MonoDETR: Depth-guided Transformer for Monocular 3D Object Detection

Renrui Zhang[1,2], Han Qiu[2], Tai Wang[1,2], Ziyu Guo[2], Ziteng Cui[2]
Yu Qiao[2], Hongsheng Li[†1,2,3], Peng Gao[†2]

[1]CUHK MMLab   [2]Shanghai Artificial Intelligence Laboratory
[3]Centre for Perceptual and Interactive Intelligence (CPII)

{zhangrenrui, wangtai, gaopeng}@pjlab.org.cn, hsli@ee.cuhk.edu.hk

Abstract

*Monocular 3D object detection has long been a challenging task in autonomous driving. Most existing methods follow conventional 2D detectors to first localize object centers, and then predict 3D attributes by neighboring features. However, only using local visual features is insufficient to understand the scene-level 3D spatial structures and ignores the long-range inter-object depth relations. In this paper, we introduce the first DETR framework for **Mono**cular **DE**tection with a depth-guided **TR**ansformer, named **MonoDETR**. We modify the vanilla transformer to be depth-aware and guide the whole detection process by contextual depth cues. Specifically, concurrent to the visual encoder that captures object appearances, we introduce to predict a foreground depth map, and specialize a depth encoder to extract non-local depth embeddings. Then, we formulate 3D object candidates as learnable queries and propose a depth-guided decoder to conduct object-scene depth interactions. In this way, each object query estimates its 3D attributes adaptively from the depth-guided regions on the image and is no longer constrained to local visual features. On KITTI benchmark with monocular images as input, MonoDETR achieves state-of-the-art performance and requires no extra dense depth annotations. Besides, our depth-guided modules can also be plug-and-play to enhance multi-view 3D object detectors on nuScenes dataset, demonstrating our superior generalization capacity. Code is available at https: //github.com/ZrrSkywalker/MonoDETR.*

1. Introduction

With a wide range of applications in autonomous driving, 3D object detection is more challenging than its 2D counterparts, due to the complex spatial circumstances.
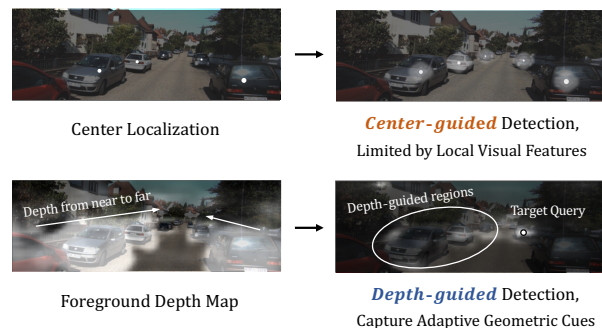
---

† Corresponding author

Figure 1. **Center-guided (Top) and Depth-guided Paradigms (Bottom) for monocular 3D object detection.** Existing center-guided methods predict 3D attributes from local visual features around the centers, while our MonoDETR guides the detection by a predicted foreground depth map and adaptively aggregates features in global context. The lower right figure visualizes the attention map of the target query in the depth cross-attention layer.

Compared to methods based on LiDAR [57, 20, 41, 50] and multi-view images [49, 22, 26, 17], 3D object detection from monocular (single-view) images [9, 1, 46] is of most difficulty, which generally does not rely on depth measurements or multi-view geometry. The detection accuracy thus severely suffers from the ill-posed depth estimation, leading to inferior performance.

Except for leveraging pseudo 3D representations [48, 51, 32, 38], standard monocular 3D detection methods [33, 52, 53, 31] follow the pipeline of traditional 2D object detection [40, 24, 43, 56]. They first localize objects by detecting the projected centers on the image, and then aggregate the neighboring visual features for 3D property prediction, as illustrated in Figure 1 (Top). Although it is conceptually straightforward, such center-guided methods are limited by the local appearances without long-range context, and fail to capture implicit geometric cues from 2D images, e.g., depth guidance, which are critical to detect objects in 3D space.

To tackle this issue, we propose MonoDETR, which

(a) DETR′s
Visual Transformer

(b) MonoDETR′s
Depth-guided Transformer

(c) PETR (v2)′s
3D Transformer
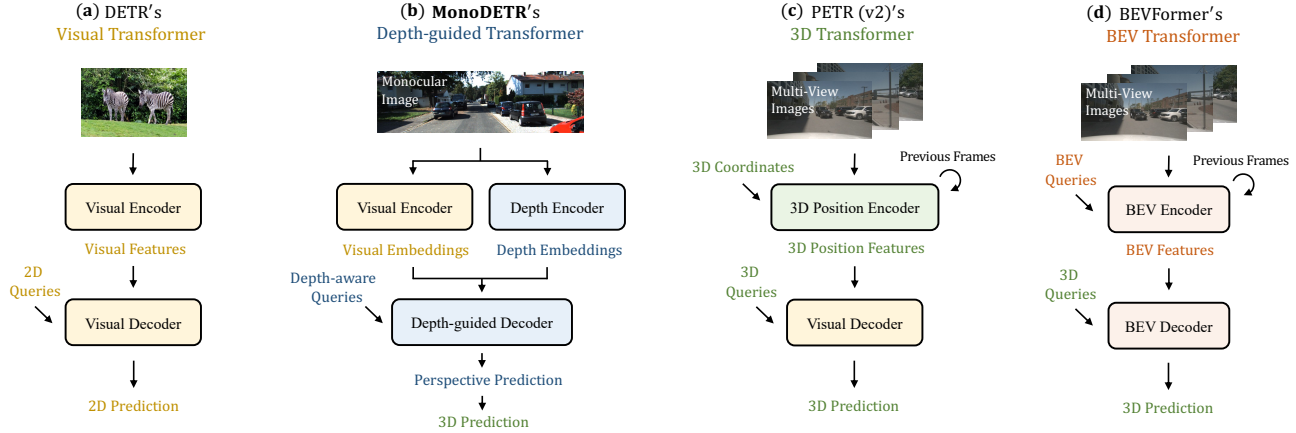
(d) BEVFormer′s
BEV Transformer

Figure 2. **Comparison of DETR-based methods for camera-based 3D object detection.** We utilize yellow, blue, green, and red to respectively denote the feature or prediction space related with 2D, depth, 3D, and BEV. Different from other methods, our MonoDETR leverages depth cues to guide 3D object detection from monocular images.

presents a novel depth-guided 3D detection scheme in Figure 1 (Bottom). Compared to DETR [4] in 2D detection, the transformer in MonoDETR is equipped with depth-guided modules to better capture contextual depth cues, serving as ***the first*** DETR model for monocular 3D object detection, as shown in Figure 2 (a) and (b). It consists of: two parallel encoders for visual and depth representation learning, and a decoder for adaptive depth-guided detection.

Specifically, after the feature backbone, we first utilize a lightweight depth predictor to acquire the depth features of the input image. To inject effective depth cues, a foreground depth map is predicted on top, and supervised only by discrete depth labels of objects, which requires no dense depth annotations during training. Then, we apply the parallel encoders to respectively generate non-local depth and visual embeddings, which represent the input image from two aspects: depth geometry and visual appearance. On top of that, a set of object queries is fed into the depth-guided decoder, and conducts adaptive feature aggregation from the two embeddings. Via a proposed depth cross-attention layer, the queries can capture geometric cues from the depth-guided regions on the image, and explore inter-object depth relations. In this way, the 3D attribute prediction can be guided by informative depth hints, no longer constrained by the limited visual features around centers.

As an end-to-end transformer-based network, MonoDETR is free from non-maximum suppression (NMS) or rule-based label assignment. We only utilize the object-wise labels for supervision without using auxiliary data, such as dense depth maps or LiDAR. Taking monocular images as input, MonoDETR achieves *state-of-the-art* performance among existing center-guided methods, and surpasses the second-best by +2.53%, +1.08%, and +0.85% for three-level difficulties on KITTI [14] *test* set.

Besides single-view images, the depth-guided mod-

ules in MonoDETR can also be extended as a plug-and-play module for multi-view 3D detection on nuScenes [3] dataset. By providing multi-view depth cues, our method can not only improve the end-to-end detection performance of PETRv2 [27] by +1.2% NDS, but also benefit the BEV representation learning in BEVFormer [22] by +0.9% NDS. This further demonstrates the effectiveness and generalizability of our proposed depth guidance.

We summarize the contributions of our paper as follows:

- We propose MonoDETR, a depth-guided framework to capture scene-level geometries and inter-object depth relations for monocular 3D object detection.

- We introduce a foreground depth map for object-wise depth supervision, and a depth cross-attention layer for adaptive depth features interaction.

- MonoDETR achieves leading results on monocular KITTI benchmark, and can also be generalized to enhance multi-view detection on nuScenes benchmark.

2. Related Work

Existing methods for camera-based 3D object detection can be categorized as two groups according to the input number of views: monocular (single-view) and multi-view methods. Monocular detectors only take as input the front-view images and solve a more challenging task from insufficient 2D signals. Multi-view detectors simultaneously encode images of surrounding scenes and can leverage cross-view dependence to understand the 3D space.

Monocular (Single-view) 3D Object Detection. Most previous monocular detectors adopt center-guided pipelines following conventional 2D detectors [40, 43, 56]. As early

Method	DETR-based	Extra Data	Guided by	Object Query	Feat. Aggre.
Mutli-view Methods					
DETR3D [49]	✓	-	Visual	3D	Local
PETR (v2) [26]	✓	Temporal	Visual	3D	Global
BEVFormer [22]	✓	Temporal	Visual	BEV, 3D	Global
Monocular Methods					
MonoDTR [18]	✗	LiDAR	Center	✗	Local
MonoDETR	✓	-	**Depth**	**Depth**	Global

Table 1. **Comparison of DETR-based methods for camera-based 3D object detection.** Our MonoDETR is uniquely guided by depth cues with depth-aware object queries.
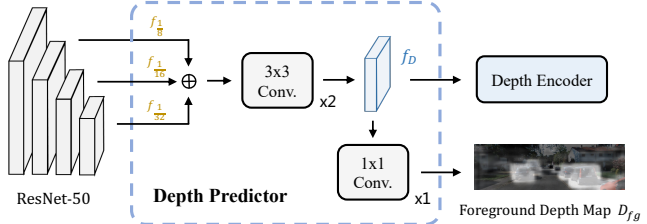

Figure 3. **The lightweight depth predictor.** We utilize the depth predictor to predict the depth features and foreground depth map, which only contains discrete object-wise depth values.

works, Deep3DBox [35] introduces discretized representation with perspective constraints, and M3D-RPN [1] designs a depth-aware convolution for better 3D region proposals. With very few handcrafted modules, SMOKE [28] and FCOS3D [46] propose concise architectures for efficient one-stage detection, while MonoDLE [33] and PGD [47] analyze depth errors on top with improved performance. To supplement the limited 3D cues, additional data are utilized for assistance: dense depth annotations [32, 12, 45, 36], CAD models [29], and LiDAR [7, 38, 18]. Some recent methods introduce complicated geometric priors into the networks: adjacent object pairs [10], 2D-3D keypoints [21], and uncertainty-related depth [52, 31]. Despite this, the center-guided methods are still limited by local visual features without scene-level spatial cues. In contrast, MonoDETR discards the center localization step and conducts adaptive feature aggregation via a depth-guided transformer. MonoDETR requires no additional annotations and contains minimal 2D-3D geometric priors.

Multi-view 3D Object Detection. For jointly extracting features from surrounding views, DETR3D [49] firstly utilizes a set of 3D object queries and back-projects them onto multi-view images for feature aggregation. PETR series [26, 27] further proposes to generate 3D position features without unstable projection and explores the advantage of temporal information from previous frames. From another point of view, BEVDet [17, 16] follows [37] to lift 2D images into a unified Bird's-Eye-View (BEV) representation and appends BEV-based heads [50] for detection. BEVFormer [22] instead generates BEV features via a set of learnable BEV queries, and introduces a spatiotemporal BEV transformer for visual features aggregation. Follow-up works also introduce cross-modal distillation [19, 11] and masked image modeling [25, 6] for improved performance. Different from the above methods for multi-view input, MonoDETR targets monocular images and extracts depth guidance to capture more geometric cues. Our depth-guided modules can also be generalized to surrounding views as a plug-and-play module to enhance multi-view detectors.

Comparison of DETR-based Methods. DETR [5] and its follow-up works [13, 59, 54, 34] have attained great success on 2D object detection without NMS or anchors. Inspired by this, some efforts have transferred DETR into camera-based 3D object detection. We specifically compare our MonoDETR with existing DETR-based 3D object detectors in Figure 2 and Table 1. **(1) MonoDTR [18].** Also as a single-view detector, MonoDTR utilizes transformers [44] to incorporate depth features with visual representations. However, MonoDTR is *not a DETR-based method*, and still adopts the traditional center-guided paradigm, which localizes objects by their centers and only aggregate local features. MonoDTR contains no object queries for global feature aggregation, and follows YOLOv3 [39] to adopt complicated NMS post-processing with pre-defined anchors. **(2) DETR3D [49] and PETR (v2) [26, 27]** (Figure 2 (c)) are multi-view methods and follow the DETR detection pipeline. In contrast, they contain no transformer-based encoders (visual or depth), and detect objects by 3D object queries without the perspective transformation. Importantly, they are only guided by visual features and explores no depth cues from the input images. **(3) BEVFormer [22]** (Figure 2 (d)) firstly utilizes a BEV transformer to lift multi-view images into BEV representations, and then conducts DETR-based detection within the BEV space. Different from all aforementioned methods, MonoDETR introduces a unique depth-guided transformer that guides the 3D detection by geometric depth cues, which can generalize well to both monocular and multi-view inputs.

3. Method

The overall framework of MonoDETR is shown in Figure 4. We first illustrate the concurrent visual and depth feature extraction in Section 3.1, and detail our depth-guided transformer for aggregating appearance and geometric cues in Section 3.2. Then, we introduce the attribute prediction and loss functions of MonoDETR in Section 3.3. Finally, we illustrate how to plug our depth-guided transformer into existing multi-view object detectors in Section 3.4.
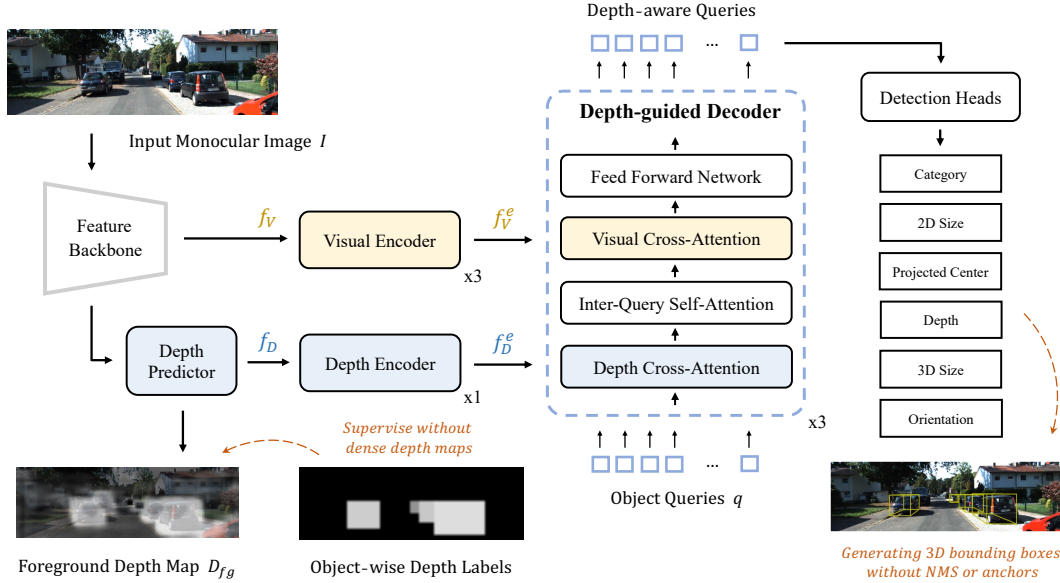
Figure 4. **Overall pipeline of MonoDETR.** We first acquire the visual and depth features of the input image and utilize two parallel encoders for non-local encoding. Then, we propose a depth-guided decoder to adaptively aggregate scene-level features in global context.

3.1. Feature Extraction

Taking as input a monocular (single-view) image, our framework utilizes a feature backbone, e.g., ResNet-50 [15], and a lightweight depth predictor to generate its visual and depth features, respectively.

Visual Features. Given the image $I \in \mathbb{R}^{H \times W \times 3}$, where H and W denote its height and width, we obtain its multi-scale feature maps, $f_{\frac{1}{8}}$, $f_{\frac{1}{16}}$, and $f_{\frac{1}{32}}$, from the last three stages of ResNet-50. Their downsample ratios to the original size are $\frac{1}{8}$, $\frac{1}{16}$ and $\frac{1}{32}$. We regard the highest-level $f_{\frac{1}{32}} \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times C}$ with sufficient semantics as the visual features f_V of the input image.

Depth Features. We obtain the depth features from the image by a lightweight depth predictor, as shown in Figure 3. We first unify the sizes of three-level features to the same $\frac{1}{16}$ downsample ratio via bilinear pooling, and fuse them by element-wise addition. In this way, we can integrate multi-scale visual appearances and also preserve fine-grained patterns for objects of small sizes. Then, we apply two 3×3 convolutional layers to obtain the depth features $f_D \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C}$ for the input image.

Foreground Depth Map. To incorporate effective depth information into the depth features, we predict a foreground depth map $D_{fg} \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times (k+1)}$ on top of f_D via a 1×1 convolutional layer. We supervise the depth map only by discrete object-wise depth labels, without extra dense depth annotations. The pixels within the same 2D bounding box are assigned with the same depth label of the corresponding object. For pixels within multiple boxes, we select the

depth label of the object that is nearest to the camera, which accords with the visual appearance of the image. Here, We discretize the depth into $k+1$ bins [38], where the first ordinal k bins denote foreground depth and the last one denotes the background. We adopt linear-increasing discretization (LID), since the larger depth estimation errors of farther objects can be suppressed with a wider categorization interval. We limit the foreground depth values within $[d_{min}, d_{max}]$, and set both the first interval length and LID's common difference as δ. We then categorize a ground-truth depth label d into the k-th bin as:

$$k = \lfloor -0.5 + 0.5\sqrt{1 + \frac{8(d - d_{min})}{\delta}} \rfloor, \qquad (1)$$

where $\delta = \frac{2(d_{max} - d_{min})}{k(k+1)}$. By focusing on the object-wise depth values, the network can better capture foreground spatial structures and inter-object depth relations, which produces informative depth features for the subsequent depth-guided transformer.

3.2. Depth-guided Transformer

The depth-guided transformer of MonoDETR is composed of a visual encoder, a depth encoder, and a depth-guided decoder. The two encoders produce non-local visual and depth embeddings, and the decoder enables object queries to adaptively capture scene-level information.

Visual and Depth Encoders. Given depth and visual features f_D, f_V, we specialize two transformer encoders to generate their scene-level embeddings with global receptive fields, denoted as $f_D^e \in \mathbb{R}^{\frac{HW}{16^2} \times C}$ and $f_V^e \in \mathbb{R}^{\frac{HW}{32^2} \times C}$.

We set three blocks for the visual encoder and only one block for the depth encoder, since the discrete foreground depth information is easier to be encoded than the rich visual appearances. Each encoder block consists of a self-attention layer and a feed-forward neural network (FFN). By the global self-attention mechanism, the depth encoder explores long-range dependencies of depth values from different foreground areas, which provides non-local geometric cues of the stereo space. In addition, the decoupling of depth and visual encoders allows them to better learn features for themselves, encoding the input image from two perspectives, i.e., depth geometry and visual appearance.

Depth-guided Decoder. Based on the non-local f_D^e, f_V^e, we utilize a set of learnable object queries $q \in \mathbb{R}^{N \times C}$ to detect 3D objects via the depth-guided decoder, where N denotes the pre-defined maximum number of objects in the input image. Each decoder block sequentially contains a depth cross-attention layer, an inter-query self-attention layer, a visual cross-attention layer, and an FFN. Specifically, the queries first capture informative depth features from f_D^e via the depth cross-attention layer, in which we linearly transform the object queries and depth embeddings into queries, keys, and values,

$$Q_q = \text{Linear}(q), \quad K_D, V_D = \text{Linear}(f_D^e), \quad (2)$$

where $Q_q \in \mathbb{R}^{N \times C}$ and $K_D, V_D \in \mathbb{R}^{\frac{HW}{16^2} \times C}$. Then, we calculate the query-depth attention map $A_D \in \mathbb{R}^{N \times \frac{HW}{16^2}}$, and aggregate informative depth features weighted by A_D to produce the depth-aware queries q', formulated as,

$$A_D = \text{Softmax}(Q_q K_D^T / \sqrt{C}), \quad (3)$$
$$q' = \text{Linear}(A_D V_D). \quad (4)$$

Such a mechanism enables each object query to adaptively capture spatial cues from depth-guided regions on the image, leading to better scene-level spatial understanding. Then, the depth-aware queries are fed into the inter-query self-attention layer for feature interaction between objects, and the visual cross-attention layer for collecting visual semantics from f_V^e. We stack three decoder blocks to fully fuse the scene-level depth cues into object queries.

Depth Positional Encodings. In the depth cross-attention layer, we propose learnable depth positional encodings for f_D^e instead of conventional sinusoidal functions. In detail, we maintain a set of learnable embeddings, $p_D \in \mathbb{R}^{(d_{max} - d_{min} + 1) \times C}$, where each row encodes the depth positional information for a meter, ranging from d_{min} to d_{max}. For each pixel (x, y) in D_{fg}, we first obtain its $(k + 1)$-categorical depth prediction confidence, $D_{fg}(x, y) \in \mathbb{R}^{k+1}$, each channel of which denotes the predicted confidence for the corresponding depth bin. The estimated depth

of pixel (x, y) can then be obtained by the weighted summation of the depth-bin confidences and their corresponding depth values, which is formulated as

$$d_{map}(x, y) = \sum_{i=1}^{k+1} D_{fg}(x, y)[i] \cdot d_{bin}^i, \quad (5)$$

where d_{bin}^i denotes the starting value of the i-th depth bin and $\sum_{i=1}^{k+1} D_{fg}(x, y)[i] = 1$. Then, we linearly interpolate p_D according to the depth $d_{map}(x, y)$ to obtain the depth positional encoding for the pixel (x, y). By pixel-wisely adding f_D^e with such encodings, object queries can better capture scene-level depth cues and understand 3D geometry in the depth cross-attention layer.

3.3. Detection Heads and Loss

After the decoder, the depth-aware queries are fed into a series of MLP-based heads for 3D attribute predictions, including object category, 2D size, projected 3D center, depth, 3D size, and orientation. For inference, we convert these perspective attributes into 3D-space bounding boxes using camera parameters without NMS post-processing or pre-defined anchors. For training, we match the orderless queries with ground-truth labels and compute losses for the paired ones. We refer to Supplementary Material for details.

Bipartite Matching. To correctly match each query with ground-truth objects, we calculate the loss for each query-label pair and utilize Hungarian algorithm [4] to find the globally optimal matching. For each pair, we integrate the losses of six attributes into two groups. The first contains object category, 2D size and the projected 3D center, since these attributes mainly concern 2D visual appearances of the image. The second group consists of depth, 3D size and orientation, which are 3D spatial properties of the object. We respectively sum the losses of two groups and denote them as $\mathcal{L}_{2D}$ and $\mathcal{L}_{3D}$. As the network generally predicts less accurate 3D attributes than 2D attributes, especially at the beginning of training, the value of $\mathcal{L}_{3D}$ is unstable and would disturb the matching process. We only utilize $\mathcal{L}_{2D}$ as the matching cost for matching each query-label pair.

Overall Loss. After the matching, we obtain N_{gt} valid pairs out of N queries, where N_{gt} denotes the number of ground-truth objects. Then, the overall loss of a training image is formulated as

$$\mathcal{L}_{overall} = \frac{1}{N_{gt}} \cdot \sum_{n=1}^{N_{gt}} (\mathcal{L}_{2D} + \mathcal{L}_{3D}) + \mathcal{L}_{dmap}, \quad (6)$$

where $\mathcal{L}_{dmap}$ represents the Focal loss [23] of the predicted categorical foreground depth map D_{fg} in Section 3.1.

(a) PETR (v2) with
Depth-guided 3D Transformer
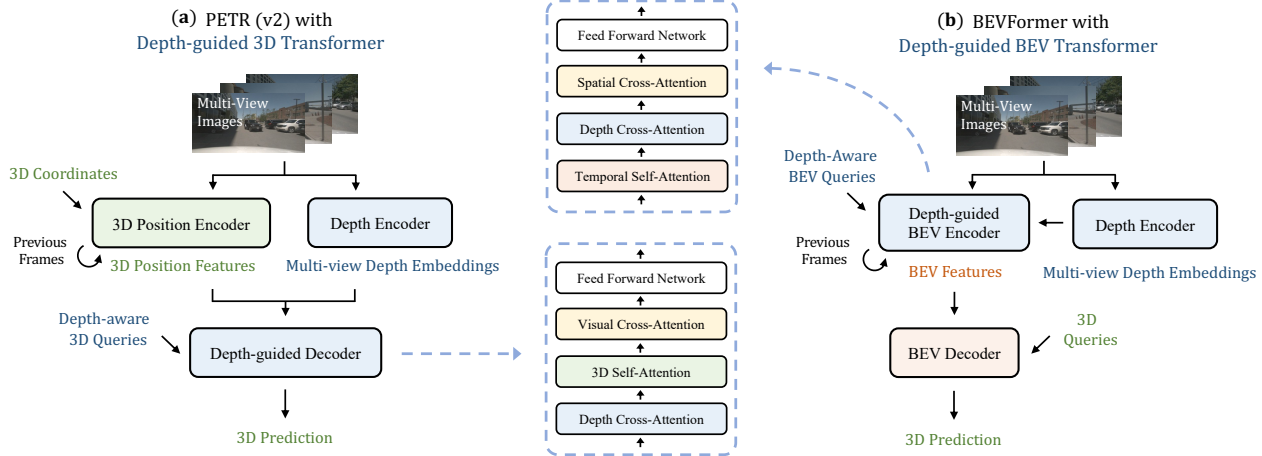
(b) BEVFormer with
Depth-guided BEV Transformer

Figure 5. **Plug-and-play for multi-view 3D object detection.** We utilize yellow, blue, green, and red to respectively denote the feature space related to 2D, depth, 3D, and BEV. The depth-guided transformer of MonoDETR is adopted to enhance PETR (v2) [26, 27] and BEVFormer [22] in a plug-and-play manner, which provides depth guidance from surrounding scenes.

3.4. Plug-and-play for Multi-view Detectors

Besides monocular images, our depth-guided transformer can also serve as a plug-and-play module upon multi-view methods for depth-guided detection. Specifically, we append our depth predictors and depth encoders after the backbones of multi-view methods, which are shared across views and extract surrounding depth embeddings. Then, we inject our depth cross-attention layer into their transformer blocks to guide the 3D or BEV object queries by scene-level depth cues.

For PETR (v2) [26, 27] in Figure 5 (a), we modify its previous visual decoder as a depth-guided decoder. In each decoder block, the 3D object queries are first fed into our depth cross-attention layer for depth cues aggregation, and then into the original 3D self-attention and visual cross-attention for 3D position features interaction. This enables PETR's 3D queries to be depth-aware and better capture spatial characteristics of surrounding scenes.

For BEVFormer [22] in Figure 5 (b), as its decoder is conducted in BEV space, we incorporate the depth guidance into its BEV encoder, which lifts image features into BEV space by transformers. In each encoder block, the BEV queries also sequentially pass through our depth cross-attention layer and the original spatial cross-attention layer. This contributes to better BEV representation learning guided by the multi-view depth information.

4. Experiments

4.1. Settings

Dataset. We evaluate MonoDETR on the widely-adopted KITTI [14] benchmark, including 7,481 training and 7,518 test images. We follow [8, 9] to split 3,769 *val* images from the training set. We report the detection results with three-level difficulties, easy, moderate, and hard, and evaluate by the average precision (AP) of bounding boxes in 3D space and the bird-eye view, denoted as AP_{3D} and AP_{BEV}, respectively, which are both at 40 recall positions.

Implementation details. We adopt ResNet-50 [15] as our feature backbone. To save GPU memory, we apply deformable attention [59] for the visual encoder and visual cross-attention layers, and utilize the vanilla global attention [4] to better capture non-local geometries for the depth encoder and depth cross-attention layers. We utilize 8 heads for all attention modules and set the number of queries N as 50, which are learnable embeddings with predicted 2D reference points. We set the channel C and all MLP's latent dimensions as 256. For the foreground depth map, we set $[d_{min}, d_{max}]$ as $[0m, 60m]$ and the number of bins k as 80. On a single RTX 3090 GPU, we train MonoDETR for 195 epochs with batch size 16 and a learning rate 2×10^{-4}. We adopt AdamW [30] optimizer with weight decay 10^{-4}, and decrease the learning rate at 125 and 165 epochs by 0.1. For training stability, we discard the training samples with depth labels larger than 65 meters or smaller than 2 meters. During inference, we simply filter out the object queries with the category confidence lower than 0.2 without NMS post-processing, and recover the 3D bounding box using the predicted six attributes following previous works [33, 31].

4.2. Comparison

Performance. In Table 2, MonoDETR achieves *state-of-the-art* performance on KITTI *test* and *val* sets. On *test* set, MonoDETR exceeds all existing methods including those with different additional data input and surpasses

Method	Extra data	Test, AP_{3D}			Test, AP_{BEV}			Val, AP_{3D}		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
PatchNet [32]	Depth	15.68	11.12	10.17	22.97	16.86	14.97	-	-	-
D4LCN [12]		16.65	11.72	9.51	22.51	16.02	12.55	-	-	-
DDMP-3D [45]		19.71	12.78	9.80	28.08	17.89	13.44	-	-	-
Kinematic3D [2]	Video	19.07	12.72	9.17	26.69	17.52	13.10	19.76	14.10	10.47
MonoRUn [7]	LiDAR	19.65	12.30	10.58	27.94	17.34	15.24	20.02	14.65	12.61
CaDDN [38]		19.17	13.41	11.46	27.94	18.91	17.19	23.57	16.31	13.84
MonoDTR [18]		21.99	15.39	12.73	28.59	20.38	17.14	24.52	18.57	15.51
AutoShape [29]	CAD	22.47	14.17	11.36	30.66	20.08	15.59	20.09	14.65	12.07
SMOKE [28]	None	14.03	9.76	7.84	20.83	14.49	12.75	14.76	12.85	11.50
MonoPair [10]		13.04	9.99	8.65	19.28	14.83	12.89	16.28	12.30	10.42
RTM3D [21]		13.61	10.09	8.18	-	-	-	19.47	16.29	15.57
PGD [47]		19.05	11.76	9.39	26.89	16.51	13.49	19.27	13.23	10.65
IAFA [55]		17.81	12.01	10.61	25.88	17.88	15.35	18.95	14.96	14.84
MonoDLE [33]		17.23	12.26	10.29	24.79	18.89	16.00	17.45	13.66	11.68
MonoRCNN [42]		18.36	12.65	10.03	25.48	18.11	14.10	16.61	13.19	10.65
MonoGeo [53]		18.85	13.81	11.52	25.86	18.99	16.19	18.45	14.48	12.87
MonoFlex [52]		19.94	13.89	12.07	28.23	19.75	16.89	23.64	17.51	14.83
GUPNet [31]		20.11	14.20	11.77	-	-	-	22.76	16.46	13.72
MonoDETR (Ours)	None	**25.00**	**16.47**	**13.58**	**33.60**	**22.11**	**18.60**	**28.84**	**20.61**	**16.38**
Improvement	*v.s. second-best*	+2.53	+1.08	+0.85	+2.94	+1.73	+1.41	+4.32	+2.04	+0.81

Table 2. **Monocular performance of the car category on KITTI *test* and *val* sets.** We utilize bold numbers to highlight the best results, and color the second-best ones and our gain over them in blue.

the second-best under three-level difficulties by +2.53%, +1.08% and +0.85% in AP_{3D}, and by +2.94%, +1.73% and +1.41% in AP_{BEV}. The competitive MonoDTR [18] also applies transformers to fuse depth features, but it is still a center-guided method and highly relies on additional dense depth supervision, anchors and NMS. In contrast, MonoDETR performs better without extra input or handcrafted designs, illustrating its simplicity and effectiveness.

Efficiency. Compared to existing methods in Table 3, MonoDETR can achieve the best detection performance without consuming too much computational budget. As illustrated in Section 3, we only process the feature maps with $\frac{1}{16}$ and $\frac{1}{32}$ downsample ratios, which reduces our Runtime and GFlops, while others adopt $\frac{1}{4}$ and $\frac{1}{8}$ ratios.

4.3. Ablation Studies

We verify the effectiveness of each our component and report AP_{3D} for the car category on the KITTI *val* set.

Depth-guided Transformer. In Table 5, we first remove the entire depth-guided transformer along with the depth predictor, which constructs a pure center-guided baseline. This variant, denoted as 'w/o Depth-guided Trans.' can be regarded as a re-implementation of MonoDLE [33] with our detection heads and loss functions. As shown, the absence

Method	MonoDLE	GUPNet	MonoDTR	MonoDETR
Runtime↓	40	**34**	37	38
GFlops↓	79.12	62.32	120.48	**62.12**
AP_{3D} Mod.	12.26	15.02	15.39	**16.47**

Table 3. **Efficiency comparison.** We test the Runtime (ms) on one RTX 3090 GPU with batch size 1, and compare AP_{3D} on *test* set.

of the depth-guided transformer greatly hurts the performance, for the lack of non-local geometric cues. Then, we investigate two key designs within the depth-guided transformer: the transformer architecture and depth guidance. For 'w/o Transformer', we only append the depth predictor upon the center-guided baseline to provide implicit depth guidance without transformers. For 'w/o Depth Guidance', we equip the center-guided baseline with a visual encoder and decoder, but include no depth predictor, depth encoder, and the depth cross-attention layer in the decoder. This builds a transformer network guided by visual appearances, without any depth guidance for object queries. The performance degradation of both variants indicates their significance for our depth-guided feature aggregation paradigm.

Depth Encoder. The depth encoder produces non-local depth embeddings f_D^e, which are essential for queries to

Method	Image Size	NDS↑	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
CenterNet [56]	-	0.328	0.306	0.716	0.264	0.609	1.426	0.658
FCOS3D* [46]	1600×900	0.415	0.343	0.725	0.263	0.422	1.292	0.153
PGD* [47]	1600×900	0.428	0.369	0.683	0.260	0.439	1.268	0.185
DETR3D† [49]	1600×900	0.434	0.349	0.716	0.268	0.379	0.842	0.200
BEVDet† [17]	1408×512	0.417	0.349	0.637	0.269	0.490	0.914	0.268
PETR† [26]	1600×900	0.442	0.370	0.711	0.267	0.383	0.865	0.201
PETRv2 [27]	800×320	0.496	0.401	0.745	0.268	0.448	0.394	0.184
+ *Depth-gudied*		**0.508**	**0.410**	0.727	0.265	0.389	0.419	0.187
BEVFormer [22]	1600×900	0.517	0.416	0.673	0.274	0.372	0.394	0.198
+ *Depth-gudied*		**0.526**	**0.423**	0.661	0.272	0.349	0.371	0.192

Table 4. **Multi-view performance on nuScenes *val* set.** * denotes the two-step fine-tuning with test-time augmentation, and † denotes CBGS [58] training. We compare with the best-performing variants of other methods and utilize bold numbers to highlight the best results.

Architecture	Easy	Mod.	Hard
MonoDETR	**28.84**	**20.61**	**16.38**
w/o Depth-guided Trans.	19.69	15.15	13.93
w/o Transformer	20.19	16.05	14.18
w/o Depth Guidance	24.14	17.81	15.60

Table 5. **Effectiveness of depth-guided transformer.** 'Depth-guided Trans.' and 'Depth Guidance' denote the depth-guided transformer and the depth cross-attention layer, respectively

Mechanism	Easy	Mod.	Hard
Global SA	**28.84**	**20.61**	**16.38**
Deform. SA	26.43	18.91	15.55
3×3 Conv.$_{×2}$	25.55	18.36	15.28
w/o	24.25	18.38	15.41

Table 6. **The design of depth encoder.** 'Deform. SA' denotes a one-block deformable self-attention layer. 'w/o' denotes directly feeding depth features into the decoder without the depth encoder.

explore scene-level depth cues in the depth cross-attention layer. We experiment with different encoder designs in Table 6. 'Deform. SA' and '3×3 Conv.$_{×2}$' represent one-block of deformable attention and two 3×3 convolutional layers, respectively. As reported, 'Global SA' with only one block generates the best f_D^e for global gemoetry encoding.

Depth-guided Decoder. As the core depth-guided component, we explore how to better guide object queries to interact with depth embeddings f_D^e in Table 7. With the sequential inter-query self-attention ('I') and visual cross-attention ('V') layers, we insert the depth cross-attention layer ('D') into each decoder block with four positions. For '$I \rightarrow D + V$', we fuse the depth and visual embeddings f_D^e, f_V^e by element-wise addition, and apply only one unified cross-attention layer. As shown, the '$D \rightarrow I \rightarrow V$' order performs the best. By placing 'D' in the front, object queries can first aggregate depth cues to guide the remaining operations in each decoder block.

Foreground Depth Map. We explore different representations for our depth map in Table 8. Compared to dense depth supervision ('Dense'), adopting object-wise depth labels ('Fore.') can focus the network on more important foreground geometric cues, and better capture depth relations between objects. 'LID' outperforms other discretization methods, since the linear-increasing intervals can suppress the larger estimation errors of farther objects.

Depth Positional Encodings p_D. In Table 9, we experiment with different depth positional encodings for f_D^e in the depth cross-attention layer. By default, we apply the meter-wise encodings $p_D \in \mathbb{R}^{(d_{max}-d_{min}+1)\times C}$ that assign one learnable embedding per meter with depth value interpolation for output. We then assign one learnable embedding for each depth bin, denoted as 'k-bin $p_D \in \mathbb{R}^{k\times C}$', and also experiment sinusoidal functions to encode either the depth values or 2D coordinates of the feature map, denoted as 'Depth sin/cos' and '2D sin/cos', respectively. As shown, 'meter-wise p_D' performs the best for encoding more fine-grained depth cues ranging from d_{min} to d_{max}, which provides the queries with more scene-level spatial structures.

4.4. Multi-view Experiments

As a plug-and-play module for multi-view 3D object detection, we append our depth-guided transformer upon two DETR-based multi-view networks, PETR v2 [27] and BEVFormer [22]. The detailed network architectures are shown in Figure 5. For a fair comparison, we adopt the same training configurations as the two baseline models, and utilize the same $[d_{min}, d_{max}]$ with k as monocular experiments. We report the performance on nuScenes [3] *val* set in Table 4, where we apply no test-time augmentation or CGBS [58] training. For end-to-end detection in PETRv2, the depth guidance contributes to +1.2% NDS and +0.9% mAP by providing sufficient multi-view geometric cues.
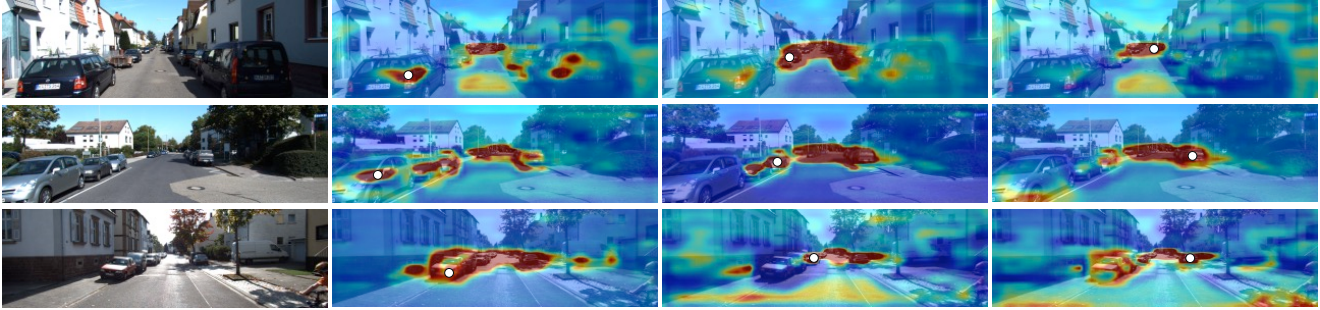
Figure 6. Visualizations of attention maps A_D in the depth cross-attention layer. The top column denotes the input image, and the last three columns denote the attention maps of the target queries (denoted as white dots). Hotter colors indicate higher attention weights.

Architecture	Easy	Mod.	Hard
$D \rightarrow I \rightarrow V$	**28.84**	**20.61**	**16.38**
$I \rightarrow D \rightarrow V$	26.24	19.28	16.03
$I \rightarrow V \rightarrow D$	25.84	18.85	15.72
$I \rightarrow D + V$	24.94	18.41	15.39

Table 7. **The design of depth-guided decoder.** 'D', 'I', and 'V' denote the depth cross-attention, inter-query self-attention, and visual cross-attention layers, respectively.

Depth Map	Easy	Mod.	Hard
Fore. LID	**28.84**	**20.61**	**16.38**
Dense LID	27.69	19.85	15.98
Fore. UD	25.61	18.90	15.49
Fore. SID	26.05	18.95	15.59

Table 8. **Different representations of the predicted depth map.** 'UD', 'SID', and 'LID' denote uniform, spacing-increasing, and linear-increasing discretizations.

For the BEV feature generation, our modules benefit BEV-Former by +0.9% NDS and +0.7% mAP, indicating the importance of auxiliary depth information for BEV-space feature encoding. The additional experiments on multi-view 3D object detection well demonstrate the effectiveness and generalizability of our approach.

5. Visualization

We visualize the attention maps A_D in Equation 3 of the depth cross-attention layer at the last decoder block. As shown in Figure 6, the areas with high attention scores for the target object query spread over the entire image, concentrating on other objects with long distances. This indicates, via our depth guidance, object queries are able to adaptively capture non-local depth cues from the image and are no longer limited by neighboring visual features.

6. Conclusion

We propose MonoDETR, an end-to-end transformer-based framework for monocular 3D object detection, which is free from any additional input, anchors, or NMS. Different from existing center-guided methods, we enable object queries to explore geometric cues adaptively from the depth-guided regions, and conduct inter-object and object-scene depth interactions via attention mechanisms. Extensive experiments have demonstrated the effectiveness of our approach for both single-view (KITTI) and multi-view (nuScenes) input. We hope MonoDETR can serve as a

Settings	Easy	Mod.	Hard
Meter-wise p_D	**28.84**	**20.61**	**16.38**
k-bin p_D	28.06	19.68	16.04
Depth sin/cos	27.42	19.57	15.82
2D sin/cos	26.48	18.63	15.52
w/o	26.76	18.94	15.85

Table 9. **The design of depth positional encodings.** 'Meter-wise' and 'k-bin' assign learnable embeddings by meters and depth bins, respectively. 'sin/cos' denotes sinusoidal functions for encodings.

strong DETR baseline for future research in monocular 3D object detection. **Limitations.** How to effectively incorporate multi-modal input into our transformer framework is not discussed in the paper. Our future direction will focus on this to further improve the performance of depth-guided transformers, e.g., distilling more sufficient geometric knowledge from LiDAR and RADAR modalities.

Acknowledgement

References

[1] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *IEEE International Conference on Computer Vision*, 2019. 1, 3

[2] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele. Kinematic 3d object detection in monocular video. In *Proceedings of the European Conference on Computer Vision*, 2020. 7

[3] 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. *CoRR*, abs/1903.11027, 2019. 2, 8

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, 2020. 2, 5, 6

[5] 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. 3

[6] Anthony Chen, Kevin Zhang, Renrui Zhang, Zihan Wang, Yuheng Lu, Yandong Guo, and Shanghang Zhang. Pimae: Point cloud and image interactive masked autoencoders for 3d object detection. *CVPR 2023*, 2023. 3

[7] Hansheng Chen, Yuyao Huang, Wei Tian, Zhong Gao, and Lu Xiong. Monorun: Monocular 3d object detection by reconstruction and uncertainty propagation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3, 7

[8] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 6

[9] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G. Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. In *Conference on Neural Information Processing Systems*, 2015. 1, 6

[10] Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li. Monopair: Monocular 3d object detection using pairwise spatial relationships. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 3, 7

[11] Zehui Chen, Zhenyu Li, Shiquan Zhang, Liangji Fang, Qinhong Jiang, and Feng Zhao. Bevdistill: Cross-modal bev distillation for multi-view 3d object detection. *arXiv preprint arXiv:2211.09386*, 2022. 3

[12] Mingyu Ding, Yuqi Huo, Hongwei Yi, Zhe Wang, Jianping Shi, Zhiwu Lu, and Ping Luo. Learning depth-guided convolutions for monocular 3d object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 3, 7

[13] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of detr with spatially modulated co-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3621–3630, 2021. 3

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 2, 6

[15] 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 (CVPR)*, June 2016. 4, 6

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

[17] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 1, 3, 8

[18] Kuan-Chih Huang, Tsung-Han Wu, Hung-Ting Su, and Winston H Hsu. Monodtr: Monocular 3d object detection with depth-aware transformer. *arXiv preprint arXiv:2203.10981*, 2022. 3, 7

[19] Peixiang Huang, Li Liu, Renrui Zhang, Song Zhang, Xinli Xu, Baichao Wang, and Guoyi Liu. Tig-bev: Multi-view bev 3d object detection via target inner-geometry learning. *arXiv preprint arXiv:2212.13979*, 2023. 3

[20] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1

[21] Peixuan Li, Huaici Zhao, Pengfei Liu, and Feidao Cao. Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving. In *European Conference on Computer Vision*, 2020. 3, 7

[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, 3, 6, 8

[23] 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

[24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1

[25] Jihao Liu, Tai Wang, Boxiao Liu, Qihang Zhang, Yu Liu, and Hongsheng Li. Towards better 3d knowledge transfer via masked image modeling for multi-view 3d understanding. *arXiv preprint arXiv:2303.11325*, 2023. 3

[26] 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, 3, 6, 8

[27] 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. 2, 3, 6, 8

[28] Zechen Liu, Zizhang Wu, and Roland Tóth. SMOKE: single-stage monocular 3d object detection via keypoint estimation. *CoRR*, abs/2002.10111, 2020. 3, 7

[29] Zongdai Liu, Dingfu Zhou, Feixiang Lu, Jin Fang, and Liangjun Zhang. Autoshape: Real-time shape-aware monocular 3d object detection. *CoRR*, abs/2108.11127, 2021. 3, 7

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 6

[31] Yan Lu, Xinzhu Ma, Lei Yang, Tianzhu Zhang, Yating Liu, Qi Chu, Junjie Yan, and Wanli Ouyang. Geometry uncertainty projection network for monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3111–3121, October 2021. 1, 3, 6, 7

[32] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking pseudo-lidar representation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 3, 7

[33] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4721–4730, June 2021. 1, 3, 6, 7

[34] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional DETR for fast training convergence. *CoRR*, abs/2108.06152, 2021. 3

[35] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

[36] 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. 3

[37] 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. 3

[38] Cody Reading, Ali Harakeh, Julia Chae, and Steven L. Waslander. Categorical depth distributionnetwork for monocular 3d object detection. *CVPR*, 2021. 1, 3, 4, 7

[39] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 3

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 1, 2

[41] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1

[42] Xuepeng Shi, Qi Ye, Xiaozhi Chen, Chuangrong Chen, Zhixiang Chen, and Tae-Kyun Kim. Geometry-based distance decomposition for monocular 3d object detection. In *IEEE International Conference on Computer Vision*, 2021. 7

[43] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. 1, 2

[44] 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. 3

[45] Li Wang, Liang Du, Xiaoqing Ye, Yanwei Fu, Guodong Guo, Xiangyang Xue, Jianfeng Feng, and Li Zhang. Depth-conditioned dynamic message propagation for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 454–463, June 2021. 3, 7

[46] 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, 3, 8

[47] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Probabilistic and geometric depth: Detecting objects in perspective. In *Conference on Robot Learning*, 2021. 3, 7, 8

[48] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1

[49] 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, 3, 8

[50] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *CVPR*, 2021. 1, 3

[51] 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. In *ICLR*, 2020. 1

[52] Yunpeng Zhang, Jiwen Lu, and Jie Zhou. Objects are different: Flexible monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3289–3298, June 2021. 1, 3, 7

[53] Yinmin Zhang, Xinzhu Ma, Shuai Yi, Jun Hou, Zhihui Wang, Wanli Ouyang, and Dan Xu. Learning geometry-guided depth via projective modeling for monocular 3d object detection. *arXiv preprint arXiv:2107.13931*, 2021. 1, 7

[54] Minghang Zheng, Peng Gao, Xiaogang Wang, Hongsheng Li, and Hao Dong. End-to-end object detection with adaptive clustering transformer. *arXiv preprint arXiv:2011.09315*, 2020. 3

[55] Dingfu Zhou, Xibin Song, Yuchao Dai, Junbo Yin, Feixiang Lu, Miao Liao, Jin Fang, and Liangjun Zhang. Iafa: Instance-aware feature aggregation for 3d object detection

from a single image. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 7

[56] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *CoRR*, abs/1904.07850, 2019. 1, 2, 8

[57] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1

[58] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *CoRR*, abs/1908.09492, 2019. 8

[59] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 3, 6