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

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

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

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

Method	General Settings			Feature Aggregation			Object Detection			Handcrafted Designs	
	DETR -based	Input View	Extra Data	Guided by	Global Aggre.	3D-to-2D Project	Query Type	Prediction Space	Bipartite Matching	Anchors	Post Processing
Mutli-view Methods											
DETR3D [33]	✓	Multi	-	Visual	×	✓	3D	3D	3D Loss	-	-
PETR [16]	✓	Multi	-	Visual	✓	-	3D	3D	3D Loss	-	-
PETRv2 [17]	✓	Multi	Temporal	Visual	✓	-	3D	3D	3D Loss	-	-
BEVFormer [12]	✓	Multi	Temporal	Visual	✓	✓	BEV	3D	3D Loss	-	-
Monocular Methods											
CaDDN [23]	×	Single	LiDAR	Center	×	-	×	3D	×	✓	NMS
MonoDTR [9]	×	Single	LiDAR	Center	×	-	×	Perspective	×	✓	NMS
MonoDETR	✓	Single	-	**Depth**	✓	-	**Depth**	Perspective	2D Loss	-	-

Table 1. **Detailed comparison of MonoDETR and existing methods for camera-based 3D object detection**.

A. Overview

- Appendix B: Detailed comparison of our MonoDETR and existing methods.
- Appendix C: Details of attribution prediction and loss functions.
- Appendix D: Additional results on KITTI *val* and *test* sets with the pedestrian and cyclist categories.
- Appendix E: Additional ablation study.
- Appendix G: Depth error analysis.
- Appendix F: Additional visualization.

† Corresponding author

B. Additional Related Work

Object Detection with Transformers. 2D object detectors [8, 13, 14, 25, 31] have attained excellent performance but count on cumbersome non-maximum suppression (NMS) post-processing with rule-based label assignment. To circumvent it, the seminal work DETR [2] constructs a novel framework by adapting the powerful transformer [32] for 2D detection. DETR detects objects on the image by an encoder-decoder architecture and conducts set prediction aided by Hungary Matching Algorithm [2]. 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 [41] designs sparse deformable attention mechanisms and achieves better performance with only 50-epoch training. ACT [38] boosts the time efficiency by adaptive clustering algorithms during inference. Besides, DETR is further enhanced by modulated co-attention [6], placing an-

chors [34], redesigning as two stages [29, 30], setting conditional attention [21], embedding dense priors [35], and so on [5, 10, 22]. For camera-based 3D object detection, we have discussed the DETR-based methods in Section 2 and Table 1 of the main paper. In Table 1, we further provide more detailed comparison to our MonoDETR as follows.

B.1. vs. MonoDTR [9]

Similarity: **(1) Single-view Detectors.** MonoDTR and our MonoDETR are both specially developed for 3D detection from monocular (single-view) images, and require no 3D-to-2D projection for feature aggregation. **(2) Depth Features.** MonoDTR also extracts depth features of the input image and directly fuses them with the visual features. **(3) Perspective Prediction.** MonoDTR and MonoDETR both predict 3D attributes of objects in the perspective view and then transform them into 3D space as the output.

Difference: **(1) Center-guided Paradigm.** MonoDTR is still a conventional center-guided method following YOLOv3 [24], and utilizes local features around centers to predict 3D properties, while our MonoDETR adopts a novel depth-guided paradigm. **(2) Not a DETR-based Detector.** MonoDTR is not a DETR-based method and only utilizes vanilla transformers to fuse depth and visual features, while our MonoDETR fully follows the pipeline of DETR for global feature aggregation. **(3) Using Extra Data.** MonoDTR requires additional dense depth maps projected from LiDAR to supervise the depth features, while our MonoDETR adopts foreground depth map that only needs discrete object-wise depth labels. **(4) No Object Queries.** MonoDTR contains no object queries, but relies on pre-defined anchors to detect objects and rule-based label assignment to compute losses. Our MonoDETR applies learnable object queries for detection and leverages Hungarian algorithm for bipartite matching. **(5) NMS Post-Processing.** MonoDTR still requires complicated NMS for post-processing to remove the duplicated boxes, while ours need not. **(6) Depth Positional Encodings.** We refer to the discussion in **??**.

B.2. vs. DETR3D [33] and PETR (v2) [16, 17]

Similarity: **(1) DETR-based Detectors.** DETR3D and PETR (v2) are also DETR-based methods with object queries for adaptive feature aggregation. **(2) No Handcrafted Designs.** Via bipartite matching, all DETR-based methods require no anchors or NMS post-processing.

Difference: **(1) Multi-view Detectors.** DETR3D and PETR (v2) are specially designed for 3D detection from multi-view images without single-view variant, while we aim at monocular 3D object detection. **(2) No Depth Cues.** They only extract multi-view features guided by visual appearances without exploring the geometric depth cues. **(3) Object Queries in 3D.** They initialize the object queries in 3D space and projects them onto multi-view images for feature aggregation, while MonoDETR directly adopts 2D depth-aware object queries without any 3D reference points. **(4) 3D Detection Space.** From sufficient multi-view features, they directly predict objects' attributes in the 3D space. As a monocular method, our MonoDETR extracts limited 3D semantics from single-view images and predicts the objects' attributes in perspective views on the images. **(5) Bipartite Matching via 3D Losses.** They utilize the 3D losses derived from 3D object queries for matching. In contrast, our MonoDETR adopts only 2D losses for matching, as discussed in Section 3.3 of the main paper.

B.3. vs. BEVFormer [12]

Similarity: BEVFormer is also a DETR-based method as DETR3D and PETR (v2), similar to MonoDETR by **(1) DETR-based Detectors** and **(2) No Handcrafted Designs**.

Difference: Similar to DETR3D and PETR (v2), BEVFormer obtains the following properties different from MonoDETR: **(1) Multi-view Detectors**, **(2) No Depth Cues**, **(3) 3D Detection Space**, and **(4) Bipartite Matching via 3D Losses**. Further, BEVFormer also utilizes previous frames as **(5) Temporal Extra Data**, and adopts **(6) Queries in BEV Space**. More importantly, concerning **(7) Purpose of Queries**, BEVFormer leverages BEV queries to generate the BEV representations from multi-view images, but MonoDETR's queries aim to detect objects from monocular images.

B.4. Depth Positional Encodings

As shown in Figure 1, we propose learnable depth positional encodings for f_D^e instead of conventional sinusoidal functions in the depth cross-attention layer. Existing work MonoDTR [9] also utilizes depth positional encodings in the transformer, but has three main differences from ours. **(1) Representation.** Our p_D is represented by meters, namely, one meter corresponding to a learnable depth embedding, but MonoDTR assigns each depth bin with an embedding. As the meters are more dense than depth bins, especially for farther distances, our meter-wise p_D can encode more sufficient depth positional cues. In addition, the optimal distribution of depth bins can be different for different datasets, but our meter-wise representation is dataset-agnostic and more general for various scenarios. **(2) Acquisition.** Our p_D is obtained by weighted summation of the depth-bin confidences and their corresponding depth values in Equation **??**, which is adaptive for different depth-bin confidences and incorporates the predicted depth prior

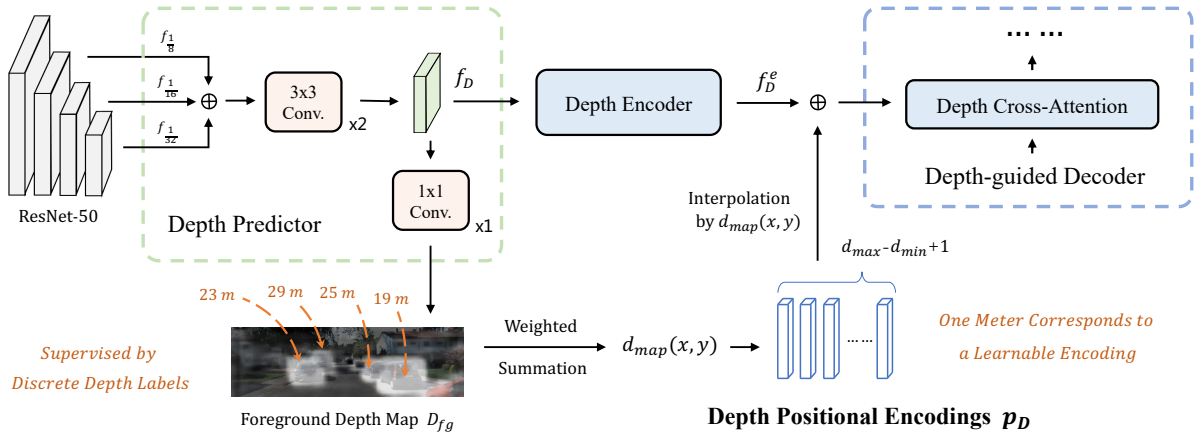

Figure 1. **Details of depth positional encodings.** In the depth cross-attention layer, we adopt learnable positional encodings for the depth embeddings by the depth value interpolation.

into the encodings. In contrast, MonoDTR adopts argmax to only utilize the depth encoding with the highest depth-bin confidence. **(3) Position.** MonoDTR utilizes depth positional encodings during the fusion of visual and depth features, which is irrelevant to object detection. Our p_D is equipped at the depth cross-attention layer at the depth-guided decoder, which directly guides the object queries for 3D attribute prediction.

C. 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 [7], and adopt Focal loss [14] for optimization, denoted as $\mathcal{L}_{class}$. Referring to FCOS [31], 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 [26] for the recovered 2D bounding box following DETR [3], denoted as $\mathcal{L}_{lrtb}$ and $\mathcal{L}_{GIoU}$.

3D Size (h_{3D}, w_{3D}, l_{3D}) and Orientation α. Instead of predicting the residuals to the mean shape values, we follow MonoDLE [20] to use the 3D IoU oriented loss for 3D sizes. We divide the heading angle into multiple bins with residuals and adopt MultiBin loss [4, 40] to optimize the prediction of orientation. The two losses are respectively denoted as $\mathcal{L}_{size3D}$ and $\mathcal{L}_{orien}$.

Object 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 following GUPNet [19], and $d_{map}(x_{3D}, y_{3D})$ obtained from the foreground depth map D_{fg} by weighted summation. We formulate it as

$$d_{geo} = f\,\frac{h_{3D}}{t+b}, \tag{1}$$

$$d_{pred} = (d_{reg} + d_{geo} + d_{map}(x_{3D}, y_{3D}))/3, \tag{2}$$

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 [4] for the overall d_{pred}, formulated as

$$\mathcal{L}_{depth} = \frac{\sqrt{2}}{\sigma}\|d_{gt} - d_{pred}\|_1 + \log(\sigma), \tag{3}$$

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 [18]	19.99	15.61	15.28	-	-	-	-	-	-
MonoPair [4]	24.12	18.17	15.76	55.38	42.39	37.99	61.06	47.63	41.92
MonoRCNN [27]	25.29	19.22	15.30	-	-	-	-	-	-
MonoDLE [20]	24.97	19.33	17.01	55.41	43.42	37.81	60.73	46.87	41.89
IAFA [39]	22.75	19.60	19.21	-	-	-	-	-	-
MonoGeo [37]	27.15	21.17	18.35	56.59	43.70	39.37	61.96	47.84	43.10
RTM3D [11]	24.74	22.03	18.05	52.59	40.96	34.95	56.90	44.69	41.75
GUPNet [19]	31.07	22.94	19.75	57.62	42.33	37.59	61.78	47.06	40.88
MonoDTR [9]	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

Table 2. **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	Extra data	Pedestrian, AP_{3D}			Cyclist, AP_{3D}		
		Easy	Mod.	Hard	Easy	Mod.	Hard
CaDDN [23]	LiDAR	12.87	8.14	6.76	7.00	3.41	3.30
MonoDTR [9]		15.33	10.18	8.61	5.05	3.27	3.19
M3D-RPN [1]	None	4.92	3.48	2.94	0.94	0.65	0.47
Movi3D [28]		8.99	5.44	4.57	1.08	0.63	0.70
MonoGeo [37]		8.00	5.63	4.71	4.73	2.93	2.58
MonoFlex [36]		9.43	6.31	5.26	4.17	2.35	2.04
MonoDLE [20]		9.64	6.55	5.44	4.59	2.66	2.45
MonoPair [4]		10.02	6.68	5.53	3.79	2.12	1.83
MonoDETR (Ours)	None	12.54	7.89	6.65	7.33	4.18	2.92

Table 3. **Performance of the pedestrian and cyclist categories on KITTI *test* set.**

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

Two Groups of Losses $\mathcal{L}_{2D}$ and $\mathcal{L}_{3D}$. For bipartite matching, we calculate the matching cost of each query-label pair only by the group of $\mathcal{L}_{2D}$, formulated as

$$\mathcal{L}_{2D} = \lambda_1 \mathcal{L}_{class} + \lambda_2 \mathcal{L}_{xy3D} + \lambda_3 \mathcal{L}_{lrtb} + \lambda_4 \mathcal{L}_{GIoU}, \quad (4)$$

where we set $\lambda_{1\sim4}$ as 2, 10, 5, 2, respectively. For $\mathcal{L}_{3D}$ in the final loss, we simply sum other 3D-related losses without weights as

$$\mathcal{L}_{3D} = \mathcal{L}_{size3D} + \mathcal{L}_{orien} + \mathcal{L}_{depth}. \quad (5)$$

Foreground Depth Map D_{fg}. We categorize the depth values of the foreground area into $k+1$ bins, and adopt Focal loss [14] to supervise each pixel in D_{fg}. The loss of the depth map is denoted as $\mathcal{L}_{dmap}$, which is not utilized for bipartite matching but serves as a term in the overall loss.

D. 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 2, where our MonoDETR all achieves the highest detection accuracy. Compared to the second-best MonoDTR [9] 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 3, we report the scores for pedestrian and cyclist categories on KITTI *test* set both under the IoU threshold of 0.5. As shown, MonoDETR achieves competitive AP_{3D} to other methods,

Settings	Easy	Mod.	Hard
d_{pred}	**28.84**	**20.61**	**16.38**
only d_{reg}	24.28	16.83	13.68
w/o d_{geo}	25.75	18.74	15.36
w/o d_{map}	26.04	18.89	15.45
w/o uncertainty σ	24.22	16.98	13.65

Table 4. **The design of depth prediction.** 'd_{pred}' denotes the average of three predicted depth values, 'd_{reg}', 'd_{geo}', and 'd_{map}' in Appendix C. 'uncertainty σ' denotes the standard of 'd_{reg}'.

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

	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

Table 6. **Transformer blocks and FFN channels.** FFN denotes the feed-forward neural network.

indicating our superior generalization ability on other categories. Compared to MonoDTR [9] with additional data input, it utilizes pre-defined anchors of average object sizes over the dataset, and just needs to predict the offset value to the average size. While MonoDETR introduces no such dataset prior and directly predicts their sizes. As the two categories are rare in the training data, it is harder for MonoDETR to learn their properties from scratch.

E. Additional Ablation Study

Depth Prediction. We regard the average of three predicted depth values as the overall depth, d_{pred}, of an object. As shown in Table 4, each depth component plays a part in the final depth prediction. The absence of either 'd_{geo}' or 'd_{map}' would harm the performance, since both two depth predictions are converted from other representations, i.e., the 3D/2D size and foreground depth map, which are independent from the depth regression head and might provide complementary geometric cues. Also, the uncertainty σ can largely boost the performance of monocular 3D detectors as analyzed in MonoDLE [20].

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 5, 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 6, we experiment with different block numbers of the visual encoder and depth-guided decoder, along with the latent channels of the 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 [3] 6-block encoder, 6-block decoder, and 1024-channel FFN for COCO [15] dataset, MonoDETR adopts a lighter-weight transformer architecture because of the limited training samples in KITTI [7] dataset.

Ablation Variants of Table 4 in the Main Paper. For the first ablation study in the main paper, we implement four variants of MonoDETR to investigate the effectiveness of our approach. **(1) 'w/o Depth-guide Trans.'.** We discard both depth guidance and the transformer architecture of MonoDETR to build a pure center-guided pipeline, which can be regarded as a re-implementation of MonoDLE [20]. After the ResNet-50 backbone, we apply one 1×1 convolutional layer to predict a center heatmap from the extracted visual feature, which predicts the projected 3D object centers. Concurrently, two 3×3 convolutional layers are adopted to predict 3D attributes via local visual features, which are supervised by losses in MonoDLE. This variant indicates the effectiveness of both transformer and depth guidance. **(2) 'w/o Transformer'.** On top of the variant, 'w/o Depth-guide Trans.', we add the prediction of foreground depth map in MonoDETR, including the lightweight depth predictor and the supervision of object-wise discrete depth labels. By this, the depth guidance can still be implicitly injected into the visual features without the transformer architecture, validating the adaptive feature aggregation in transformer. **(3) 'w/o Depth Guidance'.** Upon MonoDETR, we remove the prediction of foreground depth map

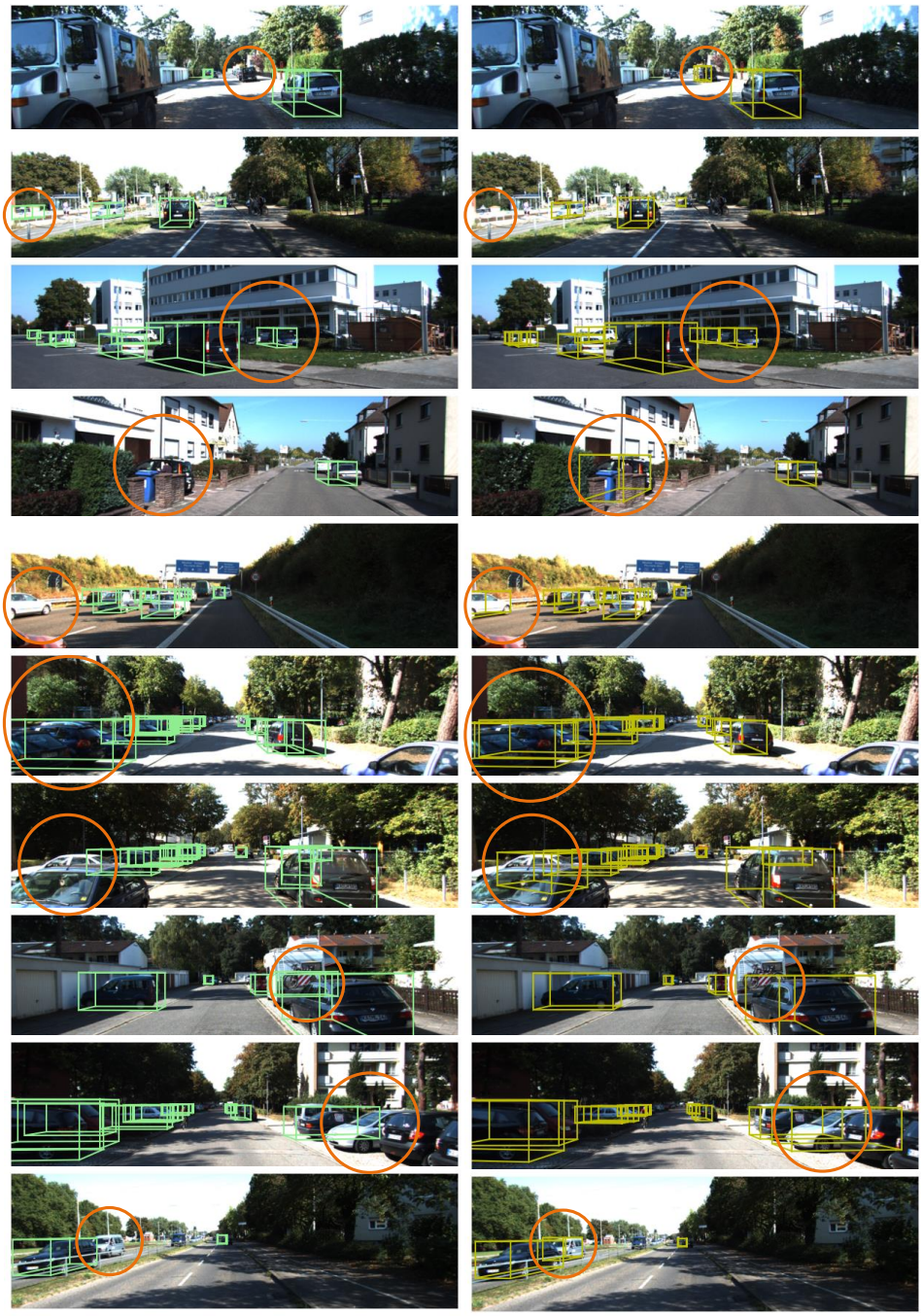

Figure 2. **Visualization of detection results.** We utilize green boxes for the variant without the depth-guided transformer (Left) and yellow boxes for MonoDETR (Right). We use red circles to emphasize the detection difference.

(depth predictor and depth encoder), and the depth cross-attention layer in the decoder. This derives a DETR model for adaptive monocular 3D object detection but not guided by depth cues. This variant demonstrates the significance of our depth-guided paradigm.

F. Additional Visualization

In Fig. 2, we show the detection results of our MonoDETR and the variant without the depth-guided transformer on KITTI *val* set. Benefiting from the depth guidance, MonoDETR obtains a global understanding of the

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)	18.94	1.49±2.29

Table 7. **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.

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.

G. 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 3 and Table 7. 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 method can produce a more stable depth estimation of objects.

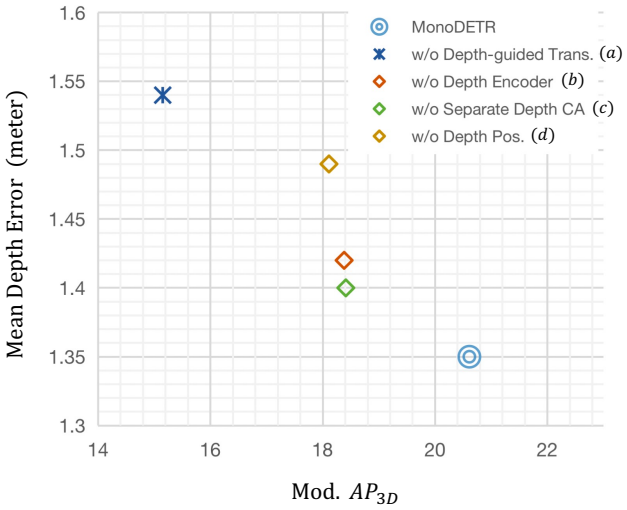

Figure 3. **Depth errors for different variants of MonoDETR.** The x axis and y axis denote the AP_{3D} under the moderate level and the mean depth errors on KITTI *val* set, respectively.

References

[1] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *IEEE International Conference on Computer Vision*, 2019. 4

[2] 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. 1

[3] 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. 3, 5

[4] 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, 4

[5] 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. 2

[6] 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. 1

[7] 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. 3, 5

[8] Ross Girshick. Fast r-cnn. In *IEEE International Conference on Computer Vision*, 2015. 1

[9] 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. 1, 2, 4, 5

[10] 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*, pages 13619–13627, 2022. 2

[11] 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. 4

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

[13] 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. 1

[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*, 2017. 1, 3, 4

[15] 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*, pages 740–755. Springer, 2014. 5

[16] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. *arXiv preprint arXiv:2203.05625*, 2022. 1, 2

[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*, 2022. 1, 2

[18] Zechen Liu, Zizhang Wu, and Roland Tóth. SMOKE: single-stage monocular 3d object detection via keypoint estimation. *CoRR*, abs/2002.10111, 2020. 4

[19] 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. 3, 4

[20] Xinzhu Ma, Yinmin Zhang, Dan Xu, Dongzhan Zhou, Shuai Yi, Haojie Li, and Wanli Ouyang. Delving into localization errors for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4721–4730, June 2021. 3, 4, 5

[21] 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. 2

[22] 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. 2

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

[24] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2

[25] 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

[26] 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*, pages 658–666, 2019. 3

[27] 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. 4

[28] 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. 4

[29] 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. 2

[30] 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. 2

[31] 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

[32] 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. 1

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

[34] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. *arXiv preprint arXiv:2109.07107*, 2021. 2

[35] 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. 2

[36] 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. 4

[37] 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. 4

[38] 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. 1

[39] 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. 4

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

[41] 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. 1