work, we introduce a novel camera-based multi-view 3D object detection framework that learns from viewpoint equivariance regularization. VEDet employs a transformer decoder with a set of view-conditioned queries to decode bounding boxes from image features with geometric positional encodings. The view-conditioning of queries enables us to enforce viewpoint equivariance on predictions made from different viewpoints, and therefore generate richer geometric learning signals to guide the model in better understanding the 3D structure of the scene. VEDet achieves state-of-the-art 3D detection performance, and we conduct extensive experiments to show the effectiveness of its components. We also point out a few meaningful limitations for future works.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Rick Szeliski. Building rome in a day. *Communications of the ACM*, 54:105–112, 2011. 2

[2] Junaid Ahmed Ansari, Sarthak Sharma, Anshuman Majumdar, J Krishna Murthy, and K Madhava Krishna. The earth ain't flat: Monocular reconstruction of vehicles on steep and graded roads from a moving camera. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8404–8410. IEEE, 2018. 2

[3] Ivan Barabanau, Alexey Artemov, Evgeny Burnaev, and Vyacheslav Murashkin. Monocular 3d object detection via geometric reasoning on keypoints. *arXiv preprint arXiv:1905.05618*, 2019. 2

[4] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9287–9296, 2019. 2

[5] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 1

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2, 3, 4, 5

[7] Florian Chabot, Mohamed Chaouch, Jaonary Rabarisoa, Céline Teuliere, and Thierry Chateau. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2040–2049, 2017. 2

[8] Hansheng Chen, Pichao Wang, Fan Wang, Wei Tian, Lu Xiong, and Hao Li. Epro-pnp: Generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2781–2790, 2022. 2

[9] 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. *arXiv preprint arXiv:2010.11929*, 2020. 2

[10] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 2022. 2

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[12] Tong He and Stefano Soatto. Mono3d++: Monocular 3d vehicle detection with two-scale 3d hypotheses and task priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8409–8416, 2019. 2

[13] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022. 8

[14] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021. 2

[15] 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. *arXiv preprint arXiv:2206.15398*, 2022. 1, 2, 6

[16] Jason Ku, Alex D Pon, and Steven L Waslander. Monocular 3d object detection leveraging accurate proposals and shape reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11867–11876, 2019. 2

[17] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 5

[18] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13906–13915, 2020. 5

[19] Buyu Li, Wanli Ouyang, Lu Sheng, Xingyu Zeng, and Xiaogang Wang. Gs3d: An efficient 3d object detection framework for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1019–1028, 2019. 2

[20] Yanwei Li, Yilun Chen, Xiaojuan Qi, Zeming Li, Jian Sun, and Jiaya Jia. Unifying voxel-based representation with transformer for 3d object detection. *arXiv preprint arXiv:2206.00630*, 2022. 2

[21] 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. *arXiv preprint arXiv:2206.10092*, 2022. 1, 2, 8

[22] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022. 1, 2

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 5

[24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 5

[25] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d

object detection. *arXiv preprint arXiv:2203.05625*, 2022. 1, 2, 3, 4, 6, 7

[26] 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, 2, 3, 4, 6

[27] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[29] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 3

[30] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7074–7082, 2017. 2

[31] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3142–3152, 2021. 1, 2, 6

[32] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European Conference on Computer Vision*, pages 194–210. Springer, 2020. 1, 2

[33] Rui Qian, Divyansh Garg, Yan Wang, Yurong You, Serge Belongie, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. End-to-end pseudo-lidar for image-based 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5881–5890, 2020. 1, 2

[34] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1991–1999, 2019. 1, 2

[35] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020. 3

[36] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020. 3, 7

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 3

[38] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 913–922, 2021. 1, 2

[39] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019. 1, 2

[40] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, pages 180–191. PMLR, 2022. 1, 2, 4

[41] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 2567–2575, 2022. 3

[42] Xinshuo Weng and Kris Kitani. Monocular 3d object detection with pseudo-lidar point cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 1, 2

[43] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *arXiv*, 2021. 3

[44] Bin Xu and Zhenzhong Chen. Multi-level fusion based 3d object detection from monocular images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2345–2353, 2018. 2

[45] Wang Yifan, Carl Doersch, Relja Arandjelović, João Carreira, and Andrew Zisserman. Input-level inductive biases for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6176–6186, 2022. 2, 3

[46] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021. 6

[47] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. *arXiv preprint arXiv:1906.06310*, 2019. 2

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