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

Renrui Zhang[*1,3], Han Qiu[*1], Tai Wang[1,3], Xuanzhuo Xu[2], Ziyu Guo[2]
Yu Qiao[1], Peng Gao[†1], Hongsheng Li[3]
[1]Shanghai AI Laboratory
[2] Peking University [3] The Chinese University of Hong Kong
{zhangrenrui, qiuhan, gaopeng}@pjlab.org.cn

Abstract

*Monocular 3D object detection has long been a challenging task in autonomous driving, which requires to decode 3D predictions solely from a single 2D image. Most existing methods follow conventional 2D object detectors to first localize objects by their centers, and then predict 3D attributes using center-neighboring local features. However, such center-based pipeline views 3D prediction as a subordinate task and lacks inter-object depth interactions with global spatial clues. In this paper, we introduce a simple framework for **Mono**cular **DE**tection with depth-aware **TR**ansformer, named **MonoDETR**. We enable the vanilla transformer to be depth-aware and enforce the whole detection process guided by depth. Specifically, we represent 3D object candidates as a set of queries and produce non-local depth embeddings of the input image by a lightweight depth predictor and an attention-based depth encoder. Then, we propose a depth-aware decoder to conduct both inter-query and query-scene depth feature communication. In this way, each object estimates its 3D attributes adaptively from the depth-informative regions on the image, not limited by center-around features. With minimal handcrafted designs, MonoDETR is an end-to-end framework **without additional data, anchors or NMS** and achieves competitive performance on KITTI benchmark among state-of-the-art center-based networks. Extensive ablation studies demonstrate the effectiveness of our approach and its potential to serve as a transformer baseline for future monocular research. Code is available at https://github.com/ZrrSkywalker/MonoDETR.git.*

1. Introduction

It is an important capability to understand the scenario and detect objects in 3D space for autonomous driving, navigation and robotics. With a wider application prospect, 3D object detection is non-trivial and much more challenging than its 2D counterpart, resulting from the complex real-world circumstances and inevitable sensor-captured noises. Compared to methods processing LiDAR-scanned point clouds [18, 33, 49] and binocular images [14, 16], 3D object detection from monocular images [2, 8, 30] receives no external inputs to prompt the depth clues and thus severely suffers from the ill-posed depth estimation, leading to a large performance drop, which remains to be further explored.

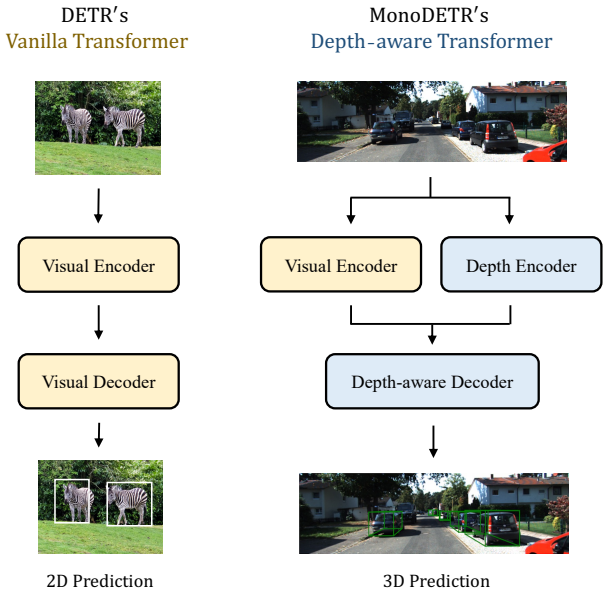

Figure 1. **DETR v.s. MonoDETR.** Based on the vanilla transformer of DETR for 2D detection, we propose the depth-aware transformer with a depth encoder and a depth-aware decoder to best adapt to the monocular domain. MonoDETR contains minimal 3D geometric priors and serves as a simple transformer baseline for monocular 3D object detection.

Following the pipelines of traditional 2D object detectors [21, 31, 37, 48], existing monocular methods [26, 40, 44, 45] first localize objects on the image by detecting their 2D

[*] Equal contribution. † Corresponding author.

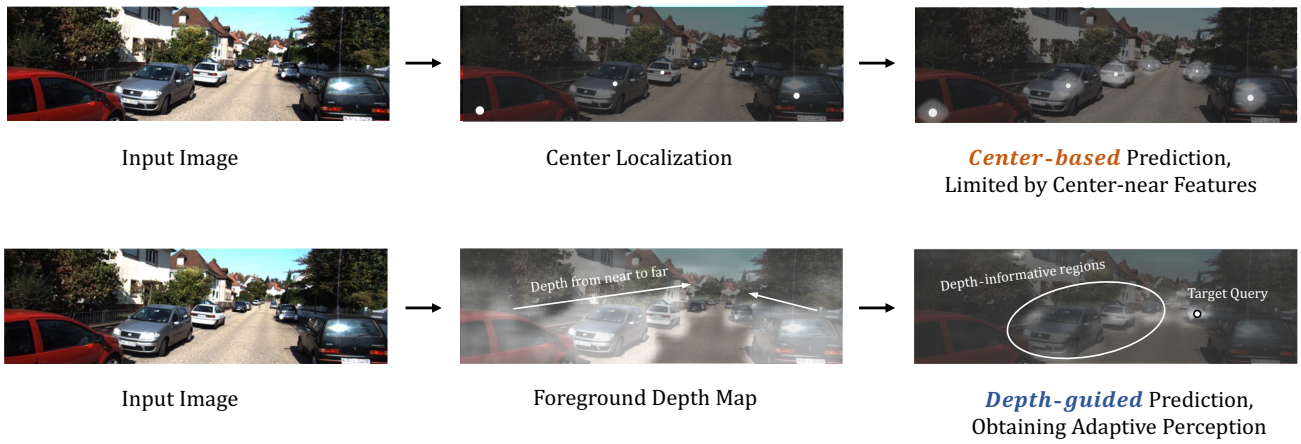

Figure 2. **Center-based Pipeline v.s. Depth-guided Pipeline.** Center-based methods first detect object centers and subsequently predict 3D attributes. Our depth-guided pipeline guides the whole process by the predicted foreground depth map and grabs features from depth-informative regions for each object. The third image of depth-guided pipeline represents the visualization of the attention map in depth cross-attention for the target query.

or 3D projected centers from a predicted heatmap. Then, the visual features nearby each object center are aggregated by convolutional kernels to predict 3D properties, e.g. depth, dimension and orientation. As shown in the top part of Figure 2, we denote such detection paradigm to be center-based, which implements center detection as a first-line task and utilizes neighboring features to estimate 3D attributes. Although it is conceptually straightforward and efficient for implementation, merely using center-near features without scene-level understanding is arduous for the network to well parse the spatial geometry, especially for depth of sparse-distributed objects. Suppose the network lacks sufficient stereo signals to judge the depth of one object, it is unable to adaptively explore long-range depth interrelations with others for assistance. Therefore, we ask the following question: can we discard the limitation based on centers and predict 3D objects guided by depth?

To tackle this issue, we rethink the pipeline for monocular 3D detection and introduce a simple transformer-based framework, **MonoDETR**, which presents a novel depth-guided paradigm, as shown in the bottom part of Figure 2. Thanks to the superiority of DETR [4] for global dependency exploration, objects in different regions could fully interact with each other by their visual features during attention mechanisms. On top of that, we further enable the vanilla transformer to be depth-aware for better adapting to the 3D monocular domain. Specifically, after the backbone extracting visual features from input images, we utilize a lightweight depth predictor to generate the depth features and predict a foreground depth map. As the depth map is produced upon depth features, the supervision for depth map implicitly endows the features with effective depth information. We supervise the depth map only contains depth values within foreground regions of different objects, as shown in Figure 3, and we utilize no extra depth labels for dense supervision. Based on the depth features, we specialize a depth encoder to produce non-local depth embeddings via self-attention, concurrent to the original encoder processing visual features in DETR [4]. On top of that, we design a depth-aware decoder, which appends a depth cross-attention into DETR's decoder to conduct depth interactions between object queries and the encoded depth embeddings. In this way, each object can adaptively aggregate features from depth-informative regions on the image with a global receptive field, and is capable of perceiving inter-object relations, as illustrated in Figure 2. The whole process of object detection is now guided by depth, other than the previous center-based manner.

To our best knowledge, MonoDETR is the first end-to-end DETR-based model in monocular 3D object detection without any extra input data, anchors or non-maximum suppression (NMS). In addition, different from recent center-based networks introducing sophisticated geometric priors to enhance the performance [24, 44, 45], MonoDETR contains minimal handcrafted designs but still performs competitively with a novel depth-guided paradigm.

We summarize the contributions of our paper as follows:

- We propose MonoDETR, the first end-to-end DETR-based detector for monocular 3D detection without extra inputs, which enables object queries to adaptively explore informative image features guided by depth.
- MonoDETR introduces minimal handcrafted designs, but achieves competitive performance with state-of-the-art center-based methods with complicated geometric priors.

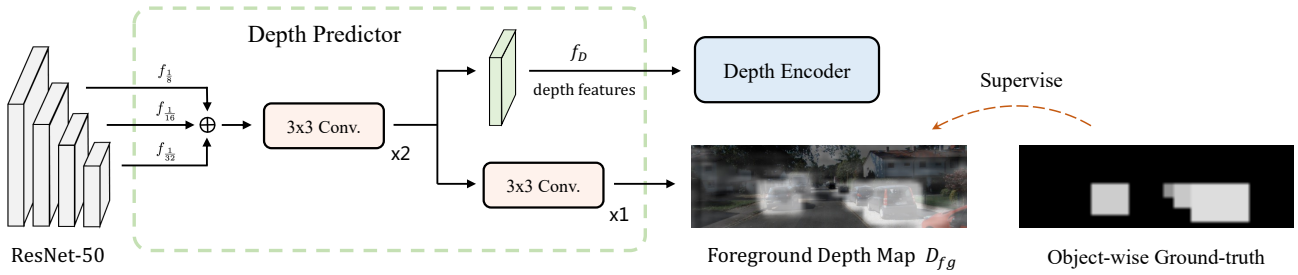

Figure 3. **The architecture of Depth Predictor.** The lightweight depth predictor within green dotted box outputs the depth features and foreground depth map of the input image. By supervising the predicted foreground depth map, we encode effective depth signals into the depth features. Note that we only utilize object-wise ground-truth without additional depth labels.

- We view MonoDETR as a simple but effective transformer baseline for future research, and conduct extensive ablation studies to demonstrate its characteristics.

2. Related Work

Monocular 3D Object Detection without Extra Data. Except for methods with additional data inputs, such as depth maps, CAD models and LiDAR data, standard monocular detectors take as input only a single image and adopt center-based pipelines following conventional 2D detectors [21, 31, 37, 48]. Deep3DBox [29] introduces discretized representation with 2D-3D prospective constraints to predict accurate object poses. M3D-RPN [2] designs a depth-aware convolution along with 3D anchors for better 3D region proposals. With very few handcrafted modules, SMOKE [22] and FCOS3D [40] propose concise architectures for one-stage monocular detection built on CenterNet [48] and FCOS [37], respectively. MonoDLE [26] and PGD [41] analyze depth errors on top and enhance their performance with customized designs. To further strengthen monocular detectors, recent methods have introduced more effective but complicated geometric priors. MonoPair [9] considers adjacent object pairs and parses their spatial relations with uncertainty. RTM3D [19] predicts 2D-3D keypoints of objects and optimizes the bounding box by nonlinear least squares. MonoFlex [44] conducts uncertainty-guided depth ensemble and categorizes different objects for distinctive processing. GUPNet [24] solves the error amplification problem by geometry-guided depth uncertainty and collocates a hierarchical learning strategy to reduce the training instability. The above geometrically dependent designs largely promote the performance of center-based methods, but the underlying problem still exists, namely, limiting 3D prediction as a subordinate task after 2D localization. On the contrary, our proposed MonoDETR adopts a depth-guided paradigm via a depth-aware transformer and discards the traditional center detection step, which extracts features from depth-informative regions on the image and contains minimal 3D-specific inductive biases.

Object Detection with Transformer. 2D object detectors [20, 21, 37] have achieved excellent performance in recent years but are equipped with cumbersome post-processing, e.g. non-maximum suppression (NMS) [1, 31]. To circumvent it, the seminal work DETR [5] constructs a novel and simple framework by adapting the powerful transformer [38] into visual detection domain. DETR detects objects on the image by an encoding-decoding paradigm and outputs set prediction via Hungary Matching Algorithm [5]. However, due to the quadratic computational complexity of attention, DETR requires 500 epochs to be fully trained, which normally takes expensive 10 days on 8 V100 GPU. To accelerate the convergence, Deformable DETR [50] designs the sparse deformable attention mechanism and achieves better performance with only 50-epoch training. ACT [46] boosts the efficiency of DETR's encoder for both time and memory by introducing adaptive clustering algorithms during inference. SMCA [13] proposes the Gaussian-modulated co-attention mechanism, which refocuses the attention of each query into object-centric areas and serves as a drop-in module to the original DETR. Besides, DETR is further enhanced by placing anchors [42], redesigning as two stages [35, 36], setting conditional attention [27], embedding dense priors [43] and so on [10, 28]. Our MonoDETR inherits DETR's superiority for non-local encoding and is free from time-consuming NMS post-processing. Based on DETR, we endow the vanilla transformer to be depth-aware by newly equipping a depth encoder and a depth-aware decoder to significantly improve the accuracy for monocular 3D detection.

3. Method

In this section, we introduce our depth-aware transformer for monocular 3D object detection, named MonoDETR. We solve the monocular detection task by set prediction and get rid of many rule-based components, such as NMS, pre-defined anchors and IoU-based label assignment. We design the pipeline to be depth-guided with minimal geometric priors, but still achieves competitive performance.

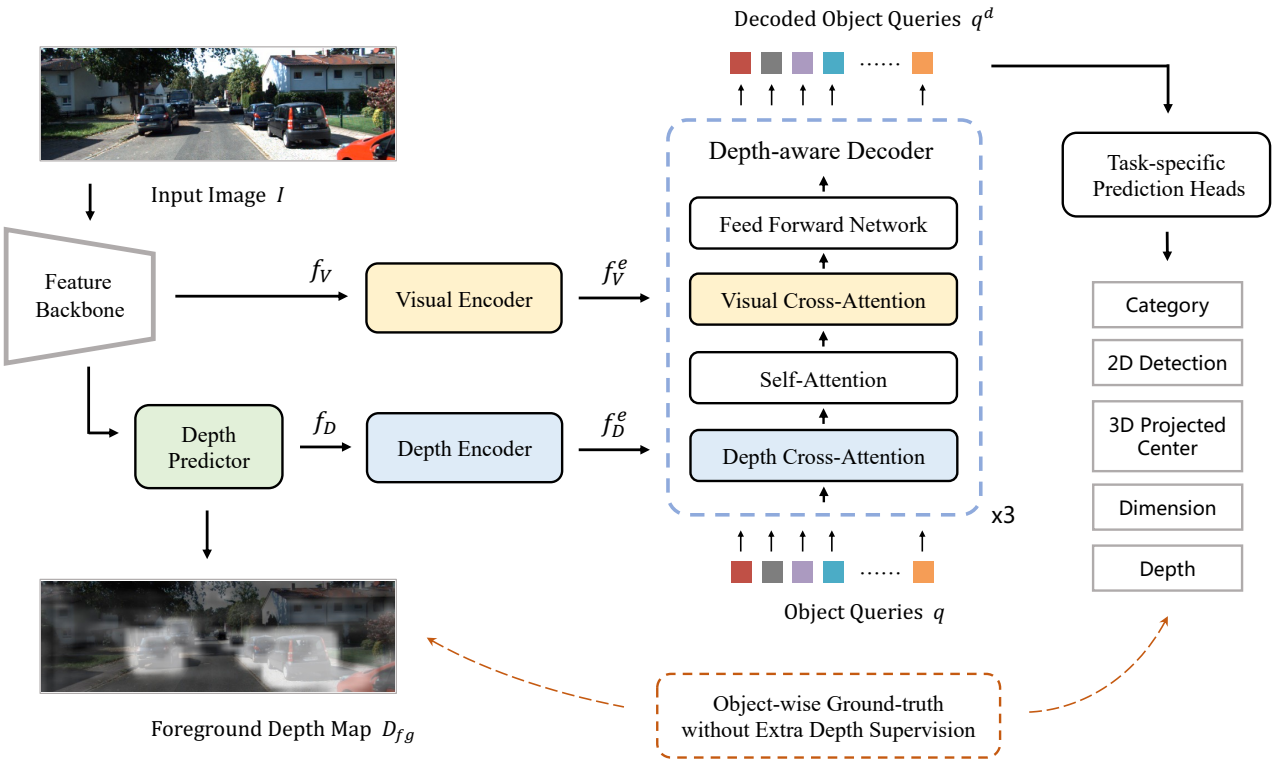

Figure 4. **The pipeline of MonoDETR.** We append the vanilla transformer in DETR with a depth predictor, a depth encoder and a depth-aware cross-attention module. The network requires no additional depth labels and only utilizes object-wise ground-truth to supervise the 3D attributes prediction and foreground depth map.

As shown in Figure 4, our MonoDETR consists of a feature backbone, a depth-aware transformer, several task-specific prediction heads and a set of learnable queries for object prediction.

3.1. Feature Backbone

The feature backbone, typically, ResNet-50 [17], takes as input a raw image and outputs both its visual and depth features, which are utilized to generate visual embeddings and depth embeddings, respectively in the later transformer.

Visual Features. Given the input image $I \in \mathbb{R}^{H\times W\times 3}$, where H and W denote its height and width, we extract the multi-scale feature maps, $f_{\frac{1}{8}}$, $f_{\frac{1}{16}}$, and $f_{\frac{1}{32}}$. They are directly acquired from the outputs of last three stages in ResNet-50 without FPN [20] and the downsample ratios are $\frac{1}{8}$, $\frac{1}{16}$ and $\frac{1}{32}$, respectively. We regard the high-level $f_{\frac{1}{32}} \in \mathbb{R}^{\frac{H}{32}\times\frac{W}{32}\times C}$ as the visual features f_V of the input image and formulate as,

$$f_{\frac{1}{8}},\ f_{\frac{1}{16}},\ f_{\frac{1}{32}} = \text{Backbone}(I), \quad f_V = f_{\frac{1}{32}}. \tag{1}$$

Depth Features. Beyond DETR for 2D detection, we require to extract the depth features from the image without extra input to guide the monocular 3D object detection. We first unify the sizes of three-level outputs into $\frac{1}{16}$ to integrate multi-scale semantics and preserve the fine-grained information. We resample the $f_{\frac{1}{32}}$ and $f_{\frac{1}{8}}$ by nearest-neighbor interpolation and utilize one convolutional projection layer for feature smoothing. Then, we aggregate all three feature maps by element addition and apply two 3×3 convolutional layers to generate the depth features for the input image, denoted as $f_D \in \mathbb{R}^{\frac{H}{16}\times\frac{W}{16}\times C}$, the dimension of which equals to visual features f_V. To supervise f_D to embed effective depth information, we generate a foreground depth map D_{fg} from f_D and compute the loss between D_{fg} and ground-truth depth labels of objects. Note that we introduce no extra depth supervision but only utilize the depth label of objects in the image to guided D_{fg}. For clearness, we sum up the aforementioned depth-related modules as a lightweight depth predictor, formulated as,

$$f_D,\ D_{fg} = \text{DepthPredictor}(f_{\frac{1}{8}}, f_{\frac{1}{16}}, f_{\frac{1}{32}}), \tag{2}$$

Foreground Depth Map. We apply a 1×1 convolutional layer on top of f_D to predict the depth map $D_{fg} \in$

$\mathbb{R}^{\frac{H}{16}\times\frac{W}{16}\times(k+1)}$. Here, we discretize $k+1$ bins for depth representation following [30], where the ordinal k bins represent foreground depth and the last one represents the background. We adopt linear-increasing discretization (LID), since the depth estimation of farther objects inherently yields larger errors, which can be suppressed with a wider categorization interval. We limit the foreground depth values within $[d_{min}, d_{max}]$, and set the first interval length and common difference both as δ. Their relationship is denoted as:

$$\delta = \frac{2(d_{max} - d_{min})}{k \cdot (k+1)}. \tag{3}$$

We define the area within 2D bounding boxes of objects as the foreground regions. For all pixels within one bounding box, we supervise them to predict the same depth value of the corresponding object. In regard to box-overlapped pixels, we give the depth supervision by the one nearer to the camera, according with the visual appearance on the image. Considering the linearly increasing intervals, we categorize a ground-truth depth value d into the k-th bin by

$$k = \lfloor -0.5 + 0.5\sqrt{1 + \frac{8(d - d_{min})}{\delta}} \rfloor. \tag{4}$$

For pixels within background area, we directly assign them to the $(k+1)$-th category. Converting the continuous value prediction into classification, we use Focal loss [21] to enforce sharp categorical distributions for depth and denote the loss as $\mathcal{L}_{dmap}$. By such instance-level depth supervision, we not only endow f_D with sufficient depth information, but also implicitly guide the backbone to understand the inter-object depth relations and to distinguish the foreground and background.

3.2. Depth-aware Transformer

The transformer in our MonoDETR is composed of a visual encoder, a depth encoder and a depth-aware decoder. The two encoders produce non-local visual and depth embeddings, respectively. The depth-aware decoder conducts feature interactions between those embeddings and object queries for thorough information interchange.

Visual and Depth Encoders. We apply three encoder blocks for visual embeddings production. Following the DETR [4], each block consists of a self-attention layer and a feed-forward neural network (FFN). The former explores the long-range dependency from all pixel pairs on the feature map, and the latter further transforms the features by a two-layer MLP. Other than the intricate visual signals, we only apply one encoder block for the depth encoder to model the scene-level depth information, which contains the same components to the block of visual encoder. The self-attention layer communicates the depth values from different foreground area and provides the network with the global clues for the entire stereo space. We utilize sine/cosine positional encodings for both encoders to supplement the absent 2D spatial structure by element-wise addition referring to [4]. We formulate the two encoders as,

$$f_V^e = \text{VisualEncoder}_{\times 3}(f_V), \tag{5}$$
$$f_D^e = \text{DepthEncoder}_{\times 1}(f_D), \tag{6}$$

where $f_V^e, f_D^e \in \mathbb{R}^{\frac{HW}{16^2}\times C}$ denote the global-encoded visual and depth embeddings.

Depth-aware Decoder. After obtaining the visual and depth embeddings, we feed them together with learnable object queries $q \in \mathbb{R}^{N\times C}$ into the depth-aware decoder, where N denotes the pre-defined maximum object number in an image. We apply three depth-aware decoder blocks, each of which consists of a depth cross-attention, a self-attention, a visual cross-attention and a FFN in sequence. During each block, the queries first explore depth information from embeddings f_D^e via depth cross-attention. To be specific, we map object queries into $Q_q \in \mathbb{R}^{N\times C}$ and the depth embeddings into $K_D, V_D \in \mathbb{R}^{\frac{HW}{16^2}\times C}$ by linear layers, which respectively serve as queries, keys and values in the attention mechanism. Then, we calculate,

$$\begin{aligned} A_D &= \text{SoftMax}(Q_q K_D^T / \sqrt{C}), \\ q' &= \text{Linear}(A_D V_D), \end{aligned} \tag{7}$$

where $A_D \in \mathbb{R}^{N\times\frac{HW}{16^2}}$ denotes the depth attention map between each query and the depth embeddings, and q' denotes the output depth-encoded queries. Thanks to this attention over depth, each object query adaptively aggregates features from depth-informative regions on the image, which makes it possible for one object to borrow effective depth signals from another and perceive the depth distribution from a global receptive field. Subsequently in order, the queries pass through the self-attention for inter-object feature interactions, the visual cross-attention for visual features exploration from f_V^e, and the FFN for non-linear transformation. After all three blocks, the queries are decoded from visual and depth embeddings, denoted as q^d, and well prepared for parallel object prediction,

$$q^d = \text{DaDecoder}_{\times 3}(q,\ f_V^e,\ f_D^e), \tag{8}$$

where $\text{DaDecoder}(\cdot,\ \cdot,\ \cdot)$ represents our depth-aware decoder.

Depth Positional Encodings. In the depth cross-attention of each decoder block, we propose learnable depth posi-

tional encodings for f_D^e, instead of using sine/cosine encodings in other attention modules. We maintain a set of learnable embeddings $p_D \in \mathbb{R}^{(d_{max}-d_{min}+1)\times C}$, where each row vector encodes the depth positional information respectively from d_{min} to d_{max}. For each pixel in f_D^e, we first estimate its depth by the predicted foreground depth map D_{fg}, and implement unidimensional interpolation from p_D to obtain the correlated depth encoding. Specifically, for pixel (x, y) on D_{fg}, we extract its categorical depth prediction, $D_{fg}(x, y) \in \mathbb{R}^{k+1}$, each channel of which denotes the depth bin probability. We represent each bin with its interval-starting depth value and assign the $(k+1)$-th bin of background with d_{max}, denoted as $\{d_{bin}^i\}_{i=1}^{k+1}$. By weighted summation, we calculate the depth of pixel (x, y) as,

$$d_{map}(x, y) = \sum_{i=1}^{k+1} D_{fg}(x, y)[i] \cdot d_{bin}^i, \tag{9}$$

where $\sum_{i=1}^{k+1} D_{fg}(x, y)[i] = 1$. Then, we utilize $d_{map}(x, y)$ to acquire the interpolated depth positional encoding of pixel (x, y) from p_D. Equipping such learnable depth positional encodings to f_D^e, object queries are facilitated to capture more sufficient knowledge during the depth cross-attention module.

3.3. Task-specific Prediction and Loss Functions

After the encoding and decoding by the depth-aware transformer, object queries are fed into a series of MLP-based heads for task-specific prediction. All queries share the head weights for the same task. We adopt one linear layer for category prediction, three-layer MLP for 3D projected center and its distances to four sides of the 2D bounding box, and two-layer MLP for depth, dimension and orientation, respectively.

Category Prediction. We detect objects of three categories in KITTI dataset [15], which are car, pedestrian and cyclist, and adopt Focal loss [21] for optimization, denoted as $\mathcal{L}_{class}$.

3D Projected Center. We directly output the normalized coordinates (x_{3D}, y_{3D}) of 3D projected centers for better accuracy, instead of the widely-adopted 2D centers with 3D-2D offsets [9, 24]. As our prediction ranges from 0 to 1, we introduce no quantization error caused by the downsampled heatmap of center-based methods, and thus discard the estimation for quantization offsets. We adopt L1 loss for 3D center localization, denoted as $\mathcal{L}_{C3D}$.

2D Detection. We recover the 2D center and size from the 3D projected center and its normalized distances to four sides of the 2D bounding box, l, r, t, b, following FCOS [37]. We apply L1 loss for predicting four distances and GIoU loss [32] for the recovered 2D bounding box, denoted as $\mathcal{L}_{lrtb}$ and $\mathcal{L}_{GIoU}$.

Dimension and Orientation. Referring to [26], we predict the 3D size without mean shapes and divide the heading angle into multiple bins with residuals. We adopt dimension-aware L1 loss [26] for 3D dimension and Multi-Bin loss [9,48] for orientation, respectively denoted as $\mathcal{L}_{dim}$ and $\mathcal{L}_{orien}$.

Depth Estimation. To produce a robust depth estimation and fully utilize the available output, we simply average three predicted depth values to form d_{pred}, which are d_{reg} from the MLP-based head, d_{geo} converted by 2D-3D size, and $d_{map}(x_{3D}, y_{3D})$ interpolated from foreground depth map D_{fg}. We formulate as,

$$d_{pred} = (d_{reg} + d_{geo} + d_{map}(x_{3D}, y_{3D}))/3. \tag{10}$$

We adopt Laplacian aleatoric uncertainty loss [9] for the overall d_{pred}, denoted as $\mathcal{L}_{depth}$.

Bipartite Matching. Different from previous methods [24, 26] with rule-based label assignment, we utilize Hungarian algorithm [4] to match the orderless queries with ground-truth labels. For training stability, we utilize 2D-related losses to calculate the matching cost. For each query-label pair, we formulate the matching cost as,

$$\begin{aligned} \mathcal{C}_{match} = \lambda_1 \cdot \mathcal{L}_{class} + \lambda_2 \cdot \mathcal{L}_{C3D} \\ + \lambda_3 \cdot \mathcal{L}_{lrtb} + \lambda_4 \cdot \mathcal{L}_{GIoU}, \end{aligned} \tag{11}$$

where we adopt the weighted values $\lambda_{1\sim4}$ as 2, 10, 5, 2, respectively.

Overall Loss. For each matched pair, we compute the loss based on the matching cost as,

$$\begin{aligned} \mathcal{L}_{pair} = \mathcal{C}_{match} + \lambda_5 \cdot \mathcal{L}_{dim} \\ + \lambda_6 \cdot \mathcal{L}_{orien} + \lambda_7 \cdot \mathcal{L}_{depth}, \end{aligned} \tag{12}$$

where $\lambda_{5\sim7}$ all equal 1. We then acquire the overall loss with all N queries for one training image as,

$$\mathcal{L}_{overall} = \frac{1}{N_{gt}} \cdot \sum_{n=1}^{N} \mathcal{L}_{pair}^n + \mathcal{L}_{dmap}, \tag{13}$$

where N_{gt} denotes the number of ground-truth labels, that is, the number of valid query-label pairs. $\mathcal{L}_{dmap}$ represents the Focal loss [21] for predicted foreground depth map illustrated in Section 3.1.

Method	Extra data	Test, $AP_{3D\|R40}$			Test, $AP_{3D\|BEV}$			Val, $AP_{3D\|R40}$		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
PatchNet [25]	Depth	15.68	11.12	10.17	22.97	16.86	14.97	-	-	-
D4LCN [11]		16.65	11.72	9.51	22.51	16.02	12.55	-	-	-
DDMP-3D [39]		19.71	12.78	9.80	28.08	17.89	13.44	-	-	-
Kinematic3D [3]	Multi-frames	19.07	12.72	9.17	26.69	17.52	13.10	19.76	14.10	10.47
MonoRUn [6]	Lidar	19.65	12.30	10.58	27.94	17.34	15.24	20.02	14.65	12.61
CaDDN [30]		19.17	13.41	11.46	27.94	18.91	17.19	23.57	16.31	13.84
AutoShape [23]	CAD	22.47	14.17	11.36	30.66	20.08	15.59	20.09	14.65	12.07
SMOKE [22]	None	14.03	9.76	7.84	20.83	14.49	12.75	14.76	12.85	11.50
MonoPair [9]		13.04	9.99	8.65	19.28	14.83	12.89	16.28	12.30	10.42
RTM3D [19]		13.61	10.09	8.18	-	-	-	19.47	16.29	15.57
PGD [41]		19.05	11.76	9.39	26.89	16.51	13.49	19.27	13.23	10.65
IAFA [47]		17.81	12.01	10.61	25.88	17.88	15.35	18.95	14.96	14.84
MonoDLE [26]		17.23	12.26	10.29	24.79	18.89	16.00	17.45	13.66	11.68
MonoRCNN [34]		18.36	12.65	10.03	25.48	18.11	14.10	16.61	13.19	10.65
MonoGeo [45]		18.85	13.81	11.52	25.86	18.99	16.19	18.45	14.48	12.87
MonoFlex [44]		19.94	13.89	12.07	28.23	19.75	16.89	23.64	17.51	14.83
GUPNet [24]		20.11	14.20	11.77	-	-	-	22.76	16.46	13.72
MonoDETR (Ours)	None	**23.65**	**15.92**	**12.99**	**32.08**	**21.44**	**17.85**	**26.66**	**20.14**	**16.88**
Improvement	*v.s. second-best*	+1.18	+1.72	+0.92	+1.42	+1.36	+0.66	+3.02	+2.63	+1.31

Table 1. **Performance of the car category on KITTI *test* and *val* set.** We utilize bold to highlight the best results, and color the second-best ones and our performance gain over them in blue.

4. Experiments

4.1. Settings

Dataset. We test MonoDETR on the widely-adopted KITTI 3D object detection benchmark [15], including 7,481 training images and 7,518 test images. We follow [7, 8] to split the training images into 3,712 sub-training set and 3,769 *val* set. We report the detection results with three-level difficulties, i.e. easy, moderate and hard, in which the moderate scores are normally for ranking. We evaluate our performance with the average precision (AP) of bounding boxes in 3D space and the bird-eye view, denoted as $AP_{3D|R40}$ and $AP_{BEV|R40}$, where $R40$ denotes the 40 recall positions. For the three categories, we set the intersection-over-union (IoU) threshold as 0.7, 0.5 and 0.5 for car, pedestrian and cyclist, respectively.

Implementation Details. We adopt ResNet-50 [17] as our feature backbone following DETR [4]. We equip both visual encoder and depth-aware decoder with three transformer blocks and set their attention mechanisms as deformable attention in [50] to save GPU memory. We only set depth encoder with one block, whose attention is the seminal attention in [4] to capture global depth features. We utilize 8 heads for all attention modules and select the maximum object number N as 50. We set the channel C and the latent feature dimension of both FFN and MLP-based heads as 256. For the foreground depth map, we set $d_{min} \sim d_{max}$ as $0m \sim 80m$ and the number of bins k as 80. On one GeForce RTX 3090 GPU, we train MonoDETR for 195 epochs with batch size 16 and the learning rate $2e^{-4}$. We adopt AdamW optimizer with weight decay $1e^{-4}$ and decrease the learning rate at 125 and 165 epochs by 10. We adopt random flip, random crop and photometric distortion for data augmentation following [26, 44, 48].

4.2. Performance Comparison

As shown in Table 1, thanks to the powerful attention mechanism and our proposed depth-aware transformer, MonoDETR achieves competitive performance for car category detection among state-of-the-art methods. For the official *test* set, we exceed all existing methods with different additional data inputs. Compared to the second-best models, MonoDETR surpasses them under easy, moderate and hard levels respectively by +1.18%, +1.72% and +0.92% in $AP_{3D|R40}$, and by +1.42%, +1.36% and +0.66% in $AP_{BEV|R40}$. For the *val* set, MonoDETR also achieves the best performance among methods without extra depth

Center -based	Geo. Priors	Deform. Attn.	Depth -guided	Easy	Mod.	Hard
✓	-	-	-	17.45	13.66	11.68
✓	✓	-	-	23.64	17.51	14.83
-	-	-	-	16.54	14.33	11.56
-	-	✓	-	23.14	16.81	14.80
-	-	✓	✓	**26.66**	**20.14**	**16.88**
Improvement by depth guidance				+3.52	+3.33	+2.08

Table 2. **Effectiveness of our depth-guided pipeline.** The first two rows represent two center-based baselines: MonoDLE [26] and MonoFlex [44]. The third row denotes DETR transferred for monocular 3D detection. In the last row, we color the performance gain by depth guidance in blue.

	Set.	Easy	Mod.	Hard
Encoder Blocks	2	22.37	15.67	13.20
	3	**23.14**	**16.81**	**14.80**
	4	21.80	16.52	13.93
Decoder Blocks	2	22.08	16.22	14.24
	3	**23.14**	**16.81**	**14.80**
	4	23.10	16.28	14.67
Channels of FFN	256	23.14	**16.81**	**14.80**
	512	21.32	15.82	13.09
	1024	**23.28**	16.38	14.39

Table 3. **Transformer settings.** FFN denotes the feed-forward neural network. The best settings for monocular 3D detection are more lightweight compared to DETR for 2D detection.

labels, surpassing the second-best by +3.02%, +2.63% and +1.31% in $AP_{3D|R40}$. More importantly, our depth-guided approach introduces much less geometric priors than those compared above, which indicates MonoDETR to serve as a simple and effective baseline for transformers in monocular 3D tasks.

4.3. Ablation Studies

We conduct extensive ablation studies of MonoDETR concerning depth guidance, transformer settings, depth-aware transformer and depth representation. We report the results of $AP_{3D|R40}$ for car category on KITTI *val* set.

Depth-guided Pipeline. We first illustrate the effectiveness of our proposed depth-guided pipeline. For center-based methods, we select the powerful MonoDLE [26] and MonoFlex [44] for comparison. The former contains minimal geometric priors and the latter is equipped with complicated handcrafted modules, represented as the first two rows of Table 2. We then construct a baseline without center-based paradigm by directly transferring the DETR model from 2D into monocular 3D object detection, since

Source	Method	Easy	Mod.	Hard
Foreground Depth Map	Sine/cosine	24.41	17.90	14.85
	Learnable p_D	26.01	19.01	15.86
Depth Features Encoding	None	23.54	18.09	15.13
	1x1 Conv.	25.65	18.37	15.25
	Depth Encoder	**26.66**	**20.14**	**16.88**
	Depth Enc.$_{\times 2}$	26.12	19.68	16.16
	Deform. Attn.	23.91	18.27	14.94

Table 4. **Depth embeddings f_D^e generation.** The two sources are producing f_D^e by foreground depth map D_{fg} and by encoding depth features f_D. Depth Enc$_{\times 2}$ and Deform. Attn. denote depth encoder with two blocks and deformable attention.

Source	Method	Easy	Mod.	Hard
Combined with Visual CA	Addition	24.97	18.42	15.35
	Concat.	25.09	18.49	15.40
Independent Depth Cross-Attention	1st Position	**26.66**	**20.14**	**16.88**
	2nd Position	26.24	19.28	16.03
	3rd Position	25.84	18.85	15.72
	Deform. Attn.	24.33	18.48	15.40
Depth Pos. Encodings	Sine/cosine	26.05	19.18	15.97
	Learnable p_D	**26.66**	**20.14**	**16.88**

Table 5. **Designs of depth-aware decoder.** Visual CA and Concat. denote visual cross-attention and concatenation. The Position represents where to place the depth cross-attention in each depth-aware decoder block. Depth Pos. Encodings represent depth positional encodings for f_D^e in depth cross-attention.

the queries in DETR simultaneously predict object centers and their 3D attributes. Based on its 2D architecture, we only append several MLP-based prediction heads for 3D tasks and adopt reference points to accelerate the convergence, represented as the third row in Table 2. However, its performance lags far behind MonoDLE [26], probably due to the low feature resolutions with downsample ratio $\frac{1}{32}$, which might filter out some faraway objects with only a few pixels on the image. Therefore, we utilize the deformable attention [50] to conduct multi-scale feature encoding, and it performs better than MonoDLE and comparable to MonoFlex. We set this variant as our baseline, which discards the traditional center-based paradigm but lacks sufficient depth clues for 3D attributed prediction. On top of that, we introduce our proposed depth-guided improvement, which contributes to great performance gain over the baseline without depth guidance. In addition, our method with minimal priors surpasses both center-based networks with a large margin, demonstrating the superiority of MonoDETR.

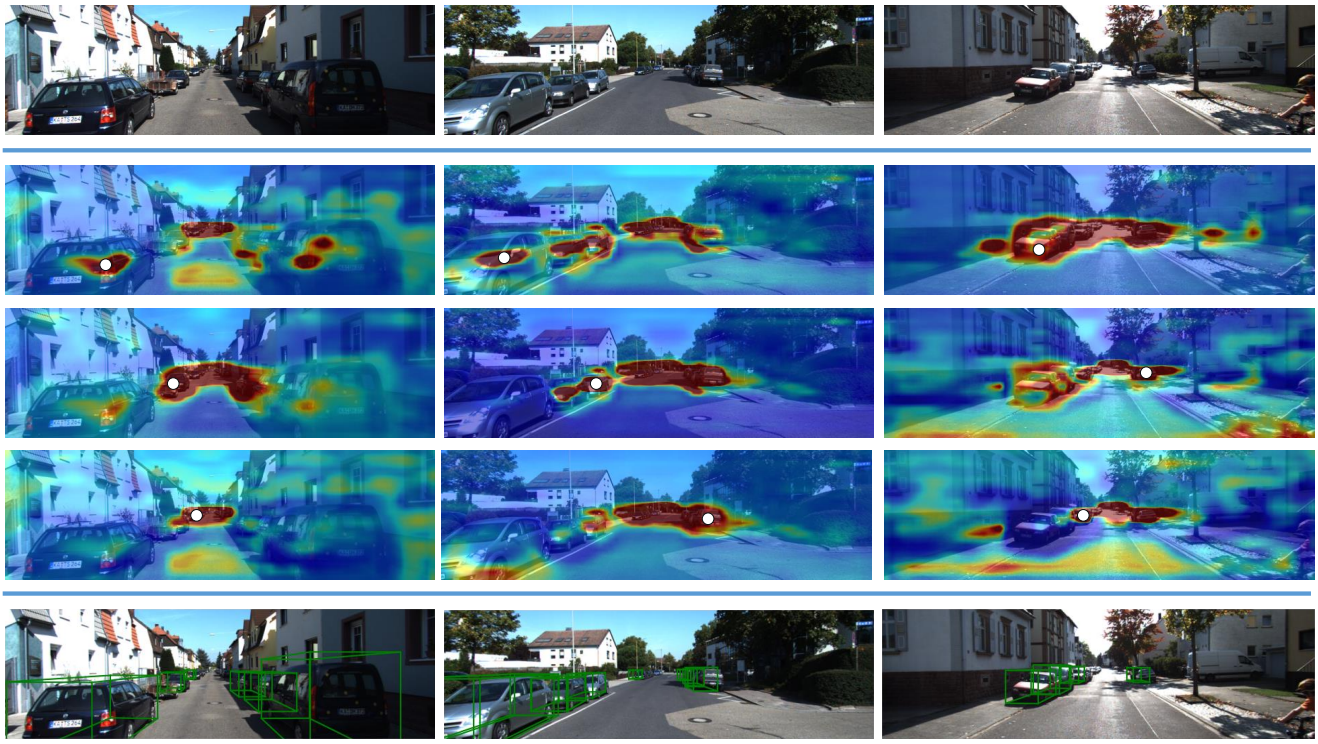

Figure 5. Visualizations of attention maps in depth cross-attention of the depth-aware decoder. The top and bottom rows locate the input images and our detection results, respectively. The middle three rows are the attention maps of the target queries colored in white.

Method	Depth Format	Easy	Mod.	Hard
Conti.	-	24.36	17.24	14.48
UD	Weighted Ave.	25.61	18.90	15.49
SID	Weighted Ave.	26.05	18.95	15.59
LID	Argmax	21.61	15.21	12.13
LID	Weighted Ave.	**26.66**	**20.14**	**16.88**

Table 6. **Depth representation settings.** Conti., UD, SID and LID indicate depth representations of continuous values, uniform, spacing-increasing and linear-increasing discretizations, respectively.

Transformer Settings. We explore the best transformer settings for monocular 3D detection concerning the numbers of encoder and decoder's blocks and the channels of feed-forward neural network (FFN). We select the variant without depth-aware transformer as the comparison baseline to better show the influence of the transformer itself. As shown in Table 3, different from DETR's [4] original settings of 6 encoder blocks, 6 decoder blocks and 1024-channel FFN, the transformer for KITTI [15] only requires 3 blocks for both encoder and decoder, and 256-channel FFN. The lighter architecture accords with the similar scenarios of street views in the dataset and we adopt such settings as default.

Depth Embeddings. Depth embeddings f_D^e are essential for queries to explore scene-level depth features during depth cross-attention in the depth-aware decoder. We experiment two ways to generate depth embeddings in Table 4. The first is to produce from the predicted foreground depth map D_{fg}, where we utilize sine/cosine functions in [4] to encode per-pixel depth values into 256-dimensional vectors or the learned depth positional encodings p_D illustrated in Section 3.2. The other processes the depth features f_D with 1×1 convolutional layer, depth encoder, and depth encoder with deformable attention. As shown, learned positional encodings interpolated by foreground depth map perform better than sine/cosine functions, since the learnable 256-dimensional vectors embeds more depth semantics ranging from d_{min} to d_{max}. Compared to 1×1 convolution and deformable attention with local receptive fields, the one-block depth encoder with global self-attention performs the best for its non-local depth interactions, which provides queries with sufficient spatial understanding.

Depth-aware Decoder. As the core component of depth-guided pipeline, we explore how to conduct depth communication between queries q and depth embeddings f_D^e. In each decoder block, we first attempt to combine f_D^e with f_V^e by addition or concatenation to enforce queries to si-

multaneously capture visual and depth features from the same cross-attention module. Then, we decouple the visual and depth decoding for queries and construct an independent depth cross-attention module, which we place at three positions: 1st before self-attention, 2nd before visual cross-attention and 3rd before FFN. In Table 5, we observe that independent depth cross-attention at the 1st position of depth-aware decoder block achieves the highest performance. In this way, each object first conducts feature interactions with depth embeddings and then passes self-attention and visual cross-attention, enabling the decoder to be better depth-guided. Also, though deformable attention works well for visual encoder and decoder, its local receptive field limits the spatial understanding in both depth encoder and depth cross-attention. On top of the best-performing settings, we test the importance of our learnable depth positional encodings p_D. As analyzed in the above paragraph, p_D contains more depth signals during training and benefits the queries a lot.

Depth Representation. Here, we experiment with different depth representations for the foreground depth map D_{fg}. We test with continuous depth vales and three discretization approches [12] for comparison, including uniform (UD), spacing-increasing (SID), and linear-increasing (LID) discretizations. For continuous depth, we directly regress the value and optimize it by L1 loss. As shown in Table 6, LID achieves the highest performance among other settings. Also, calculating the weighted average of depth bins, denoted as Weight Ave., performs much better than simply selecting the depth value with the highest probability, denoted as Argmax, since the probability distribution normally contains more semantic information.

5. Visualization

To ease the understanding of our depth-aware transformer, we visualize the attention maps of depth cross-attention in the depth-aware decoder and color the query points in white in Figure 5. As shown, the area concerned of each query spreads over pixels of other objects and the background. This indicates the query is able to borrow depth information from other informative regions under our depth-guided pipeline, which assists the prediction itself and is no more limited by center-neighboring features.

6. Conclusion

We propose MonoDETR, an end-to-end DETR-based framework for monocular 3D object detection, which contains minimal geometric designs and is free from any additional data inputs, anchors or NMS. Different from conventional center-based pipelines, we enable the prediction process to be depth-guided by designing a depth-aware transformer. By this, the queries explore depth features adaptively from depth-informative regions on the image and conducts inter-object and object-scene interactions. Extensive experiments and analysis have demonstrated the effectiveness of our MonoDETR. We hope our method could serve as a simple transformer baseline for future monocular research and motivates the community.

References

[1] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017. 3

[2] 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

[3] 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

[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, 7, 9

[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] 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. 7

[7] 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. 7

[8] 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, 7

[9] 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, 6, 7

[10] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1601–1610, 2021. 3

[11] 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 *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2020. 7

[12] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 10

[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 (ICCV)*, pages 3621–3630, October 2021. 3

[14] Ravi Garg, Vijay Kumar BG, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *Proceedings of the European Conference on Computer Vision*, 2016. 1

[15] 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. 6, 7, 9

[16] Clement Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with leftright consistency. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1

[17] 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, 7

[18] 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

[19] 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

[20] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3, 4

[21] 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, 3, 5, 6

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

[23] 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. 7

[24] 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. 2, 3, 6, 7

[25] 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. 7

[26] Xinzhu Ma, Yinmin Zhang, Dan Xu, Dongzhan Zhou, Shuai Yi, Haojie Li, and Wanli Ouyang. Delving into localization errors for monocular 3d object detection. *CoRR*, abs/2103.16237, 2021. 1, 3, 6, 7, 8

[27] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3651–3660, 2021. 3

[28] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2906–2917, 2021. 3

[29] 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

[30] Cody Reading, Ali Harakeh, Julia Chae, and Steven L. Waslander. Categorical depth distribution network for monocular 3d object detection. *CVPR*, 2021. 1, 5, 7

[31] 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*, 28:91–99, 2015. 1, 3

[32] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 6

[33] 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

[34] 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

[35] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14454–14463, 2021. 3

[36] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris M Kitani. Rethinking transformer-based set prediction for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3611–3620, 2021. 3

[37] 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, 3, 6

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3

[39] 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. 7

[40] 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 (ICCV) Workshops*, 2021. 1, 3

[41] 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

[42] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. *arXiv preprint arXiv:2109.07107*, 2021. 3

[43] Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient detr: Improving end-to-end object detector with dense prior. *arXiv preprint arXiv:2104.01318*, 2021. 3

[44] 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, 2, 3, 7, 8

[45] 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, 2, 7

[46] 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

[47] 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

[48] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *CoRR*, abs/1904.07850, 2019. 1, 3, 6, 7

[49] 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

[50] 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, 7, 8