

SSD-MonoDETR: Supervised Scale-aware Deformable Transformer for Monocular 3D Object Detection

Xuan He, Fan Yang, Kailun Yang[†], Jiacheng Lin, Haolong Fu, Meng Wang, *Fellow, IEEE*,
Jin Yuan^{*†}, and Zhiyong Li^{*}

Abstract—Transformer-based methods have demonstrated superior performance for monocular 3D object detection recently, which aims at predicting 3D attributes from a single 2D image. Most existing transformer-based methods leverage both visual and depth representations to explore valuable query points on objects, and the quality of the learned query points has a great impact on detection accuracy. Unfortunately, existing unsupervised attention mechanisms in transformers are prone to generate low-quality query features due to inaccurate receptive fields, especially on hard objects. To tackle this problem, this paper proposes a novel “**Supervised Scale-aware Deformable Attention**” (SSDA) for monocular 3D object detection. Specifically, SSDA presets several masks with different scales and utilizes depth and visual features to adaptively learn a scale-aware filter for object query augmentation. Imposing the scale awareness, SSDA could well predict the accurate receptive field of an object query to support robust query feature generation. Aside from this, SSDA is assigned with a Weighted Scale Matching (WSM) loss to supervise scale prediction, which presents more confident results as compared to the unsupervised attention mechanisms. Extensive experiments on the KITTI and Waymo Open datasets demonstrate that SSDA significantly improves the detection accuracy, especially on moderate and hard objects, yielding state-of-the-art performance as compared to the existing approaches. Our code will be made publicly available at <https://github.com/mikasa3lili/SSD-MonoDETR>.

Index Terms—Monocular 3D Object Detection, Vision Transformer, Scene Understanding, Autonomous Driving.

I. INTRODUCTION

THE recent progresses in 3D-related studies have tremendously promoted their wide applications in multiple object tracking [1], [2], depth and ego-motion estimations [3], [4], and object detection for autonomous driving [5], [6] or

This work was supported in part by the National Natural Science Foundation of China (No.61976086, No.U21A20518, and No.62272157), in part by the Natural Science Foundation of Changsha (No. kq2202177), and in part by Hangzhou SurImage Technology Company Ltd. (*Corresponding authors: Jin Yuan and Zhiyong Li.*)

X. He, F. Yang, J. Lin, H. Fu, J. Yuan, and Z. Li are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China.

K. Yang and Z. Li are with the School of Robotics, Hunan University, China 410012, China.

K. Yang and Z. Li are also with the National Engineering Research Center of Robot Visual Perception and Control Technology, Hunan University, Changsha 410082, China.

M. Wang is with the School of Computer Science, Hefei University of Technology, Hefei 230009, China.

^{*}Corresponding authors: Jin Yuan and Zhiyong Li. (E-mail: yuan-jin@hnu.edu.cn, zhiyong.li@hnu.edu.cn.)

[†]Equal advising.

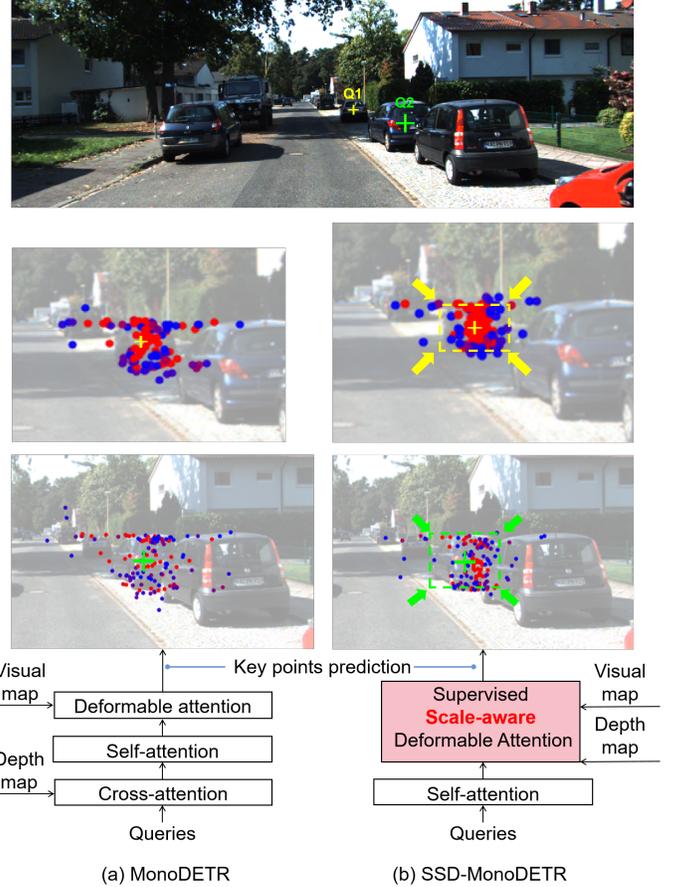


Fig. 1: Two examples to visualize the predicted key points (cycle) on object queries (cross) by (a) MonoDETR and (b) SSD-MonoDETR, where the red indicates the higher attention weight. It is obvious that SSDA could generate key points with higher quality by using the scale-aware mechanism.

indoor robotics [7], [8]. Benefiting from high-performance hardware, 3D object detection methods based on LiDAR points [9], [10], [11], [12] or binocular images [13], [14], [15] have achieved promising performance, but still suffer from high hardware costs. Comparatively, monocular 3D object detection [16], [17], [18], which predicts 3D attributes from a single image, could greatly save computation- and equipment costs and thus has attracted increasingly more research attention.

Most existing monocular 3D detection methods revise

the traditional 2D object detectors [19], [20], [21], which either require adjusting abundant manual parameters like anchors, region proposal and NMS [16], [22], [23], [24], or over-depend on exploring the geometric relationship between 2D and 3D objects [25], [26], [27]. Comparatively, newly proposed transformer-based detectors like MonoDETR [28] could avoid tedious parameter settings, and have demonstrated superior performance for monocular 3D object detection [29], [30], [28]. Given an input image with random query points, MonoDETR [28] first adopts several attention operations on visual and depth maps (see Figure 1 (a)) to search for relative key points for each query, and then integrates the features from these key points to predict 3D attributes. Technically, although deformable attention could find the amount of relative key points for a query, still suffers from the serious noisy point problem, especially on hard objects, where many key points deviate to background or irrelative objects (see Figure 1 (a)). This problem stems from the inherent mechanism of deformable attention, which is fed by a set of randomly initialized queries, and needs to adaptively explore relative key points for queries without the ability to estimate their receptive fields. Without the assistance of a precise receptive field, the predicted key points for a query tend to drift to other similar objects, generating noisy key points outside the object. As a result, the feature aggregation from noisy points would significantly affect 3D attribute prediction.

To alleviate this problem, this paper proposes a Supervised Scale-aware Deformable Transformer for monocular 3D object detection (SSD-MonoDETR). Different from MonoDETR, SSD-MonoDETR reduces the depth cross-attention layer and introduces a novel Supervised Scale-aware Deformable Attention (SSDA) layer to emphasize key point prediction for object queries (see Figure 1 (b)). Specifically, given a query, SSDA first presets several masks with different scales to extract multi-scale local features for the query from the input visual map, as well as predicts the scale probability distribution of the query from the depth map. On this basis, SSDA then adopts a lightweight adaptive layer to learn the receptive field of the query, which could offer valuable scale information for keypoint prediction. To guide the learning of SSDA, we design a Weighted Scale Matching (WSM) loss to extract the scale probability distribution output by the SSDA layer and use the scale ground truth to supervise it. As a result, the inner parameter updating of deformable attention could receive direct supervision, thus owning the ability to estimate the receptive field of a query to guide key point generation as compared to the original deformable attention. Benefiting from this, the noisy key point generation could be well alleviated, and more accurate query features are extracted for 3D attribute prediction (see Figure 1 (b)). We conduct extensive experiments on the KITTI and Waymo Open datasets, and the experimental results demonstrate the effectiveness of SSDA especially on moderate and hard objects, yielding state-of-the-art performance as compared to the existing approaches.

At a glance, this work yields the following contributions:

- 1) We propose a Supervised Scale-aware Deformable Attention (SSDA) mechanism to improve the quality of

the learned object queries in transformers. Compared to deformable attention, SSDA could better predict the receptive field of an object query to support accurate key point generation, yielding high-quality query features for 3D attribute prediction.

- 2) We design a Weighted Scale Matching (WSM) loss on SSDA to directly supervise the scale learning of queries without extra labeling costs, which is more effective as compared to the existing unsupervised attention mechanisms in transformers.
- 3) An extensive set of experimental results on the KITTI and Waymo Open datasets demonstrates the leading performance of our approach as compared to the state-of-the-art approaches in moderate and hard subsets. Furthermore, the near real-time inference time indicates the high applicability of our method in real cases.

In the rest of this paper, we first introduce the works related to our research in Section II, and then we elaborate on the details of our framework in Section III. Finally, we conduct a comprehensive variety of comparative and evaluation experiments to verify the effectiveness of the proposed method in Section IV, and give the conclusion, limitation, and future perspective of our work in Section V.

II. RELATED WORK

Monocular 3D object detection aims to estimate the 3D attributes of objects from a single image, where depth estimation is an ill-posed problem thus especially difficult. To boost the performance, many approaches explore the assistance of external resources like depth maps [23], [31], [32], [33] or LiDAR [29], [34], [35]. Although effective, they inevitably bring extra costs to data collection and calculation. To this end, image-only methods without extra data have attracted increasing attention.

Image-only monocular 3D object detection: Most existing image-only monocular 3D object detection approaches [36], [37], [38], [39] revise their pipelines based on 2D object detectors [40], [41], [42]. To improve performance, the geometry prior or key point prediction are the common strategies. For the methods that use geometric prior, OFT-Net [43] proposes an orthographic feature transform to transcend the limitations of the image domain by converting image-based features into a 3D orthographic space. MonoRCNN [24] proposes a geometry-based distance decomposition to factor the object distance into more stable physical and projected 2D heights. Based on MonoRCNN, MonoRCNN++ [44] further models the joint probability distribution of the physical height and visual height. Monopair [45] further explores the spatial relationship between pairs of objects to augment the 3D location. MonoJSG [46] utilizes pixel-level geometric constraints to progressively refine the depth estimation. Another stream of methods first predicts the key points of the 3D bounding box and regards it as an auxiliary task. For example, RTM3D [47] predicts nine key points of a 3D bounding box to explore the 2D-3D geometric relationship to recover the 3D attributes. SMOKE [18] is built based on the CenterNet [48], which treats the objects as points and combines the key points estimation

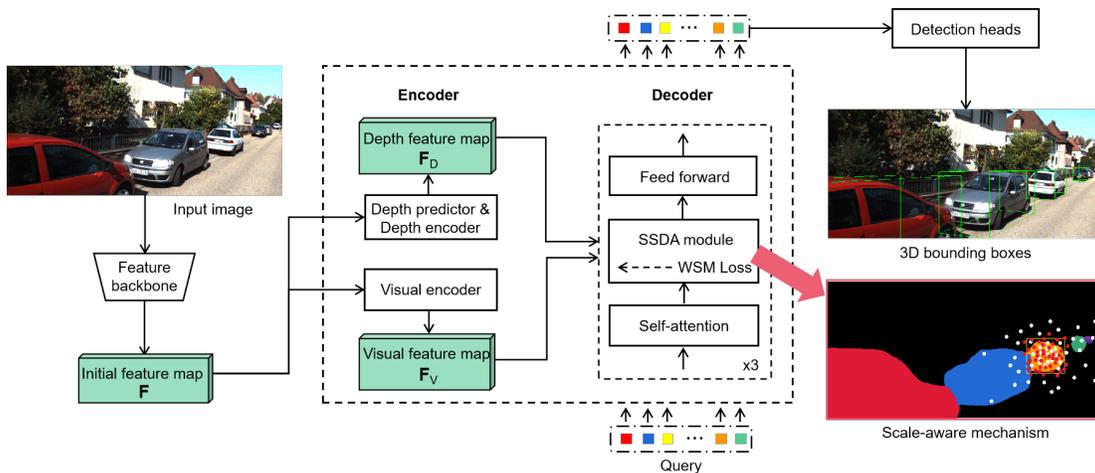


Fig. 2: The architecture of SSD-MonoDETR, where SSDA with a WSM loss is introduced to localize the scope of each object query, yielding a more accurate query receptive field (the red points) as compared to MonoDETR (the white points).

with 3D attributes regression. MonoDLE [49] revisits the issue of misalignment between the center of the 2D bounding box and the projected center of the 3D object, thus proposing to directly detect the projected 3D center. MonoFlex [50] addresses the prediction of long-tail truncated objects by decoupling the edge of the feature map, whose backbone utilizes three perspective-projection-based and one direct-regression-based depth estimators. Compared to MonoFlex, PDR [51] only requires one perspective-projection-based estimator to regress depth to realize a lighter architecture but better performance. With complicated designs, the above methods have achieved performance improvements, but there is still room for breakthroughs in accuracy and speed.

Transformer-based object detection: Transformer [52] was initially introduced in sequential modeling and has made significant advancements in the area of natural language processing (NLP). For object detection, DETR [53] first designs a novel pipeline based on the successful self-attention mechanism in transformers and abandons the complex manual settings in traditional 2D detectors. Based on DETR, there are many works [54], [55], [56] that strive to make further improvements. For example, Anchor DETR [57] designs an anchor-based object query thus the object queries could focus on the objects near the anchor points. [58] proposes a feature of interest selection mechanism to tackle the slow convergence of DETR caused by the Hungarian loss and the cross-attention mechanism. Conditional DETR [59] learns a conditional spatial query for the decoder, which could shrink the spatial range for queries and thus realize faster training. To accelerate the training process as well as improve the performance on small objects, Deformable DETR [60] proposes a novel deformable attention module where the queries only pay attention to a small set of key sampling points around themselves.

Transformer-based monocular 3D object detection: Inspired by the successful applications of transformers in 2D object detection [61], [62], [63], [64], recent researchers have turned to paying increasing attention to transformers for monocular 3D object detection, which could save cumbersome post-processing such as non-maximum suppression

(NMS). For instance, MonoDTR [29] introduces LiDAR point clouds as auxiliary supervision of its transformer pipeline and utilizes the learned depth features as the input query of a decoder. Without any extra data, DST3D proposes a novel structure [30] combining Swin Transformer [65] with deep layer aggregation [66] to realize 3D object detection. MonoPGC [67] proposes a depth-space-aware transformer and a depth-gradient positional encoding to combine 3D space positions with depth-aware features. MonoDETR [28] designs a depth-guided decoder, which utilizes a depth cross-attention layer to extract the global depth cues and a deformable attention layer to aggregate local visual features for queries. Thanks to the creative depth-guided decoder, MonoDETR achieves competitive performance.

Differently, this paper focuses on improving the quality of object queries in MonoDETR by imposing scale awareness for monocular 3D object detection. We newly design a Supervised Scale-aware Deformable Attention (SSDA) to replace deformable attention in MonoDETR, which could predict the receptive field of an object query to better support accurate feature generation for it.

III. METHOD

A. Architecture Overview

Figure 2 demonstrates the architecture of the proposed Supervised Scale-aware Deformable Transformer for monocular 3D object detection (SSD-MonoDETR). Specifically, SSD-MonoDETR first adopts a feature backbone to generate the initial feature map for a given image. On this basis, the visual and depth encoders are employed to extract global visual and depth representations respectively, which are then passed to a transformer decoder to generate valuable queries for object prediction by detection heads. Compared to MonoDETR, SSD-MonoDETR introduces a Supervised Scale-aware Deformable Attention (SSDA) module to integrate vision and depth information on each object query with a scale awareness, yielding more accurate receptive field as well as better multi-scale local feature representations for object queries. Moreover, we design a Weighted Scale Matching (WSM) loss for

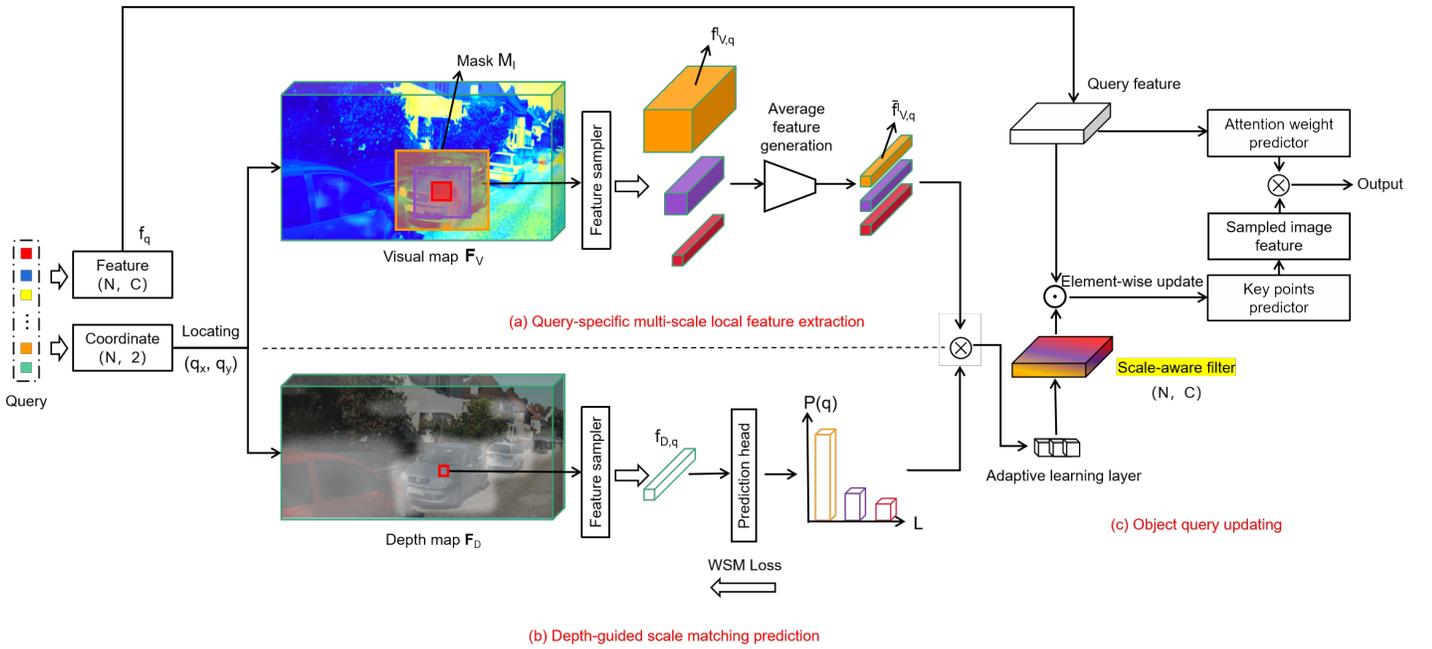


Fig. 3: The detailed implementation of SSDA, is composed of three steps: (a) query-specific multi-scale local feature extraction, (b) depth-guided scale matching prediction, and (c) object query updating.

SSDA to impose scale awareness on object queries. As a result, our approach could better localize the scope of a query for accurate feature extraction.

B. Supervised Scale-aware Deformable Attention

SSDA belongs to the transformer decoder in SSD-MonoDETR and receives the visual embedding \mathbf{F}_V , the depth embedding \mathbf{F}_D , and a set of learnable queries \mathcal{Q} from the last self-attention layer. Specifically, SSDA first extracts multi-scale local visual features for each query in \mathcal{Q} (see Section III-B1), as well as predicts the corresponding depth-guided scale matching probabilities for different scales (see Section III-B2). Then, SSDA integrates multi-scale local features and scale-matching probabilities by using an adaptively learning layer to form a scale-aware filter, which is used for key point prediction and query updating (see Section III-B3).

1) Query-specific Multi-scale Local Feature Extraction:

As Figure 3 shows, given the global visual representation $\mathbf{F}_V \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C}$, where H, W, C are the height, width, and channel of an input image, this step aims to generate multi-scale local visual features for each query $q \in \mathcal{Q}$. Here, each query q is initiated with a feature embedding $f_q \in \mathbb{R}^C$, and a position coordinates (q_x, q_y) , which are all learnable. Our approach first takes the query q as the center to generate N_l masks $\{M_{q,l}\}_{N_l}$ with the assumption that the scale of one of the masks is $l \times l$. We denote a set \mathbb{L} to contain all the preset scales: $\mathbb{L} = \{l_1, \dots, l_{N_l}\}$. Then, the local visual feature embedding $f_{V,q}^l \in \mathbb{R}^{(l \times l) \times C}$ is generated as follows:

$$f_{V,q}^l = \mathcal{S}_V(\mathbf{F}_V, M_{q,l}), \quad (1)$$

where \mathcal{S}_V is a feature sampler. For the $l \times l$ elements within $M_{q,l}$, \mathcal{S}_V extracts the corresponding feature embedding from \mathbf{F}_V according to their position coordinates. This generates a

local visual embedding $f_{V,q}^l$ to capture the visual representations for q at the scale of $l \times l$. Finally, we integrate all the local embeddings in $f_{V,q}^l$ to calculate an average local feature $\bar{f}_{V,q}^l \in \mathbb{R}^C$ as follows:

$$\bar{f}_{V,q}^l = \frac{1}{l \times l} \sum_{l \times l} f_{V,q}^l, \quad (2)$$

where $\bar{f}_{V,q}^l$ reflects the average local feature representation at the scale of $l \times l$. For each query q , our approach utilizes N_l masks to generate N_l local feature embedding $\{f_{V,q}^l\}_{N_l}$, which well captures multi-scale local feature information around q .

2) *Depth-guided Scale Matching Prediction:* As shown in Figure 3, given the depth representation $\mathbf{F}_D \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C}$, our approach first employs a depth feature sampler \mathcal{S}_D to directly extract the corresponding feature embedding $f_{D,q} \in \mathbb{R}^C$ from \mathbf{F}_D for each query q :

$$f_{D,q} = \mathcal{S}_D(\mathbf{F}_D, q). \quad (3)$$

Intuitively, the depth representation $f_{D,q}$ reflects the object scale of a query to some extent, and thus it is reasonable to design a projection function to map $f_{D,q}$ to a scale matching probability distribution $P(q)$, which is formulated as follows:

$$P(q) = \sigma(f_{D,q}), \quad (4)$$

where σ is a project function implemented by convolution followed by softmax. $P(q)$ is a N_l -dimensional vector to represent the N_l scale matching probabilities, which correspond to the N_l local feature representations $\{f_{V,q}^l\}_{N_l}$ in Section III-B1.

After the one-to-one paired Hungarian matching algorithm, all the queries are divided into object queries q^1 and non-object queries q^0 , and we can obtain the ground truth scale \hat{l}_{q^1} (i.e., the width of the 2D box) corresponding to the object queries q^1 . We expect the predicted scale to be consistent with the

true value. Thus, we devise a scale-matching loss to guide the learning of scale prediction:

$$L(q^1) = \frac{1}{N_l} \sum_{l \in \mathbb{L}} \|P(q^1, l) * l - \hat{l}_{q^1}\|_{L_1}, \quad (5)$$

where $\|\cdot\|_{L_1}$ is L_1 loss, $P(q^1, l)$ is the predicted probability on the scale l , and $P(q^1, l) * l$ represents the predicted scale of q^1 . $L(q^1)$ reflects the scale prediction error on q^1 , and different object queries have different error values during training. Generally, queries located on small objects are often prone to generating large errors. To accelerate the model's convergence, it is expected that more attention is paid to the queries with large errors, and thus we assign a penalty weight item $W(q^1)$ to $L(q^1)$ to form our **Weighted Scale Matching** (WSM) loss, which is expressed as:

$$L_{WSM} = \frac{1}{\mathbb{B}} \sum_{q^1 \in \mathbb{B}} W(q^1) L(q^1), \quad (6)$$

where \mathbb{B} represents the number of object queries in one training batch. $W(q^1)$ reflects the importance of object queries, and we adopt a query ranking mechanism to estimate it. Specifically, for all the object queries in \mathbb{B} , our approach generates two query ranking queues Q_P , Q_T in descending order according to their **predicted scales** and **true scales**, respectively, and $W(q^1)$ is calculated as:

$$W(q^1) = \log(|Index(Q_T, q^1) - Index(Q_P, q^1)| + 1), \quad (7)$$

where $Index(Q, q^1)$ represents the ranking index of q^1 in Q . When q^1 has the same index in both queues, $W(q^1) = 0$. Otherwise, the large index difference between $Index(Q_T, q^1)$ and $Index(Q_P, q^1)$ indicates that the predicted scale has a large error, and thus it is required to give a high penalty weight on q^1 . A graphic explanation of WSM loss is shown in Figure 4. Compared to the previous loss weighting approaches [68], [69], our ranking mechanism considers the relative weighting correlations among all the object queries in a training batch, which is more reasonable and global-aware.

As it can be observed, all the masks are preset to squares and we use the width labels of 2D boxes to supervise the scale of masks. As shown in Figure 1(a), suffered by the similar features of the surrounding objects, the key points of a query are very easily located on other objects without any supervision in the deformable attention layer. From the perspective of verity, all the objects are on the same horizontal plane, thus in a 2D image, the nearby objects can only appear on the left or right of the target object, but not on the top or bottom. To this end, only limiting and designing a loss to learn the width of the preset masks and restricting the height to be the same, can help reach a desirable balance between the performance and learning cost of the network.

3) *Object Query Updating*: The same as Deformable DETR [60], given an initial query, the query updating aims to search for several key points around the query, and then integrate these key points with attention weights to form the new query feature for the following attributes detection. As aforementioned, although MonoDETR utilizes both depth and visual features to predict key points, there still exist many

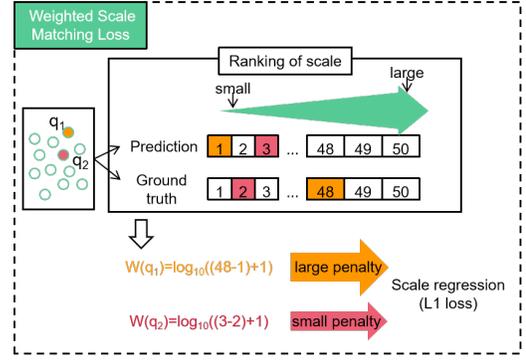


Fig. 4: A graphic explanation of WSM loss, where the ranking difference of q_1 is large thus the final WSM loss of q_1 has been further expanded.

noise key points. To alleviate this problem, as shown in Figure 3, our query updating utilizes both multi-scale local visual features and scale-matching probability to construct a scale-aware filter to help find more accurate key points. Concretely, given the multi-scale local feature and the scale-matching probability of a query, our approach first multiplies them and then employs an adaptive learning layer to generate a scale-aware filter. Here, the adaptive learning layer contains a lightweight convolutional network with a batch normalization layer and an active layer. The generation process of scale-aware filter $\mathcal{F} \in \mathbb{R}^{N \times C}$ can be expressed as:

$$\mathcal{F} = Relu(BN(Conv(\sum_{l=1}^L \sum_{q \in \mathbb{Q}} \vec{f}_{V,q}^l * P(q, l)))). \quad (8)$$

Next, we execute the element-wise operation between the filter and the initial query feature to generate the scale-aware features to better predict key points around their relative queries. For better convergence, we do not directly predict the coordinate of the key point but predict the offset $\Delta p_q = (\Delta q_x, \Delta q_y)$ from the query coordinate $p_q = (q_x, q_y)$:

$$\Delta p_q = Linear(\mathcal{F} \cdot f_q), \quad (9)$$

where the *Linear* operation project the input C -dimensional feature into $K \times 2$ -dimensional coordinate vector, and K is the number of key point for each query. Finally, the query feature is updated by integrating these key points features with the attention weights. The updated feature is obtained by:

$$\begin{aligned} & FeaUpd(\mathcal{F}, p_q) \\ &= \sum_{m=1}^M W_m [\sum_{k=1}^K A_{mqk} \cdot W'_m \mathbf{F}_V(p_q + \Delta p_{mqk})], \end{aligned} \quad (10)$$

where M is the head number of attention layer, W_m, W'_m are the learnable weights, A_{mqk} is the attention weights. $\mathbf{F}_V(p_q + \Delta p_{mqk})$ presents the key point features sampled on visual feature map \mathbf{F}_V by the bilinear interpolation.

C. Training Loss

SSD-MonoDETR is an end-to-end network, and all the components are jointly trained according to a composite loss

function, which consists of L_{2D} , L_{3D} and L_{WSM} . Specifically, the 2D object loss L_{2D} uses Focal loss [19] to estimate the object classes, L1 loss to estimate the 2D sizes (l, r, t, b) and projected 3D center (x_{3d}, y_{3d}) , and GIoU loss [70] for 2D box IoU. Finally, the L_{2D} could be expressed as:

$$L_{2D} = \lambda_1 L_{class} + \lambda_2 L_{2dsize} + \lambda_3 L_{xy3d} + \lambda_4 L_{giou}. \quad (11)$$

As for the 3D object loss L_{3D} , we follow MonoDLE [49] to predict 3D sizes (h_{3d}, w_{3d}, l_{3d}) and orientation angle α . For the depth prediction, we use the average of three depth values to present the predicted depth d_{pre} :

$$d_{pre} = \frac{1}{3}(d_{reg} + d_{geo} + d_{map}), \quad (12)$$

where d_{reg} is the depth value directly regressed by one detection head, d_{geo} is obtained by the relationship between 2D and 3D sizes:

$$d_{geo} = f \frac{h_{3D}}{t + b}, \quad (13)$$

where f is the camera focal length. d_{map} get value by the projected 3D center (x_{3d}, y_{3d}) and interpolation algorithm on the depth map (see MonoDETR [28] for more details). For the whole d_{pre} , a Laplacian aleatoric uncertainty loss [45] is adopted to form the final depth loss:

$$L_{depth} = \frac{2}{\sigma} \|d_{gt} - d_{pre}\|_1 + \log(\sigma), \quad (14)$$

where σ is the the standard deviation predicted together with d_{reg} , d_{gt} is the ground truth depth value. As a whole, the L_{3D} could be expressed as:

$$L_{3D} = \lambda_5 L_{3dsize} + \lambda_6 L_{angle} + \lambda_7 L_{depth}. \quad (15)$$

Additional, the proposed WSM loss L_{WSM} is equipped on the SSDA module to improve query features. As a result, our composite loss is expressed as:

$$L = L_{2D} + L_{3D} + \lambda_8 L_{WSM}, \quad (16)$$

where λ_1 to λ_8 are the balancing weights. For λ_8 , we will conduct its utility evaluation in the experiments.

IV. EXPERIMENTS

A. Experimental Setup

Dataset: We verify the proposed SSD-MonoDETR on the popular KITTI dataset [80] and Waymo Open dataset [81]. KITTI includes 7,481 training and 7,518 testing images. Considering the unseen labels on the testing set, we follow [41] to further split the training samples into 3,712 samples for the sub-training set and 3,769 samples for the validation set. We measure the detection results on three-level difficult samples (easy, moderate, and hard), and mainly evaluate the performance on the class ‘‘Car’’ by using the average precision (AP) in 3D space and the bird-eye view denoted as AP_{3D} and AP_{BEV} , respectively, which are at 40 recall positions. Also, the results of the class ‘‘Pedestrian’’ and ‘‘Cyclist’’ are listed by using the average precision (AP) in 3D space to make a comparison with the other existing approaches.

Waymo Open is a larger and more challenging autonomous

driving scene understanding dataset, which contains 798 training and 202 validation sequences and generates nearly $160k$ and $40k$ samples, respectively. For training and testing our method, we follow [75] to generate 52,386 training and 39,848 validation images from the front camera, and the images for training are constructed by sampling every third frame from the training sequences. For the evaluation metric, the Waymo Open dataset defines two difficulty levels ($Level_1$: points on object ≥ 5 , and $Level_2$: points on object ≥ 1) for all the instances, and each difficulty adopts two thresholds (0.5, 0.7) and four distance ranges ($Overall$, $0 - 30m$, $30 - 50m$, $50m - inf$) to evaluate the detection results.

Implementation Details: We adopt ResNet-50 [82] as the feature extraction backbone and all the attention layers have 8 heads. For the decoder, we set 3 blocks and the number N of input queries is set to 50. We select 5 scales $\{1, 3, 5, 7, 9\}$ of masks to extract local features for ‘‘Car’’. For ‘‘Pedestrian’’ and ‘‘Cyclist’’, whose aspect ratio (h/w) are larger than ‘‘Car’’ but have a relatively small size, to extract more accurate local features, we set the scales to $\{1, 3, 5\}$ and expand the vertical coordinates of mask elements to 3 and 2 times respectively. For the balancing weights λ_1 to λ_7 in Section III-C, we follow the settings in MonoDETR, which are: $\{2, 10, 5, 2, 1, 1, 1\}$. We set the λ_8 to 0.2 according to the following evaluation experiment. Our training is conducted on a single RTX A6000 GPU by using the Adam optimizer with a weight decay 10^{-4} for 200 epochs, where the batch size is 16 and the learning rate is 2×10^{-4} . Please note that all experimental results involving ‘‘Car’’, ‘‘Pedestrian’’, and ‘‘Cyclist’’ are obtained using a single-category training approach. For Waymo Open, we list the results of the ‘‘Vehicle’’ (*i.e.*, ‘‘Car’’) for comparison with other state-of-the-art methods, and the setting of masks is the same as KITTI. We also adopt the Adam optimizer with a weight decay of 10^{-4} to train our model for 40 epochs, the batch size of 40, and the learning rate of 5×10^{-4} .

B. Performance Comparison

Table I shows the comparison results of the ‘‘Car’’ category on the KITTI test and validation sets. We compare our approach with several advanced approaches by using or not using extra data. From the results, we reach the following conclusions:

- 1) Among all the approaches, SSD-MonoDETR achieves the best accuracies on moderate and hard objects, with $0.83\% \sim 4.81\%$ improvements. This improvement accelerates as sample difficulty increases, which is consistent with the characteristics of our method. This is because hard samples usually have small sizes, and thus the estimated query points by the previous algorithms are prone to deviate from objects, resulting in detection errors. Comparatively, SSDA well estimates the scales of query points by using depth-guided scale matching prediction, bringing a significant performance improvement.
- 2) We discover that our approach achieves greater performance improvements on AP_{BEV} as compared to AP_{3D} . This is because AP_{BEV} mainly focuses on evaluating

TABLE I: Performance of the “Car” category on KITTI test and val sets, where “v.✓” and “v.✗” represent the accuracy improvement compared to the methods with/without extra data. The first/second best results are highlighted in red / blue fonts and the improvements are noted in bold fonts.

Method	Reference	Test, AP_{3D}			Test, AP_{BEV}			Val, AP_{3D}			Val, AP_{BEV}			Time (ms)
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	
With extra data:														
MonoRUn [35]	CVPR2021	19.65	12.30	10.58	27.94	17.34	15.24	20.02	14.65	12.61	-	-	-	70
DDMP-3D [71]	CVPR2021	19.71	12.78	9.80	28.08	17.89	13.44	-	-	-	-	-	-	180
CaDDN [34]	CVPR2021	19.17	13.41	11.46	27.94	18.91	17.19	23.57	16.31	13.84	-	-	-	630
AutoShape [72]	ICCV2021	22.47	14.17	11.36	30.66	20.08	15.59	20.09	14.65	12.07	-	-	-	-
MonoDTR [29]	CVPR2022	21.99	15.39	12.73	28.59	20.38	17.14	24.52	18.57	15.51	33.33	25.35	21.68	37
DID-M3D [73]	ECCV2022	24.40	16.29	13.75	32.95	22.76	19.83	22.98	16.12	14.03	31.10	22.76	19.50	40
OPA-3D [27]	RAL2023	24.60	17.05	14.25	33.54	22.53	19.22	24.97	19.40	16.59	33.80	25.51	22.13	40
MonoPGC [67]	ICRA2023	24.68	17.17	14.14	32.50	23.14	20.30	25.67	18.63	15.65	34.06	24.26	20.78	46
Without extra data:														
MonoGeo [74]	CVPR2021	18.85	13.81	11.52	25.86	18.99	16.19	18.45	14.48	12.87	27.15	21.17	18.35	50
MonoFlex [50]	CVPR2021	19.94	13.89	12.07	28.23	19.75	16.89	23.64	17.51	14.83	-	-	-	30
GUPNet [38]	ICCV2021	20.11	14.20	11.77	-	-	-	22.76	16.46	13.72	31.07	22.94	19.75	-
DEVIANT [75]	ECCV2022	21.88	14.46	11.89	29.65	20.44	17.43	24.63	16.54	14.52	32.60	23.04	19.99	40
MonoDETR [28]	CVPR2022	23.65	15.92	12.99	32.08	21.44	17.85	28.84	20.61	16.38	37.86	26.95	22.80	20
MonoJSG [46]	CVPR2022	24.69	16.14	13.64	32.59	21.26	18.18	26.40	18.30	15.40	-	-	-	42
MonoRCNN++ [44]	WACV2023	20.08	13.72	11.34	-	-	-	19.07	14.87	12.59	26.41	20.80	17.27	-
MonoEdge [76]	WACV2023	21.08	14.47	12.73	28.80	20.35	17.57	25.66	18.89	16.10	33.71	25.35	22.18	37
PDR [51]	TCSVT2023	23.69	16.14	13.78	31.76	21.74	18.79	27.65	19.44	16.24	35.59	25.72	21.35	29
SSD-MonoDETR	-	24.52	17.88	15.69	33.59	24.35	21.98	29.53	21.96	18.20	38.00	29.44	26.94	21
Improvement	vs. Extra ✓	-	0.71	1.44	0.05	1.21	1.68	4.56	2.56	1.61	4.20	3.93	4.81	-
Improvement	vs. Extra ✗	-	1.41	2.11	-	2.24	3.38	0.69	1.35	1.82	0.14	2.49	4.14	-

TABLE II: Performance of the “Pedestrian” and “Cyclist” categories on KITTI test set.

Method	Pedestrian, AP_{3D}			Cyclist, AP_{3D}		
	Easy	Mod.	Hard	Easy	Mod.	Hard
With extra data:						
D4LCN [23]	4.55	3.42	2.83	2.45	1.67	1.36
MonoEF [77]	4.27	2.79	2.21	1.80	0.92	0.71
DDMP-3D [71]	4.93	3.55	3.01	4.18	2.50	2.32
DFR-Net [78]	6.09	3.62	3.39	5.69	3.58	3.10
MonoPSR [25]	6.12	4.00	3.30	8.70	4.74	3.68
CADNN [34]	12.87	8.14	6.76	7.00	3.41	3.30
MonoPGC [67]	14.16	9.67	8.26	5.88	3.30	2.85
Without extra data:						
MonoGeo [74]	8.00	5.63	4.71	4.73	2.93	2.58
MonoFlex [50]	9.43	6.31	5.26	4.17	2.35	2.04
MonoDLE [49]	9.64	6.55	5.44	4.59	2.66	2.45
MonoPair [45]	10.02	6.68	5.53	3.79	2.12	1.83
PDR [51]	11.61	7.72	6.40	2.72	1.57	1.50
MonoDETR [28]	12.54	7.89	6.65	7.33	4.18	2.92
MonoRCNN++ [44]	12.26	7.90	6.62	3.17	1.81	1.75
DEVIANT [75]	13.43	8.65	7.69	5.05	3.13	2.59
SSD-MonoDETR	12.64	9.88	8.58	7.79	5.76	4.33
Improvement v.✓	-	0.21	0.32	-	1.02	0.65
Improvement v.✗	-	1.23	0.89	0.46	1.58	1.41

the relative positions of cars with respect to streets in bird view, which is highly related to the positions of the estimated query points. Comparatively, AP_{3D} pays more attention to size parameters inside objects like the distance assessment from object center to ground, which

is less relative to the query points.

- 3) Our approach has a similar testing speed as compared to MonoDETR [28], but achieves better accuracy, especially on moderate and hard objects. Although SSDA brings extra computation costs, our approach reduces a cross-attention operation as compared to MonoDETR, resulting in a similar testing speed.
- 4) On the “Easy” subset, our method does not perform as well as the “Moderate” and “Hard” subsets. This stems from the difficulty classification criteria of the KITTI dataset, where the “Easy” objects are mostly close and their degree of occlusion by other objects is 0. Thus, the “Easy” objects usually have high-quality image features and are slightly suffered by the surrounding objects, which results in the proposed SSDA not being able to present an obvious effect as it does in the “Moderate” and “Hard” subsets.

Table II further shows the performance on “Pedestrian” and “Cyclist” categories. It is evident that these two categories present greater challenges compared to the “Car” category, mainly due to their smaller size and non-rigid body nature. Thanks to the effective SSDA module, our method surpasses all the previous methods without extra data in terms of performance across three difficulty levels, particularly for moderate and hard objects, where we achieve an approximate 2% improvement on “Pedestrian” and 1.5% improvement on “Cyclist”. The results presented in Table II validate the exceptional generality and scalability of our model, which only relies on easily accessible prior knowledge about different categories of scales. As a result, our proposed method effortlessly achieves

TABLE III: Vehicle performance on the Waymo Open Val set between different models. We use AP_{3D} ($LEVEL_1$ and $LEVEL_2$, IoU >0.5 and IoU >0.7) according to three object distance intervals.

Method	Reference	$LEVEL_1$ (IoU >0.5)			$LEVEL_2$ (IoU >0.5)			$LEVEL_1$ (IoU >0.7)			$LEVEL_2$ (IoU >0.7)						
		Overall0-30m	30-50m	50m-Inf	Overall0-30m	30-50m	50m-Inf	Overall0-30m	30-50m	50m-Inf	Overall0-30m	30-50m	50m-Inf				
With extra data:																	
PatchNet [31]	ECCV 20	2.92	10.03	1.09	0.23	2.42	10.01	1.07	0.22	0.39	1.67	0.13	0.03	0.38	1.67	0.13	0.03
PCT [79]	NIPS 21	4.20	14.70	1.78	0.39	4.03	14.67	1.74	0.36	0.89	3.18	0.27	0.07	0.66	3.18	0.27	0.07
Without extra data:																	
GUPNet [38]	ICCV 21	10.02	24.78	4.84	0.22	9.39	24.69	4.67	0.19	2.28	6.15	0.81	0.03	2.14	6.13	0.78	0.02
MonoJSG [46]	CVPR 22	5.65	20.86	3.91	0.97	5.34	20.79	3.79	0.85	0.97	4.65	0.55	0.10	0.91	4.64	0.55	0.09
DEVIANT [75]	ECCV 22	10.98	26.85	5.13	0.18	10.29	26.75	4.95	0.16	2.69	6.95	0.99	0.02	2.52	6.93	0.95	0.02
MonoRCNN++ [44]	WACV 23	11.37	27.95	4.07	0.42	10.79	27.88	3.98	0.39	4.28	9.84	0.91	0.09	4.05	9.81	0.89	0.08
SSD-MonoDETR	-	11.83	27.69	5.33	0.85	11.34	27.62	5.21	0.76	4.54	9.93	1.18	0.15	4.12	8.87	1.02	0.13

TABLE IV: Performance change by using different scale settings.

Multi-scale	Easy	Mod.	Hard
{3, 5, 7}	28.01	20.16	17.03
{3, 5, 7, 9}	29.22	21.67	17.40
{1, 3, 5, 7, 9}	29.53	21.96	18.20
{1, 3, 5, 7, 9, 11}	29.26	20.88	17.55
{1, 3, 5, 7, 9, 11, 13}	27.50	19.79	16.02

accurate detection of objects with diverse appearances.

Table III lists the average precision on 3D view (AP_{3D}) results on “Vehicle” of different methods on the Waymo Open Val set. On all the difficulty levels and IoU thresholds, our SSD-MonoDETR yields superior breakthroughs against all the other monocular 3D detectors in terms of “Overall” perspective, which proves the effectiveness of our proposed SSDA layer and WSM loss. On the distance $30m - inf$ with the strict IoU threshold of 0.7, our method also exceeds all the other methods, which is consistent with the design motivation of our proposed scale-aware mechanism, generating higher-quality query features thus obviously improving the accuracy for those hard and distant objects.

C. Evaluation on SSDA

In this experiment, we first try different scale settings in SSDA to observe the performance trend and then evaluate the quality of the generated query points by SSDA.

Multi-scale settings in SSDA: Before SSDA, the input images are resized to 1/16 during feature extraction, and thus most objects are reduced within 1 to 10 pixels in scale. Therefore, we set the scale value in the range of [1, 13] to observe the performance trend, which is shown in Table IV. Initially, we set three masks in SSDA with the scales {3, 5, 7}, which achieves unsatisfactory performance because many object scales are not covered. Then, we add one more mask with a scale of 9 and discover that the performances on easy and moderate objects are significantly improved. On this basis, the further introduction of the scale 1 boosts the performance, especially on hard objects, which usually have small sizes. Moreover, the performance begins to degrade as the scales 11 and 13 are added. We attribute this to that these scales exceed the sizes of most objects and would interrupt

TABLE V: The prediction precision of key points by MonoDETR, our method without WSM loss, and our complete method.

Method	Position Precision	Weighted Position Precision
MonoDETR	70.17 %	74.61%
SSD-MonoDETR/WSM	73.56 %	86.13 %
SSD-MonoDETR	76.34%	94.78%

the scale estimation on object queries.

Quality evaluation on query points: We evaluate the quality of query points by the three approaches: MonoDETR, SSD-MonoDETR, and SSD-MonoDETR/WSM (SSD-MonoDETR without using WSM loss), and Table V shows the comparison results, which are measured by two quantitative criteria: Position Precision, and Weighted Position Precision. Position precision is equal to the proportion of the number of key points falling inside objects to the total number of all the key points, while weighted position precision additionally multiplies the predicted attention weight of each key point to calculate position precision. Benefiting from SSDA, SSD-MonoDETR could better estimate the scales of key points and thus achieves better position precision as compared to MonoDETR. Without using WSM loss, SSD-MonoDETR/WSM suffers from a sharp performance drop, which proves the effectiveness of WSM loss in supervising the scale prediction. Moreover, we discover that the improvement in weighted position precision is enlarged as compared to position precision, which indicates that SSD-MonoDETR gives large attention weights to key points inside objects.

Figure 5 shows an example to visualize the generated key points by MonoDETR and SSD-MonoDETR. Obviously, as compared to MonoDETR, the distribution of the predicted key points by SSD-MonoDETR is more concentrated on objects with larger weights. As a result, SSD-MonoDETR could better learn query features to support object detection.

D. Evaluation of Weighted Scale Matching Loss

In this experiment, we first study with different weights λ_8 of the Weighted Scale Matching (WSM) loss in Equation 16 and then evaluate the effectiveness of the penalty weight item $W(q)$ in Equation 7.



Fig. 5: An example to illustrate the distribution of the generated key points by MonoDETR and our SSD-MonoDETR.

TABLE VI: Performance trend with respect to different values of λ_8 .

λ_8	Easy	Mod.	Hard
0	27.53	19.54	16.30
0.1	28.85	20.31	17.14
0.2	29.53	21.96	18.20
0.3	29.06	20.57	17.67
0.4	28.04	19.67	17.03
0.5	26.33	18.91	15.84

TABLE VII: Performance comparison among the three weighting schemes for the WSM loss.

Form	Easy	Mod.	Hard
WSM loss-0	28.81	20.68	17.12
WSM loss-L1	29.12	21.05	17.79
WSM loss	29.53	21.96	18.20

Table VI demonstrates the performance trend with respect to different weights. When λ_8 increases from 0 to 0.2, the performance is significantly improved by about 2% on all the samples. This is because the use of WSM loss supervises the scale prediction for query points, and thus could offer better query features for object detection. Moreover, the further increase of λ_8 would lead to a performance drop since a too-large weight on WSM loss would overshadow the utility of L_{2D} , L_{3D} detection losses, yielding prediction errors.

To verify the effectiveness of the penalty weight item $W(q)$ in Equation 7, we compare our WSM loss with the following two settings: WSM loss-0 without using the weight $W(q)$ by setting $W(q) = 1$, and WSM loss-L1 by setting $W(q) = \log L(q)$ which indicates the weight of a query is proportional to its prediction error. Table VII shows the performance comparison results. Compared to WSM loss-0, WSM loss-L1 achieves better performance because it assigns

larger training weights to error samples. Furthermore, our WSM loss considers the global-aware weighting correlations among all the queries in a training batch and thus could achieve the best performance.

E. Qualitative Results

Figure 6 shows six visualized examples with the detection results by MonoDETR and SSD-MonoDETR, where the first to fourth lines are the detection results of “Car”, the fifth and sixth lines are the visualization of “Pedestrian” and “Cyclist”, respectively. We discover that several small and partially blocked cars are missed by MonoDETR, as MonoDETR is easy to lose relevant query points for these hard objects. When it comes to the “Pedestrian” and “Cyclist” categories, which always have relatively small scales, MonoDETR generates more severe instances of missed or false detection. Comparatively, SSD-MonoDETR estimates the scale of an object query to generate more inner-the-object key points, as shown in Figure 5, thus extracting more relevant local features, which offers robust query features to the detection heads to support accurate detection, especially on hard samples.

V. CONCLUSIONS

We propose SSD-MonoDETR to first introduce Supervised Scale-aware Deformable Attention (SSDA) for monocular 3D object detection. Different from the existing transformer-based methods, SSDA could estimate the scale of a query to better capture its receptive field to impose the scale awareness on key point prediction, yielding better query features for 3D attribute prediction. Aside from this, SSDA adopts supervised learning with a WSM loss without extra labeling costs, which is more effective as compared to the unsupervised attention in transformers. Extensive experiments and analyses on KITTI have demonstrated the effectiveness of our approach. However, the proposed SSDA inevitably requires pre-setting different scales for different categories of objects, generating extra

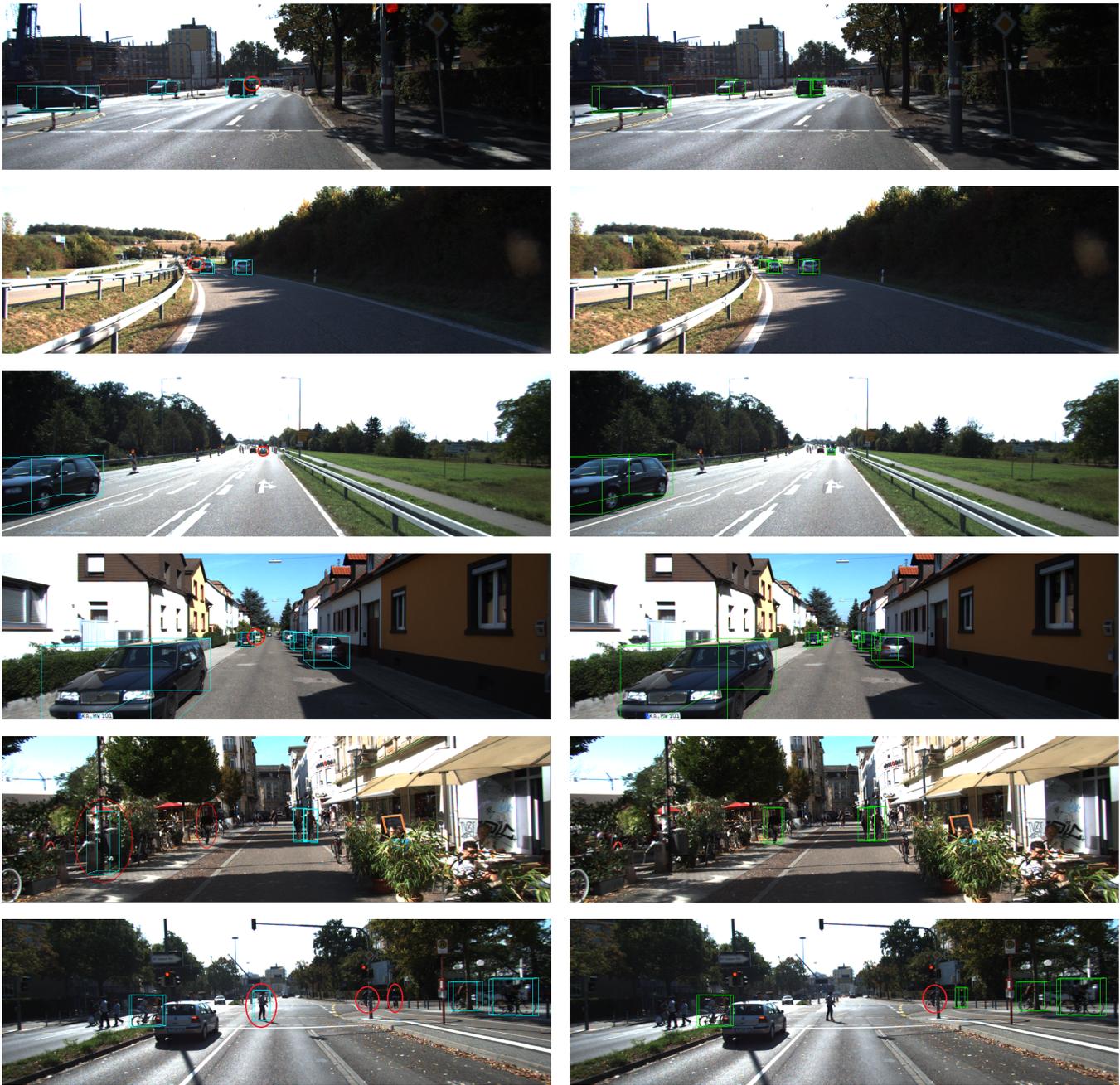


Fig. 6: Four representative examples to visualize the detection results between MonoDETR (left) and SSD-MonoDETR (right), where the red circles indicate the missing objects.

training costs on different categories. Furthermore, there is space to improve the testing speed of our approach. In the future, we intend to design a more flexible and generalized attention module and embed it into more transformer-based 3D object detection backbones for their further performance improvement in scene understanding.

REFERENCES

- [1] G. Wang, C. Peng, Y. Gu, J. Zhang, and H. Wang, "Interactive multi-scale fusion of 2D and 3D features for multi-object vehicle tracking," *IEEE Transactions on Intelligent Transportation Systems*, 2023.
- [2] G. Guo and S. Zhao, "3D multi-object tracking with adaptive cubature kalman filter for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 512–519, 2023.
- [3] J. Liu, Z. Cao, X. Liu, S. Wang, and J. Yu, "Self-supervised monocular depth estimation with geometric prior and pixel-level sensitivity," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 3, pp. 2244–2256, 2023.
- [4] G. Wang, J. Zhong, S. Zhao, W. Wu, Z. Liu, and H. Wang, "3D hierarchical refinement and augmentation for unsupervised learning of depth and pose from monocular video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 4, pp. 1776–1786, 2023.
- [5] K. Wang, T. Zhou, X. Li, and F. Ren, "Performance and challenges of 3D object detection methods in complex scenes for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1699–1716, 2023.
- [6] L. Wang *et al.*, "Multi-modal 3D object detection in autonomous driving:

- A survey and taxonomy," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 7, pp. 3781–3798, 2023.
- [7] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 567–576.
- [8] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2432–2443.
- [9] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3d object detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4490–4499.
- [10] H. Sheng *et al.*, "Improving 3D object detection with channel-wise transformer," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 2723–2732.
- [11] Q. Xu, Y. Zhong, and U. Neumann, "Behind the curtain: Learning occluded shapes for 3D object detection," in *AAAI Conference on Artificial Intelligence (AAAI)*, vol. 36, no. 3, 2022, pp. 2893–2901.
- [12] Z. Meng, X. Xia, R. Xu, W. Liu, and J. Ma, "HYDRO-3D: Hybrid object detection and tracking for cooperative perception using 3D LiDAR," *IEEE Transactions on Intelligent Vehicles*, 2023.
- [13] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun, "3D object proposals using stereo imagery for accurate object class detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1259–1272, 2018.
- [14] P. Li, X. Chen, and S. Shen, "Stereo R-CNN based 3D object detection for autonomous driving," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7636–7644.
- [15] Y. Liu, L. Wang, and M. Liu, "YOLOStereo3D: A step back to 2D for efficient stereo 3D detection," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 13 018–13 024.
- [16] G. Brazil and X. Liu, "M3D-RPN: Monocular 3D region proposal network for object detection," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9286–9295.
- [17] X. Weng and K. Kitani, "Monocular 3D object detection with pseudo-LiDAR point cloud," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 857–866.
- [18] Z. Liu, Z. Wu, and R. Tóth, "SMOKE: Single-stage monocular 3D object detection via keypoint estimation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 4289–4298.
- [19] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2999–3007.
- [20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, pp. 91–99, 2015.
- [21] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9626–9635.
- [22] G. Brazil, G. Pons-Moll, X. Liu, and B. Schiele, "Kinematic 3D object detection in monocular video," in *European Conference on Computer Vision (ECCV)*, vol. 12368, 2020, pp. 135–152.
- [23] M. Ding *et al.*, "Learning depth-guided convolutions for monocular 3D object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 669–11 678.
- [24] X. Shi, Q. Ye, X. Chen, C. Chen, Z. Chen, and T.-K. Kim, "Geometry-based distance decomposition for monocular 3D object detection," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 15 152–15 161.
- [25] J. Ku, A. D. Pon, and S. L. Waslander, "Monocular 3D object detection leveraging accurate proposals and shape reconstruction," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11 859–11 868.
- [26] T. Wang, Z. Xinge, J. Pang, and D. Lin, "Probabilistic and geometric depth: Detecting objects in perspective," in *Conference on Robot Learning (CoRL)*, 2022, pp. 1475–1485.
- [27] Y. Su *et al.*, "OPA-3D: Occlusion-aware pixel-wise aggregation for monocular 3D object detection," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1327–1334, 2023.
- [28] R. Zhang *et al.*, "MonoDETR: Depth-aware transformer for monocular 3D object detection," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.
- [29] K.-C. Huang, T.-H. Wu, H.-T. Su, and W. H. Hsu, "MonoDTR: Monocular 3D object detection with depth-aware transformer," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 4002–4011.
- [30] Z. Wu, X. Jiang, R. Xu, K. Lu, Y. Zhu, and M. Wu, "DST3D: DLA-swin transformer for single-stage monocular 3D object detection," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 411–418.
- [31] X. Ma, S. Liu, Z. Xia, H. Zhang, X. Zeng, and W. Ouyang, "Rethinking pseudo-LiDAR representation," in *European Conference on Computer Vision (ECCV)*, vol. 12358, 2020, pp. 311–327.
- [32] D. Park, R. Ambruş, V. Guizilini, J. Li, and A. Gaidon, "Is pseudo-lidar needed for monocular 3D object detection?" in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 3122–3132.
- [33] X. Ma, Z. Wang, H. Li, P. Zhang, W. Ouyang, and X. Fan, "Accurate monocular 3D object detection via color-embedded 3D reconstruction for autonomous driving," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6850–6859.
- [34] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander, "Categorical depth distribution network for monocular 3D object detection," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 8551–8560.
- [35] H. Chen, Y. Huang, W. Tian, Z. Gao, and L. Xiong, "MonoRun: Monocular 3D object detection by reconstruction and uncertainty propagation," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 10374–10383.
- [36] A. Mousavian, D. Anguelov, J. Flynn, and J. Košecká, "3D bounding box estimation using deep learning and geometry," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5632–5640.
- [37] T. Wang, X. Zhu, J. Pang, and D. Lin, "FCOS3D: Fully convolutional one-stage monocular 3D object detection," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021, pp. 913–922.
- [38] Y. Lu *et al.*, "Geometry uncertainty projection network for monocular 3D object detection," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 3091–3101.
- [39] T. Zhou, J. Chen, Y. Shi, K. Jiang, M. Yang, and D. Yang, "Bridging the view disparity between radar and camera features for multi-modal fusion 3D object detection," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1523–1535, 2023.
- [40] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3D object detection for autonomous driving," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2147–2156.
- [41] X. Chen *et al.*, "3D object proposals for accurate object class detection," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015, pp. 424–432.
- [42] A. Simonelli, S. R. Bulò, L. Porzi, M. Lopez-Antequera, and P. Kotschieder, "Disentangling monocular 3D object detection," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1991–1999.
- [43] T. Roddick, A. Kendall, and R. Cipolla, "Orthographic feature transform for monocular 3D object detection," in *British Machine Vision Conference (BMVC)*, 2018, p. 285.
- [44] X. Shi, Z. Chen, and T.-K. Kim, "Multivariate probabilistic monocular 3D object detection," in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023, pp. 4270–4279.
- [45] Y. Chen, L. Tai, K. Sun, and M. Li, "MonoPair: Monocular 3D object detection using pairwise spatial relationships," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 12 090–12 099.
- [46] Q. Lian, P. Li, and X. Chen, "MonoJSG: Joint semantic and geometric cost volume for monocular 3D object detection," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 1060–1069.
- [47] P. Li, H. Zhao, P. Liu, and F. Cao, "RTM3D: Real-time monocular 3D detection from object keypoints for autonomous driving," in *European Conference on Computer Vision (ECCV)*, vol. 12348, 2020, pp. 644–660.
- [48] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.
- [49] X. Ma *et al.*, "Delving into localization errors for monocular 3D object detection," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 4719–4728.
- [50] Y. Zhang, J. Lu, and J. Zhou, "Objects are different: Flexible monocular 3D object detection," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 3288–3297.

- [51] H. Sheng, S. Cai, N. Zhao, B. Deng, M.-J. Zhao, and G. H. Lee, "PDR: Progressive depth regularization for monocular 3D object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.
- [52] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 6000–6010.
- [53] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision (ECCV)*, vol. 12346, 2020, pp. 213–229.
- [54] M. Zheng *et al.*, "End-to-end object detection with adaptive clustering transformer," in *British Machine Vision Conference (BMVC)*, 2021, p. 226.
- [55] P. Gao, M. Zheng, X. Wang, J. Dai, and H. Li, "Fast convergence of DETR with spatially modulated co-attention," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 3601–3610.
- [56] Z. Dai, B. Cai, Y. Lin, and J. Chen, "UP-DETR: Unsupervised pre-training for object detection with transformers," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 1601–1610.
- [57] Y. Wang, X. Zhang, T. Yang, and J. Sun, "Anchor DETR: Query design for transformer-based detector," in *AAAI Conference on Artificial Intelligence (AAAI)*, vol. 36, no. 3, 2022, pp. 2567–2575.
- [58] Z. Sun, S. Cao, Y. Yang, and K. Kitani, "Rethinking transformer-based set prediction for object detection," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 3591–3600.
- [59] D. Meng *et al.*, "Conditional DETR for fast training convergence," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 3631–3640.
- [60] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *International Conference on Learning Representations (ICLR)*, 2021.
- [61] P. Zhen, X. Yan, W. Wang, T. Hou, H. Wei, and H.-B. Chen, "Towards compact transformers for end-to-end object detection with decomposed chain tensor structure," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 2, pp. 872–885, 2023.
- [62] L. Dai, H. Liu, H. Tang, Z. Wu, and P. Song, "AO2-DETR: Arbitrary-oriented object detection transformer," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 5, pp. 2342–2356, 2023.
- [63] P. Sun, T. Liu, X. Chen, S. Zhang, Y. Zhao, and S. Wei, "Multi-source aggregation transformer for concealed object detection in millimeter-wave images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 9, pp. 6148–6159, 2022.
- [64] Y. Liu, B. Schiele, A. Vedaldi, and C. Rupprecht, "Continual detection transformer for incremental object detection," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 23 799–23 808.
- [65] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9992–10 002.
- [66] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2403–2412.
- [67] Z. Wu, Y. Gan, L. Wang, G. Chen, and J. Pu, "MonoPGC: Monocular 3D object detection with pixel geometry contexts," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 4842–4849.
- [68] T. Huang, Z. Liu, X. Chen, and X. Bai, "EPNet: Enhancing point features with image semantics for 3D object detection," in *European Conference on Computer Vision (ECCV)*, vol. 12360, 2020, pp. 35–52.
- [69] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015, pp. 234–241.
- [70] H. Rezaatofghi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 658–666.
- [71] L. Wang *et al.*, "Depth-conditioned dynamic message propagation for monocular 3D object detection," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 454–463.
- [72] Z. Liu, D. Zhou, F. Lu, J. Fang, and L. Zhang, "AutoShape: Real-time shape-aware monocular 3D object detection," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 15 621–15 630.
- [73] L. Peng, X. Wu, Z. Yang, H. Liu, and D. Cai, "DID-M3D: Decoupling instance depth for monocular 3D object detection," in *European Conference on Computer Vision (ECCV)*, vol. 13661, 2022, pp. 71–88.
- [74] Y. Zhang *et al.*, "Learning geometry-guided depth via projective modeling for monocular 3D object detection," *arXiv preprint arXiv:2107.13931*, 2021.
- [75] A. Kumar, G. Brazil, E. Corona, A. Parchami, and X. Liu, "DEVIANT: Depth equivariant network for monocular 3D object detection," in *European Conference on Computer Vision (ECCV)*, vol. 13669, 2022, pp. 664–683.
- [76] M. Zhu, L. Ge, P. Wang, and H. Peng, "MonoEdge: Monocular 3D object detection using local perspectives," in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023, pp. 643–652.
- [77] Y. Zhou, Y. He, H. Zhu, C. Wang, H. Li, and Q. Jiang, "MonoEF: extrinsic parameter free monocular 3D object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 10 114–10 128, 2022.
- [78] Z. Zou *et al.*, "The devil is in the task: Exploiting reciprocal appearance-localization features for monocular 3D object detection," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 2693–2702.
- [79] L. Wang *et al.*, "Progressive coordinate transforms for monocular 3D object detection," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 13 364–13 377.
- [80] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354–3361.
- [81] S. Ettinger *et al.*, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9690–9699.
- [82] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.