MONODETR: DEPTH-GUIDED TRANSFORMER FOR MONOCULAR 3D OBJECT DETECTION

Anonymous authors
Paper under double-blind review

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 based on their centers, and then predict 3D attributes by neighboring features around them. However, only using such 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 a novel 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 explores object appearances, we specialize a depth encoder to produce the non-local depth embeddings for the scene-level geometric information. Then, we represent 3D object candidates as a set of queries and propose a depth-guided decoder with depth cross-attention modules, which conduct both inter-object and object-scene depth feature 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 only use local visual features. On KITTI benchmark with monocular images as input, MonoDETR achieves *state-of-the-art* and requires no extra dense depth annotations. In addition, our depth-guided transformer can also be extended to 3D object detection from multi-view images and show superior performance on nuScenes dataset. Extensive ablation studies have demonstrated the effectiveness of our approach.

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 real-world spatial circumstances. Compared to methods based on LiDAR-scanned point clouds (Zhou & Tuzel, 2018; Lang et al., 2019; Shi et al., 2019; Yin et al., 2021) and multi-view images (Wang et al., 2022; Li et al., 2022b; Liu et al., 2022a; Huang et al., 2021), 3D object detection from monocular (single-view) images (Chen et al., 2015; Brazil & Liu, 2019; Wang et al., 2021c) is of most difficulty, which generally does not rely on depth measurements or surrounding perception. The detection accuracy thus severely suffers from the ill-posed depth estimation and limited field of vision, leading to inferior performance.

Except for lifting 2D images into pseudo 3D representations (Wang et al., 2019; You et al., 2020; Ma et al., 2020; Reading et al., 2021), standard monocular 3D detection methods (Ma et al., 2021; Zhang et al., 2021b;a; Lu et al., 2021) mostly follow the pipelines of traditional 2D object detectors (Ren et al., 2015; Lin et al., 2017b; Tian et al., 2019; Zhou et al., 2019). They first localize objects by detecting their 2D or projected 3D centers on the image, and then aggregate the visual features around centers to predict the object's 3D properties, e.g., depth, 3D size, and orientation. Such center-guided methods are illustrated in Figure 1 (Top). Although it is conceptually straightforward, merely using visual features around object centers is insufficient for the network to understand the scene-level geometric structures and long-range depth dependency.

To tackle this issue, we propose MonoDETR, which presents a novel depth-guided feature aggregation scheme to adaptively estimate each object's 3D attributes based on global spatial context, as shown in Figure 1 (Bottom). The depth-guided transformer of our MonoDETR contains two

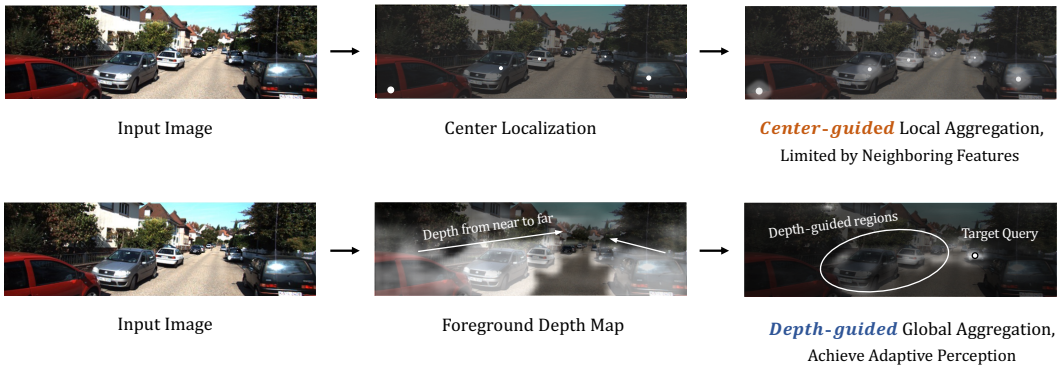

Figure 1: **Center-guided Methods (Top) and our Depth-guided Paradigms (Bottom).** Existing center-guided methods utilize visual features around the centers to predict 3D attributes of objects, while ours guides the whole process by a predicted foreground depth map and can adaptively aggregate depth features in global context. The lower right figure visualizes the attention map of the target query from the depth cross-attention layer.

parallel encoders, which are respectively for encoding the scene-level geometric depth and visual appearance information of the input image. After that, a depth-guided decoder is appended for object queries to capture non-local depth cues via attention mechanisms. Specifically, we first utilize a feature backbone along with a lightweight depth predictor to encode the visual and depth features of the input image. To endow effective depth semantics into the depth features, we predict a foreground depth map on top and supervise it only by object-wise depth labels, which requires no dense depth annotations and only contains depth values within each object's 2D bounding box. On top of that, we adopt the depth and visual encoders to respectively generate the global depth and visual embeddings via self-attention layers, which represent the input image from two perspectives, i.e., geometry and appearance. The object queries can then adaptively aggregate informative features from the two embeddings via the depth-guided decoder. Each decoder block consists of a depth cross-attention layer, an inter-query self-attention layer, and a visual cross-attention layer in order. The foremost depth cross-attention layer guides each query to adaptively capture geometric depth cues from the depth-guided regions on the image, and explore depth relations between long-distance objects. The following two attention layers further enable the object queries to interact with each other and collect the visual appearance information. By such depth-guided decoding, the prediction of 3D attributes for each object query, especially the depth, can be largely improved, which is no longer constrained by the limited visual features around centers.

As an end-to-end transformer-based network for monocular 3D object detection, MonoDETR is free from non-maximum suppression (NMS) or rule-based label assignment, and introduces minimal geometry priors between 2D and 3D. We only utilize the object-wise labels for supervision without extra input data, such as dense depth maps or LiDAR-scanned point clouds. Taking monocular images as input, MonoDETR achieves *state-of-the-art* performance among center-guided methods on KITTI (Geiger et al., 2012) benchmark.

Besides single-view 3D object detection, our MonoDETR can also be extended to object detection from multi-view images. To conduct joint perception across surrounding scenes, we respectively generate the visual and depth embeddings for each view via the shared backbone and encoders. Then, we adopt a multi-view depth-guided decoder to simultaneously aggregate appearance and geometric information from different views, which guides object queries to predict 3D attributes by surrounding scene-level depth cues. Our multi-view variant, named **MonoDETR-MV**, achieves superior performance on nuScenes (Caesar et al., 2019) benchmark compared with existing multi-view methods.

We summarize the contributions of our paper as follows:

1. We propose MonoDETR, an end-to-end transformer-based network for monocular 3D object detection, which adopts a depth-guided transformer to adaptively capture the scene-level geometric information and long-range inter-object depth relations.

2. For monocular input, MonoDETR achieves leading results on KITTI with significant gains. For multi-view input on nuScenes, our depth-guided transformer can also be extended as MonoDETR-MV with superior performance.

2 Related Work

As a low-cost solution in autonomous driving, 3D object detection from images are more economical than LiDAR, but lacks sufficient structural information for accurate 3D perception of the scene. Existing methods for image-based 3D object detection can be mainly 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 that extracts 3D properties from partial 2D signals. In contrast, multi-view detectors simultaneously encode images of surrounding scenes and 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 (Ren et al., 2015; Tian et al., 2019; Zhou et al., 2019). As early works, Deep3DBox (Mousavian et al., 2017) introduces discretized representation with 2D-3D prospective constraints, and M3D-RPN (Brazil & Liu, 2019) designs a depth-aware convolution for better 3D region proposals. With very few handcrafted modules, SMOKE (Liu et al., 2020) and FCOS3D (Wang et al., 2021c) propose concise architectures for efficient one-stage 3D detection. MonoDLE (Ma et al., 2021) and PGD (Wang et al., 2021b) analyze depth errors on top and enhance their performance with customized designs. To further supplement the limited 3D cues, various additional data are utilized for assistance, e.g., dense depth annotations (Ma et al., 2020; Ding et al., 2020; Wang et al., 2021a; Park et al., 2021), CAD models (Liu et al., 2021), videos (Brazil et al., 2020) and LiDAR (Chen et al., 2021; Reading et al., 2021; Huang et al., 2022). Also, some recent methods introduce complicated geometric priors into the networks, e.g., adjacent object pairs of MonoPair (Chen et al., 2020), optimized 2D-3D keypoints of RTM3D (Li et al., 2020), uncertainty-guided depth ensemble of MonoFlex (Zhang et al., 2021b), and geometric depth uncertainty with hierarchical learning of GUPNet (Lu et al., 2021). Despite the improvements from extra data and geometry designs, the center-guided pipelines are still limited by local visual features without scene-level spatial cues. In contrast, our MonoDETR discards the center localization 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 (Wang et al., 2022) firstly utilizes a set of 3D object queries and back-projects them onto multi-view images for feature aggregation. The queries directly detect objects within the unified 3D space and are thus free from the post-fusion across cameras. PETR (Liu et al., 2022a;b) further proposes to generate 3D position-aware features without the unstable projection and explores the advantage of temporal information from previous frames. From another point of view, BEVDet (Huang et al., 2021; Huang & Huang, 2022) follows (Philion & Fidler, 2020) to lift 2D images into a unified Bird's-Eye-View (BEV) representation and appends BEV-based heads (Yin et al., 2021) for detection. The BEV space can better indicate the spatial distributions of objects and eliminate the occlusion issue in cameras. To alleviate the sensitivity for 3D priors, BEVFormer (Li et al., 2022b) generates BEV features via a set of learnable BEV queries, and introduces a spatiotemporal transformer for visual features aggregation. Different from above methods specially designed for multi-view input, MonoDETR can process both monocular and multi-view images for 3D object detection. Our multi-view variant, MonoDETR-MV, follows DETR3D and PETR to conduct end-to-end detection without the intermediate BEV representations, and guides the 3D object queries to aggregate geometric cues adaptively from multi-view depth embeddings.

3 Method

The overall framework of MonoDETR is shown in Figure 2. We first illustrate the visual and depth features extraction in Section 3.1, and detail our depth-guided transformer for aggregating appearance and geometric cues in Section 3.2. Then in Section 3.3, we introduce how to extend MonoDETR into multi-view 3D object detection as MonoDETR-MV.

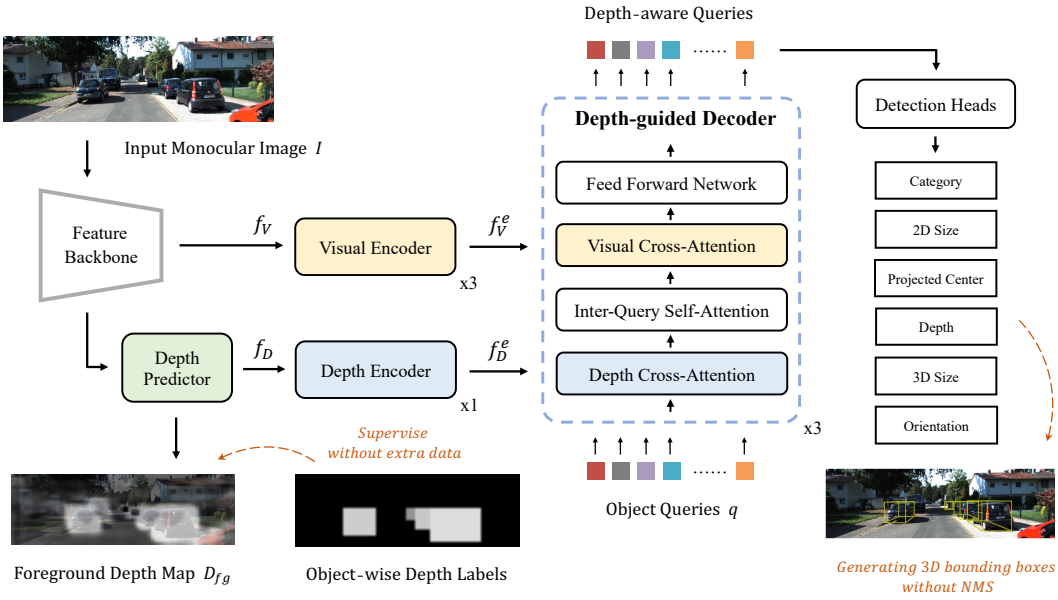

Figure 2: **Overall pipeline of MonoDETR.** We acquire the visual and depth features of the input image and utilize two parallel encoders for non-local encoding. Then, we propose a depth-guided decoder to adaptively aggregate scene-level features for object queries in global context.

3.1 Features Extraction

Taking as input a single-view image, our framework utilizes a feature backbone, e.g., ResNet-50 (He et al., 2016), 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 (Left). 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. By this, we can integrate the multi-scale visual appearances and also preserve the 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. To enforce f_D to encode informative depth information, we predict a foreground depth map D_{fg} on top of f_D and supervise it only by 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 simultaneously, we select the depth label of the object that is nearest to the camera, which accords with the visual appearance of the image.

Foreground Depth Map. To supervise the depth features f_D, we apply a 1×1 convolutional layer to predict the foreground depth map $D_{fg} \in \mathbb{R}^{\frac{H}{16}\times\frac{W}{16}\times(k+1)}$ from f_D. Here, we discretize the depth into $k+1$ bins following (Reading et al., 2021), where the first ordinal k bins represent foreground depth and the last one denotes the background. We adopt linear-increasing discretization (LID), since the depth estimation of farther objects inherently yields larger errors and 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 the common difference of LID as δ. We then categorize a ground-truth depth label d of an object into the k-th bin as

$$k = \lfloor -0.5 + 0.5\sqrt{1 + \frac{8(d - d_{min})}{\delta}} \rfloor, \quad \text{where } \delta = \frac{2(d_{max} - d_{min})}{k(k+1)}. \tag{1}$$

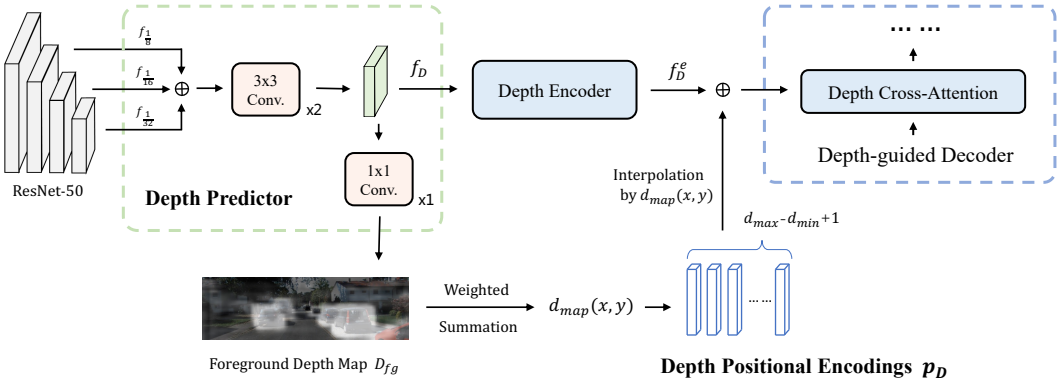

Figure 3: **Depth predictor (Left) and depth positional encodings (Right).** We utilize the depth predictor to predict the depth features and foreground depth map of the input image. In the depth cross-attention layer, we adopt learnbale depth positional encodings by depth value interpolation.

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, respectively, and the depth-guided decoder enables object queries to adaptively capture scene-level information in global context.

Visual and Depth Encoders. Given depth and visual features f_D, f_V, we specialize two transformer encoders to independently 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 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 and interacts depth features among objects with long distances. 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 visual and depth embeddings 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 consists of a depth cross-attention layer, an inter-query self-attention layer, a visual cross-attention layer and an FFN. Specifically, the queries first explore 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,

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

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 related depth features weighted by A_D to produce the depth-aware queries q', formulated as,

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

Such depth cross-attention mechanisms enable each object query to adaptively capture spatial cues from depth-guided regions on the image. Therefore, the query can better understand the scene-level spatial structures and model inter-object geometric relations to assist the 3D attribute prediction. Then, the queries pass through the inter-query self-attention layer for further feature interactions between queries. By this, one query can know what boxes would other queries are going to predict and implicitly prevent duplicate boxes. Finally, the queries are fed into visual cross-attention layer for collecting object appearance semantics from f_V^e. We stack three decoder blocks to fully fuse the scene-level depth cues into object queries and achieve the depth-guided feature aggregation scheme.

Depth Positional Encodings. In the depth cross-attention layer, we propose learnable depth positional encodings for f_D^e instead of using sinusoidal functions as others. As shown in Figure 3 (Right), 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 f_D^e, we first obtain the $(k+1)$-categorical depth prediction confidence, $D_{fg}(x, y) \in \mathbb{R}^{k+1}$, from D_{fg}, 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 denoted as $d_{map}(x, y)$. 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.

Detection Heads and Loss After the depth-guided decoder, the depth-aware object queries are fed into a series of MLP-based heads for 3D attribute prediction. During inference, we integrate the attributes to recover 3D bounding boxes without NMS post-processing. For training, we utilize Hungarian algorithm (Carion et al., 2020a) to one-to-one match the orderless queries with ground-truth object labels and compute the loss for the paired ones. We list further details in Appendix.

3.3 Multi-view 3D Object Detection

Besides monocular images, our depth-guided transformer can be extended (denoted as MonoDETR-MV) to conduct end-to-end multi-view 3D object detection. Given M-view images of a scene, $\{I_m \in \mathbb{R}^{H\times W\times 3}\}_{m=1}^M$, MonoDETR-MV shares the feature backbone, depth predictor, depth and visual encoders to concurrently extract the visual and depth embeddings $\{f_{Vm}^e, f_{Dm}^e\}_{m=1}^M$ of the M views. The multi-view foreground depth maps are also generated from $\{f_{Dm}^e\}_{m=1}^M$ and supervised by object-wise depth labels. Then, we adopt a unified multi-view depth-guided decoder that allows the object queries to simultaneously capture visual and depth information from different views, rather than independently for each view. In the depth cross-attention layer, we simply concatenate $f_{Dm}^e \in \mathbb{R}^{\frac{HW}{16^2}\times C}$ across M views to generate the multi-view depth embeddings $f_{DM}^e \in \mathbb{R}^{\frac{MHW}{16^2}\times C}$, which provide object queries with sufficient depth cues from the surrounding environments. The same concatenation for M-view visual embeddings are also used for the visual cross-attention layer. We follow existing multi-view methods (Wang et al., 2022; Liu et al., 2022a) to utilize object queries with initialized 3D reference points. After the multi-view depth-guided transformer, the depth-aware queries directly predict the attributes of objects in the unified 3D space for the M views.

4 Experiments

4.1 Settings

For monocular 3D object detection, we test MonoDETR on the widely-adopted KITTI (Geiger et al., 2012) benchmark, including 7,481 training and 7,518 test images. We follow (Chen et al., 2016) to split 3,769 training images as the *val* set. We report three-level difficulties, easy, moderate and hard, and evaluate the performance with average precision (AP) of bounding boxes in 3D space and the bird-eye view, denoted as AP_{3D} and AP_{BEV}, which are both at 40 recall positions. We report scores for the car category under IoU threshold 0.7. We illustrate the implementation details in Appendix.

4.2 Performance Comparison

As shown in Table 1, with the proposed depth-guided transformer, MonoDETR achieves *state-of-the-art* performance on KIITI *test* and *val* sets. On *test* set, MonoDETR exceeds all existing methods including those with different additional data input and surpasses the second-best under easy, moderate and hard levels respectively by +2.53%, +1.08% and +0.85% in AP_{3D}, and by +2.94%, +1.73% and +1.41% in AP_{BEV}. Therein, MonoDTR (Huang et al., 2022) also applies transformers to fuse depth features, but it is still a center-guided method without DETR's (Carion et al., 2020a) object queries for adaptive feature aggregation, and highly relies on additional dense depth supervision, anchors and NMS post-processing. In contrast, MonoDETR performs better without extra input or handcrafted designs, showing its simplicity and effectiveness for monocular 3D object detection.

Table 1: **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.

Method	Extra data	Test, AP_{3D}			Test, AP_{BEV}			Val, AP_{3D}		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
PatchNet (Ma et al., 2020)	Depth	15.68	11.12	10.17	22.97	16.86	14.97	-	-	-
D4LCN (Ding et al., 2020)		16.65	11.72	9.51	22.51	16.02	12.55	-	-	-
DDMP-3D (Wang et al., 2021a)		19.71	12.78	9.80	28.08	17.89	13.44	-	-	-
Kinematic3D (Brazil et al., 2020)	Video	19.07	12.72	9.17	26.69	17.52	13.10	19.76	14.10	10.47
MonoRUn (Chen et al., 2021)	LiDAR	19.65	12.30	10.58	27.94	17.34	15.24	20.02	14.65	12.61
CaDDN (Reading et al., 2021)		19.17	13.41	11.46	27.94	18.91	17.19	23.57	16.31	13.84
MonoDTR (Huang et al., 2022)		21.99	15.39	12.73	28.59	20.38	17.14	24.52	18.57	15.51
AutoShape (Liu et al., 2021)	CAD	22.47	14.17	11.36	30.66	20.08	15.59	20.09	14.65	12.07
SMOKE (Liu et al., 2020)	None	14.03	9.76	7.84	20.83	14.49	12.75	14.76	12.85	11.50
MonoPair (Chen et al., 2020)		13.04	9.99	8.65	19.28	14.83	12.89	16.28	12.30	10.42
RTM3D (Li et al., 2020)		13.61	10.09	8.18	-	-	-	19.47	16.29	15.57
PGD (Wang et al., 2021b)		19.05	11.76	9.39	26.89	16.51	13.49	19.27	13.23	10.65
IAFA (Zhou et al., 2020)		17.81	12.01	10.61	25.88	17.88	15.35	18.95	14.96	14.84
MonoDLE (Ma et al., 2021)		17.23	12.26	10.29	24.79	18.89	16.00	17.45	13.66	11.68
MonoRCNN (Shi et al., 2021)		18.36	12.65	10.03	25.48	18.11	14.10	16.61	13.19	10.65
MonoGeo (Zhang et al., 2021a)		18.85	13.81	11.52	25.86	18.99	16.19	18.45	14.48	12.87
MonoFlex (Zhang et al., 2021b)		19.94	13.89	12.07	28.23	19.75	16.89	23.64	17.51	14.83
GUPNet (Lu et al., 2021)		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

4.3 Ablation Studies

We report the AP_{3D} results of the car category on the KITTI *val* set for all ablation studies, which are conducted by modifying individual components of our final solution.

Depth-guided Transformer. In Table 3, we validate our proposed depth-guided transformer by removing one of its components at a time. We first construct a center-guided variant of our approach by removing the depth predictor and the entire depth guided transformer. We predict a 2D heatmap for object center detection and utilize local features to estimate their 3D attributes, denoted as 'w/o Depth-guided Trans.'. As shown, its absence greatly hurts the performance due to the limited spatial cues and non-adaptive feature aggregation. Then, we explore the effectiveness of two key designs in the depth-guided transformer: the depth guidance and the transformer architecture. For 'w/o Depth Guidance', we discard the depth cross-attention layer in the decoder, which builds a vanilla transformer network without any depth guidance for object queries. For 'w/o Transformer', we remove the transformer architecture but preserves the depth predictor to provide implicit depth guidance to the network. The performance degradation indicate the significance of both depth-guided feature aggregation schemes and transformer's scene-level encoding. We also show the superiority of learnable depth positional encodings in the depth cross-attention layer via 'w/o Depth Pos. p_D'.

Depth Encoder. The depth encoder produces non-local depth embeddings f_D^e, which are essential for queries to explore scene-level depth cues in the depth cross-attention layer. We experiment with different encoder designs in Table 4. 'Global SA($_{\times 2}$)' denotes one or two blocks of vanilla self-attention layer with a global receptive field. 'Deform. SA' and '3×3 Conv.$_{\times 2}$' represent one-block of deformable attention and two 3×3 convolutional layers, respectively. As reported, the global geometry understanding via 'Global SA' with only one block generates the best f_D^e for MonoDETR.

Depth-guided Decoder. As the core component of our depth-guided paradigm, we explore how to better guide object queries q to aggregate depth features from f_D^e in Table 5. With the sequential inter-query self-attention and visual cross-attention layers, we experiment four positions to integrate the depth cross-attention layer into each decoder block. We denote the three attention modules respectively as 'I', 'V' and 'D'. 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. The '$D \rightarrow I \rightarrow V$' order achieves the highest performance. By placing 'D' in the front, object queries could first aggregate geometric depth cues to guide the remaining operations in each decoder block, benefiting the whole 3D detection to be better depth-guided.

Table 2: **Multi-view performance on nuScenes *val* set.** * denotes the two-step fine-tuning with test-time augmentation, and † denotes training with CBGS (Zhu et al., 2019). We compare with the best-performing variants of other methods and utilize bold numbers to highlight the best results.

Method	Image Size	NDS↑	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
CenterNet (Zhou et al., 2019)	-	0.328	0.306	0.716	0.264	0.609	1.426	0.658
FCOS3D* (Wang et al., 2021c)	1600×900	0.415	0.343	0.725	0.263	0.422	1.292	0.153
PGD* (Wang et al., 2021b)	1600×900	0.428	0.369	0.683	0.260	0.439	1.268	0.185
DETR3D† (Wang et al., 2022)	1600×900	0.434	0.349	0.716	0.268	0.379	0.842	0.200
PETR† (Liu et al., 2022a)	1600×900	0.442	0.370	0.711	0.267	0.383	0.865	0.201
PETRv2 (Liu et al., 2022b)	800×320	0.496	0.401	0.745	0.268	0.448	0.394	0.184
BEVDet† (Huang et al., 2021)	1408×512	0.417	0.349	0.637	0.269	0.490	0.914	0.268
BEVDet4D† (Huang & Huang, 2022)	1600×640	0.515	0.396	0.619	0.260	0.361	0.399	0.189
BEVFormer (Li et al., 2022b)	1600×900	0.517	0.416	0.673	0.274	0.372	0.394	0.198
MonoDETR-MV (Ours)	800×320	0.508	0.410	0.727	0.265	0.389	0.419	0.187
	1600×640	**0.524**	**0.435**	0.709	0.268	0.382	0.393	0.187

Table 3: **Effectiveness of depth-guided transformer.** 'Depth-guided Trans.', 'Depth Guidance', 'Depth Pos.' denote depth-guided transformer, depth cross-attention layer and depth positional encodings.

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
w/o Depth Pos. p_D	24.04	18.11	15.10

Table 4: **The design of depth encoder.** 'Global SA$_{\times 2}$' and 'Deform. SA' denote two-block vanilla self-attention and one-block deformable self-attention layers.

Mechanism	Easy	Mod.	Hard
Global SA	**28.84**	**20.61**	**16.38**
Global SA$_{\times 2}$	26.57	19.44	16.02
Deform. SA	26.43	18.91	15.55
3×3 Conv.$_{\times 2}$	25.55	18.36	15.28
w/o	24.25	18.38	15.41

Depth Positional Encodings p_D. In Table 6, we experiment 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

Settings. For multi-view 3D object detection, we test our multi-view extension MonoDETR-MV on nuScenes (Caesar et al., 2019) benchmark, which consists of 1,000 scenes with 700, 150, 150 for training, validation, and testing, respectively. Each scene contains a 20-second video from 6 cameras with 360° horizontal FOV and is annotated with 3D bounding boxes every 0.5 seconds of 10 categories. We report the seven metrics for evaluation: nuScenes Detection Score (NDS), mean Average Precision (mAP), mean Average Translation Error (mATE), mean Average Scale Error (mASE), mean Average Orientation Error (mAOE), mean Average Velocity Error (mAVE), and mean Average Attribute Error (mAAE). We illustrate the implementation details in Appendix.

Performance Comparison. As shown in Table 2, our MonoDETR-MV achieves the highest NDS and mAP among existing multi-view methods on nuScenes *val* set. Note that we do not use the two-step fine-tuning, test-time augmentation, or CBGS (Zhu et al., 2019) training. Compared to other end-to-end methods, MonoDETR-MV of 800×320 image size surpasses PETRv2 (Liu et al., 2022b) by +1.2% NDS, +0.9% mAP, and also exceeds PETR (Liu et al., 2022a) and DETR3D (Wang et al., 2022) by large margins. Compared to BEV-based methods, we achieve better results than BEVFormer (Li et al., 2022b) by +0.7% NDS and +1.9% mAP. The multi-view experiments demonstrate the superiority of our depth-guided transformer for general image-based 3D object detection.

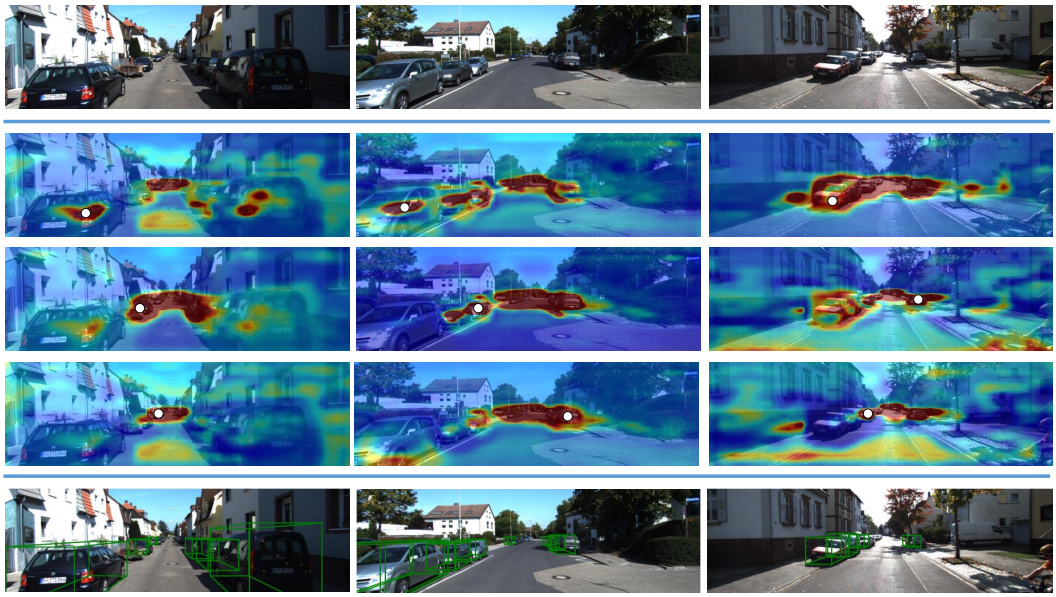

Figure 4: Visualizations of attention maps A_D in the depth cross-attention layer. The top and bottom rows represent the input images and detection results. The middle three rows are the attention maps of the target queries (denoted as white dots). Hotter colors indicate higher attention weights.

Table 5: **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.

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 6: **Depth positional encodings p_D.** 'Meter-wise' and 'k-bin' assign learnable embeddings by meters and depth bins. 'sin/cos' denotes sinusoidal functions.

Settings	Easy	Mod.	Hard
Meter-wise p_D	**28.84**	**20.61**	**16.38**
k-bin p_D	24.58	18.33	15.23
Depth sin/cos	26.05	19.18	15.97
2D sin/cos	24.65	18.11	15.16

5 Visualization

We visualize the attention maps A_D in Eq. 3 of the depth cross-attention layer at the last depth-guided decoder block. As shown in Figure 4, the areas with high attention values for the target object query spread over the entire image, which focus on other objects with long distances. This indicates, via our depth-guided transformer, 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 conducts inter-object and object-scene depth feature interactions. Extensive experiments and analyses have demonstrated the effectiveness of our approach for both single-view and multi-view input. **Limitations.** How to effectively incorporate the 3D geometric priors into our transformer framework is not discussed in the paper. Our future direction will focus on this to further improve the performance for image-based 3D object detection. **Societal Impact.** We do not foresee negative social impact in this work.

REFERENCES

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

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. 3, 7

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. URL http://arxiv.org/abs/1903.11027. 2, 8, 17

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*, 2020a. 6, 14, 15, 16, 17

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*, pp. 213–229. Springer, 2020b. 14

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

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

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

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, 14, 15, 16

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*, pp. 1601–1610, 2021. 14

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

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)*, pp. 3621–3630, October 2021. 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*, pp. 3354–3361, 2012. doi: 10.1109/CVPR.2012.6248074. 2, 6, 14, 16, 17

Ross Girshick. Fast r-cnn. In *IEEE International Conference on Computer Vision*, 2015. 14

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, 17

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

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

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, 6, 7, 15, 16

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

Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13906–13915, 2020. 17

Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13619–13627, 2022a. 14

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, 15

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*, 2022b. 1, 3, 8, 14

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014. 16

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*, 2017a. 14

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*, 2017b. 1, 14, 15

Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. *arXiv preprint arXiv:2203.05625*, 2022a. 1, 3, 6, 8, 14, 17

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*, 2022b. 3, 8, 17

Zechen Liu, Zizhang Wu, and Roland Tóth. SMOKE: single-stage monocular 3d object detection via keypoint estimation. *CoRR*, abs/2002.10111, 2020. URL https://arxiv.org/abs/2002.10111. 3, 7, 15

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. URL https://arxiv.org/abs/2108.11127. 3, 7

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 17

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)*, pp. 3111–3121, October 2021. 1, 3, 7, 15

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

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)*, pp. 4721–4730, June 2021. 1, 3, 7, 14, 15, 16, 17

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*, pp. 3651–3660, 2021. 14

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*, pp. 2906–2917, 2021. 14

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

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*, pp. 3142–3152, 2021. 3

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*, pp. 194–210. Springer, 2020. 3

Cody Reading, Ali Harakeh, Julia Chae, and Steven L. Waslander. Categorical depth distribution-network for monocular 3d object detection. *CVPR*, 2021. 1, 3, 4, 7

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, 3, 14

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*, pp. 658–666, 2019. 14

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

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, 15

Andrea Simonelli, Samuel Rota Bulò, Lorenzo Porzi, Elisa Ricci, and Peter Kontschieder. Towards generalization across depth for monocular 3d object detection. In *Proceedings of the European Conference on Computer Vision*, 2020. 16

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*, pp. 14454–14463, 2021a. 14

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*, pp. 3611–3620, 2021b. 14

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*, pp. 9627–9636, 2019. 1, 3, 14

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. 14

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)*, pp. 454–463, June 2021a. 3, 7

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

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*, pp. 913–922, 2021c. 1, 3, 8

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

Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. *arXiv preprint arXiv:2109.07107*, 2021d. 14

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*, pp. 180–191. PMLR, 2022. 1, 3, 6, 8, 14

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. 14

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

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

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*, 2021a. 1, 7, 15, 16

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)*, pp. 3289–3298, June 2021b. 1, 3, 7, 16, 17

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. 14

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, 15

Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *CoRR*, abs/1904.07850, 2019. URL https://arxiv.org/abs/1904.07850. 1, 3, 8, 14, 17

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

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. URL http://arxiv.org/abs/1908.09492. 8

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. 14, 17

A APPENDIX

A.1 ADDITIONAL RELATED WORK

Object Detection with Transformers. 2D object detectors (Girshick, 2015; Ren et al., 2015; Lin et al., 2017a;b; Tian et al., 2019) have achieved excellent performance in recent years, but count on cumbersome non-maximum suppression (NMS) post-processing and rule-based label assignment. To circumvent it, the seminal work DETR (Carion et al., 2020b) constructs a novel framework by adapting the powerful transformer (Vaswani et al., 2017) in natural language processing into computer vision for 2D detection. DETR detects objects on the image by an encoder-decoder architecture and conducts set prediction aided by Hungary Matching Algorithm (Carion et al., 2020b). However, due to the quadratic computational complexity of attention, DETR requires the expensive 500 epochs to be fully trained. To accelerate the convergence, Deformable DETR (Zhu et al., 2020) designs sparse deformable attention mechanisms and achieves better performance with only 50-epoch training. ACT (Zheng et al., 2020) boosts the time efficiency by introducing adaptive clustering algorithms during inference. SMCA (Gao et al., 2021) proposes Gaussian-modulated co-attention mechanisms that refocus the attention of each query into object-centric areas. Besides, DETR is further enhanced by placing anchors (Wang et al., 2021d), redesigning as two stages (Sun et al., 2021a;b), setting conditional attention (Meng et al., 2021), embedding dense priors (Yao et al., 2021), introducing query denoising (Li et al., 2022a) and so on (Dai et al., 2021; Misra et al., 2021). For image-based 3D object detection, DETR3D (Wang et al., 2022) and PETR (Liu et al., 2022a) adopt vanilla transformers with 3D object queries to aggregate surrounding visual features in an end-to-end way. BEVFormer (Li et al., 2022b) utilizes a spatiotemporal transformer to generate BEV representations from multi-view images. In contrast, our MonoDETR equip the vanilla transformer with depth guidance for adaptive scene-level depth understanding, and can tackle both single-view and multi-view circumstances.

A.2 DETAILS OF ATTRIBUTE PREDICTION AND LOSS FUNCTIONS

After the depth-guided transformer, we adopt detection heads to estimate six attributes for each object query: object category, 2D size (l, r, t, b), projected 3D center (x_{3D}, y_{3D}), depth d_{reg}, 3D size (h_{3D}, w_{3D}, l_{3D}) and orientation α. All queries share the head weights for the same attribute. Specifically, we utilize one linear projection layer for the object category, and two-layer MLP for depth, 3D size and orientation, and three-layer MLP for 2D size and projected 3D center.

Projected 3D Center (x_{3D}, y_{3D}). We directly output the coordinate (x_{3D}, y_{3D}) of each query's projected 3D center on the image, which thus discards two types of widely-adopted offsets. The first is the 2D-to-3D offset for recovering the projected 3D center from the predicted 2D center. The other is the quantization offset caused by the downsampled heatmap, which is a requisite for existing center-guided methods. By this, we can obtain the projected 3D center of each object in one step without the error of intermediate offsets, contributing to better localization accuracy. We adopt L1 loss for the center estimation and denote it as $\mathcal{L}_{xy3D}$.

Object Category and 2D Size (l, r, t, b). We detect objects of three categories, car, pedestrian and cyclist, in KITTI (Geiger et al., 2012), and adopt Focal loss (Lin et al., 2017b) for optimization, denoted as $\mathcal{L}_{class}$. Referring to FCOS (Tian et al., 2019), we obtain the 2D bounding box of an object by predicting the distances from its four sides, l, r, t, b, to the projected 3D center (x_{3D}, y_{3D}). Both (l, r, t, b) and (x_{3D}, y_{3D}) are normalized from 0 to 1 by the image size. We apply L1 loss for the distances and GIoU loss (Rezatofighi et al., 2019) for the recovered 2D bounding box following DETR (Carion et al., 2020a), denoted as $\mathcal{L}_{lrtb}$ and $\mathcal{L}_{GIoU}$, respectively.

3D Size (h_{3D}, w_{3D}, l_{3D}) and Orientation α. Instead of predicting the residuals to the mean shape values, we follow MonoDLE (Ma et al., 2021) to use the 3D IoU oriented loss for 3D sizes. We divide the heading angle into multiple bins with residuals and adopt MultiBin loss (Chen et al., 2020; Zhou et al., 2019) to optimize the prediction of orientation. The two losses are respectively denoted as $\mathcal{L}_{size3D}$ and $\mathcal{L}_{orien}$.

Table 7: **Performance of the car category on KITTI *val* sets under different IoU thresholds.** We utilize bold numbers to highlight the best results, and blue for the second-best ones.

Method	AP_{BEV}@IoU=0.7			AP_{3D}@IoU=0.5			AP_{BEV}@IoU=0.5		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
SMOKE (Liu et al., 2020)	19.99	15.61	15.28	-	-	-	-	-	-
MonoPair (Chen et al., 2020)	24.12	18.17	15.76	55.38	42.39	37.99	61.06	47.63	41.92
MonoRCNN (Shi et al., 2021)	25.29	19.22	15.30	-	-	-	-	-	-
MonoDLE (Ma et al., 2021)	24.97	19.33	17.01	55.41	43.42	37.81	60.73	46.87	41.89
IAFA (Zhou et al., 2020)	22.75	19.60	19.21	-	-	-	-	-	-
MonoGeo (Zhang et al., 2021a)	27.15	21.17	18.35	56.59	43.70	39.37	61.96	47.84	43.10
RTM3D (Li et al., 2020)	24.74	22.03	18.05	52.59	40.96	34.95	56.90	44.69	41.75
GUPNet (Lu et al., 2021)	31.07	22.94	19.75	57.62	42.33	37.59	61.78	47.06	40.88
MonoDTR (Huang et al., 2022)	33.33	25.35	21.68	64.03	47.32	42.20	69.04	52.47	45.90
MonoDETR (Ours)	**37.86**	**26.95**	**22.80**	**68.86**	**48.92**	**43.57**	**72.30**	**53.10**	**46.62**
Improvement	+4.53	+1.60	+1.12	+4.83	+1.60	+1.37	+3.26	+0.63	+0.72

Depth d_{pred}. To estimate the final object depth d_{pred} more accurately, we average three predicted values: d_{reg} regressed by the detection head, d_{geo} converted by the predicted 2D and 3D sizes, and $d_{map}(x_{3D}, y_{3D})$ interpolated from D_{fg}. We formulate as

$$d_{geo} = f\,\frac{h_{3D}}{t+b}, \qquad d_{pred} = (d_{reg} + d_{geo} + d_{map}(x_{3D}, y_{3D}))/3, \tag{4}$$

where h_{3D} and $t+b$ denote the predicted heights of 3D and 2D sizes, and f denotes the camera focal length. We then adopt Laplacian aleatoric uncertainty loss (Chen et al., 2020) for the overall d_{pred}, formulated as

$$\mathcal{L}_{depth} = \frac{\sqrt{2}}{\sigma}\|d_{gt} - d_{pred}\|_1 + \log(\sigma), \tag{5}$$

where σ denotes the standard deviation predicted together with d_{reg}, and d_{gt} denotes the ground-truth depth label of the object.

Bipartite Matching. To correctly match each query with a ground-truth object, we calculate the loss for each query-label pair and utilize Hungarian algorithm (Carion et al., 2020a) to find the globally optimal matching. For each pair, we integrate the losses of six attributes into two groups. The first group contains object category, 2D size and the projected 3D center, since these attributes mainly concern 2D visual appearances of the image, formulated as

$$\mathcal{L}_{2D} = \lambda_1\mathcal{L}_{class} + \lambda_2\mathcal{L}_{xy3D} + \lambda_3\mathcal{L}_{lrtb} + \lambda_4\mathcal{L}_{GIoU}, \tag{6}$$

where we set $\lambda_{1\sim4}$ as 2, 10, 5, 2, respectively. The second group consists of the depth, 3D size and orientation, which are 3D spatial properties of the object, formylated as

$$\mathcal{L}_{3D} = \mathcal{L}_{size3D} + \mathcal{L}_{orien} + \mathcal{L}_{depth}. \tag{7}$$

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

$$\mathcal{L}_{overall} = \frac{1}{N_{gt}} \cdot \sum_{n=1}^{N_{gt}} (\mathcal{L}_{2D} + \mathcal{L}_{3D}) + \mathcal{L}_{dmap}. \tag{8}$$

$\mathcal{L}_{dmap}$ represents the loss of the predicted categorical foreground depth map D_{fg}, for which we also utilize Focal loss (Lin et al., 2017b).

Table 8: **Performance of the pedestrian and cyclist categories on KITTI *test* set.** We utilize bold numbers to highlight the best results, and blue ones for the second-best ones.

Method	Pedestrian, AP_{3D}			Cyclist, AP_{3D}		
	Easy	Mod.	Hard	Easy	Mod.	Hard
Movi3D (Simonelli et al., 2020)	8.99	5.44	4.57	1.08	0.63	0.70
MonoGeo (Zhang et al., 2021a)	8.00	5.63	4.71	4.73	2.93	2.58
MonoFlex (Zhang et al., 2021b)	9.43	6.31	5.26	4.17	2.35	2.04
MonoDLE (Ma et al., 2021)	9.64	6.55	5.44	4.59	2.66	2.45
MonoPair (Chen et al., 2020)	10.02	6.68	5.53	3.79	2.12	1.83
MonoDETR (Ours)	**12.54**	**7.89**	**6.65**	**7.33**	**4.18**	**2.92**
Improvement	+2.52	+1.21	+1.12	+2.60	+1.25	+0.34

A.3 Additional Results

Car Category on KITTI *val* Set. We list more results of the car category on KITTI *val* set under different IoU thresholds in Table 7, where our MonoDETR all achieves the highest detection accuracy. Compared to the second-best MonoDTR (Huang et al., 2022) that is a center-guided method with external depth supervision, our MonoDETR only requires object-wise depth labels and surpasses it by significant gains for the easy level, e.g., +4.53% AP_{BEV}@IoU=0.7 and +4.83% AP_{3D}@IoU=0.5.

Pedestrian and Cyclist Categories. In Table 8, we report the scores for pedestrian and cyclist categories on KITTI *test* set both under the IoU threshold of 0.5. As these two categories contain much fewer training samples than car, it is more challenging for the network to accurately detect them. As shown, MonoDETR achieves superior AP_{3D} to other methods without additional data, indicating our superior generalization ability on other categories.

A.4 Additional Ablation Study

Depth Discretization. We explore different depth discretization methods for the foreground depth map d_{fg} in Table 9. 'UD', 'SID' and 'LID' denote uniform, spacing-increasing, and linear-increasing discretizations, respectively. Instead of the weighted summation of depth bins, 'LID + argmax' outputs the depth value of the most confident bin. For 'Continuous Rep.', we directly regress the continuous depth value and optimize it by L1 loss. As reported, 'LID' performs the best than other discretization methods, since the linear-increasing intervals can suppress the larger estimation errors of farther objects. Also, 'LID' with weighted summation outperforms 'LID + argmax' for aggregating more depth cues from the predicted confidence of other depth bins.

Bipartite Matching. Our best solution only utilizes $\mathcal{L}_{2D}$ as the matching cost for each query-label pair. We investigate how it performs to append more 3D losses into the matching cost. As reported in Table 10, adding $\mathcal{L}_{size3D}$ or $\mathcal{L}_{orien}$ would adversely influence the performance due to their unstable prediction during training. Further, adding $\mathcal{L}_{depth}$ or the whole $\mathcal{L}_{3D}$ even leads to training collapse, which is caused by the ill-posed depth estimation from monocular images.

Transformer Blocks and FFN Channels. In Table 11, we experiment different block numbers of the visual encoder and depth-guided decoder, along with the latent channels of feed-forward neural network (FFN). As reported, MonoDETR achieves the best performance with the 3-block visual encoder, 3-block depth-guided decoder, and 256-channel FFN. Different from DETR's (Carion et al., 2020a) 6-block encoder, 6-block decoder, and 1024-channel FFN for COCO (Lin et al., 2014) dataset, MonoDETR adopts a lighter-weight transformer architecture because of the limited training samples in KITTI (Geiger et al., 2012) dataset.

Table 9: **The design of depth discretization in the foreground depth map.** 'Continuous Rep.' denotes the continuous representation of depth.

Settings	Easy	Mod.	Hard
LID	**28.84**	**20.61**	**16.38**
UD	25.61	18.90	15.49
SID	26.05	18.95	15.59
LID + argmax	21.61	15.21	12.13
Continuous Rep.	24.36	17.24	14.48

Table 10: **The design of bipartite matching.** 'w' denotes adding the loss to the matching cost. '-' denotes training collapse.

Matching Cost	Easy	Mod.	Hard
$\mathcal{L}_{2D}$	**28.84**	**20.61**	**16.38**
w $\mathcal{L}_{size3D}$	27.13	19.21	15.93
w $\mathcal{L}_{orien}$	25.78	18.63	15.12
w $\mathcal{L}_{depth}$	-	-	-
w $\mathcal{L}_{3D}$	-	-	-

Table 11: **Transformer blocks and FFN channels.** FFN denotes the feed-forward neural network.

	Set.	Easy	Mod.	Hard
Visual Encoder Blocks	2	26.72	18.73	15.43
	3	**28.84**	**20.61**	**16.38**
	4	27.37	20.04	16.09
Depth-guided Decoder Blocks	2	25.55	18.58	15.41
	3	**28.84**	**20.61**	**16.38**
	4	25.31	18.29	15.11
FFN Channels	256	**28.84**	**20.61**	**16.38**
	512	27.24	18.93	15.54
	1024	26.77	19.07	15.87

A.5 Implementation Details

Monocular Experiments on KITTI (Geiger et al., 2012). We adopt ResNet-50 (He et al., 2016) as our feature backbone. To save GPU memory, we apply deformable attention mechanisms (Zhu et al., 2020) for the visual encoder and visual cross-attention layers, and utilize the vanilla attention (Carion et al., 2020a) to better capture global spatial structures 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. We set the channel C and all MLP's latent feature dimension 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 GeForce RTX 3090 GPU, we train MonoDETR for 195 epochs with batch size 16 and the learning rate 2×10^{-4}. We adopt AdamW (Loshchilov & Hutter, 2018) optimizer with weight decay 10^{-4} and decrease the learning rate at 125 and 165 epochs by 0.1. For data augmentation on KITTI *test* set, we adopt random flip and photometric distortion following previous works (Zhang et al., 2021b; Ma et al., 2021; Zhou et al., 2019), but for the *val* set, we also use random crop to further boost the performance. 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.

Multi-view Experiments on nuScenes (Caesar et al., 2019). For fair comparison with existing multi-view methods, MonoDETR-MV follows most of the settings in (Liu et al., 2022a;b), including VoVNetV2 (Lee & Park, 2020) feature backbone, 3D object queries, 3D position embeddings, temporal information, loss functions and data augmentation. We utilize 2 blocks for the depth encoder to better encode multi-view depth embeddings, and apply the depth cross-attention layer at the end of each decoder block for training stability. The number of queries N for 6-view images is set as 900, which predict 10 object categories. The configurations of depth predictor, e.g., $[d_{min}, d_{max}]$ and k are the same as monocular experiments. We train MonoDETR-MV for 24 epochs (2x schedule) on 8 NVIDIA A100 GPUs with a batch size of 8. We adopt AdamW (Loshchilov & Hutter, 2018) optimizer with weight decay 10^{-2} and utilize the learning rate 2×10^{-4} with the cosine scheduler.

A.6 ADDITIONAL VISUALIZATION

In Figure 5, we show the detection results of our MonoDETR and the variant without the depth-guided transformer on KITTI *val* set. Benefited from the depth guidance, MonoDETR obtains a global understanding of the scene-level spatial structure and the inter-object relations. This enables MonoDETR to well detect the objects occluded by others or truncated by images, and filter out the objects of ignored categories, e.g., van and truck.

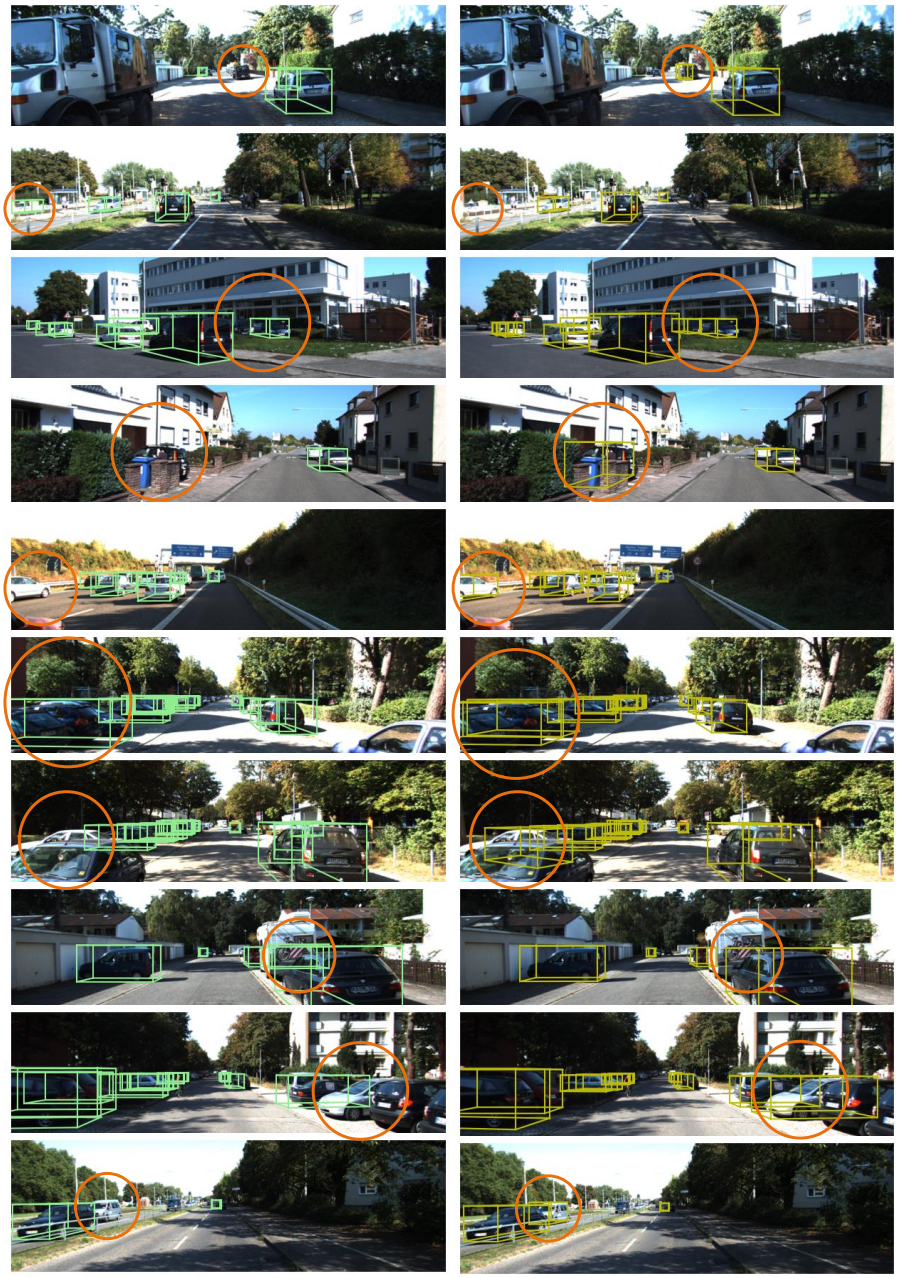

Figure 5: **Visualization of detection results.** We utilize green boxes for the variant without depth-guided transformer (Left) and yellow boxes for MonoDETR (Right). We use red circles to emphasize the detection difference.

Figure 6: **Depth errors for different variants of MonoDETR.** The x axis and y axis denote the AP_{3D} under the moderate level and the mean depth errors on KITTI *val* set, respectively.

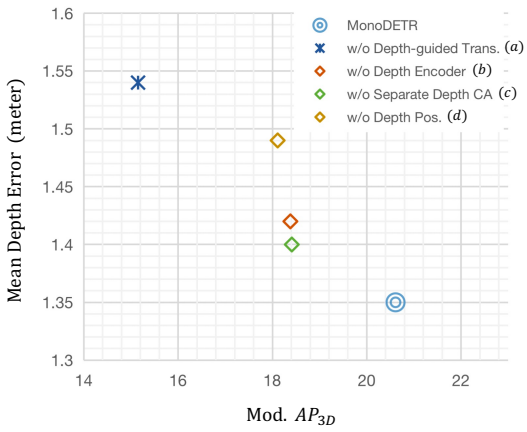

Table 12: **Quantitative results of depth errors.** We construct four network variants of MonoDETR by removing one of the components at a time. We respectively remove the depth-guided transformer, depth encoder, separate depth cross-attention layer, and depth positional encodings, denoted as '(a), (b), (c), (d)'. We show their AP_{3D} under the moderate level and the mean depth errors with standard deviations.

Architecture	AP_{3D} ↑	Depth Error ↓
MonoDETR	**20.61**	**1.35±2.07**
(a)	15.15	1.54±2.29
(b)	18.38	1.42±2.10
(c)	18.41	1.40±2.11
(d)	18.11	1.49±2.29

A.7 Depth Error Analysis

To demonstrate the effectiveness of our depth-guided design, we show the depth error comparison for different variants of MonoDETR. We utilize four network variants, denoted as '(a), (b), (c), (d)' in Figure 6 and Table 12. We calculate their predicted mean depth errors and standard deviations on KITTI *val* set. With our depth-guided transformer, the depth estimation can be well benefited, which reduces the mean error from 1.54 meters to 1.35 meters and improves the AP_{3D} by +5.46% under the moderate level. In addition, our best solution of 20.61% AP_{3D} performs lower error variance of ±2.07 than others, indicating our depth-guided transformer can produce more stable depth estimation of objects.

A.8 Anonymous Code Release

For reproducibility, we anonymously release our codes in https://anonymous.4open.science/r/MonoDETR_anonymous-FFC0/.