MonoMAE: Enhancing Monocular 3D Detection through Depth-Aware Masked Autoencoders

Xueying Jiang[1], **Sheng Jin**[1], **Xiaoqin Zhang**[2], **Ling Shao**[3], **Shijian Lu**[1*]
[1]S-Lab, Nanyang Technological University, Singapore
[2]College of Computer Science and Technology, Zhejiang University of Technology, China
[3]UCAS-Terminus AI Lab, University of Chinese Academy of Sciences, China

Abstract

Monocular 3D object detection aims for precise 3D localization and identification of objects from a single-view image. Despite its recent progress, it often struggles while handling pervasive object occlusions that tend to complicate and degrade the prediction of object dimensions, depths, and orientations. We design MonoMAE, a monocular 3D detector inspired by Masked Autoencoders that addresses the object occlusion issue by masking and reconstructing objects in the feature space. MonoMAE consists of two novel designs. The first is depth-aware masking that selectively masks certain parts of non-occluded object queries in the feature space for simulating occluded object queries for network training. It masks non-occluded object queries by balancing the masked and preserved query portions adaptively according to the depth information. The second is lightweight query completion that works with the depth-aware masking to learn to reconstruct and complete the masked object queries. With the proposed feature-space occlusion and completion, MonoMAE learns enriched 3D representations that achieve superior monocular 3D detection performance qualitatively and quantitatively for both occluded and non-occluded objects. Additionally, MonoMAE learns generalizable representations that can work well in new domains.

1 Introduction

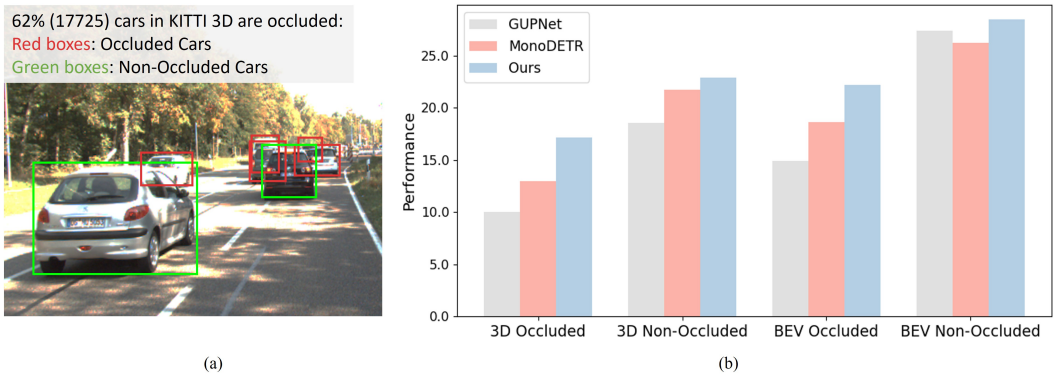
3D object detection has emerged as one key component in various navigation tasks such as autonomous driving, robot patrolling, etc. Compared with prior studies relying on LiDAR [69, 25, 64] or multi-view images [27, 33, 61], monocular 3D object detection offers a more cost-effective and accessible alternative which identifies objects and predicts their 3D locations from single-view images. On the other hand, monocular 3D object detection is much more challenging due to the lack of 3D information from multi-view images or LiDAR data.

Among various new challenges in monocular 3D detection, object occlusion, which exists widely in natural images as illustrated in Figure 1 (a), becomes a critical issue while predicting 3D locations in terms of object depths, object dimensions, and object orientations. Most existing monocular 3D detectors such as MonoDETR[66] and GUPNet [37] neglect the object occlusion issue which demonstrates clear performance degradation as illustrated in Figure 1 (b). A simple idea is to learn to reconstruct the occluded object regions whereby occluded objects can be handled similarly as non-occluded objects. On the other hand, reconstructing occluded object regions in the image space is complicated due to the super-rich variation of object occlusions in scene images.

Inspired by the Masked Autoencoders (MAE) [15] that randomly occludes image patches and reconstructs them in representation learning, we treat object occlusions as natural masking and train

*Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).



(a) (b)

Figure 1: Object occlusion is pervasive and affects monocular 3D detection: Object occlusion is pervasive, e.g., 62% (17725) cars in the KITTI 3D dataset suffer from various occlusions as illustrated in (a). Prevalent monocular 3D detection techniques such as GUPNet [37] and MonoDETR [66] are clearly affected by object occlusions in both 3D space (3D) and the bird's eye view (BEV) space as in (b). The proposed MonoMAE simulates and learns object occlusions by feature masking and completing which improves detection consistently for both occluded and non-occluded objects.

networks to complete occluded object regions to learn occlusion-tolerant representations. To this end, we design MonoMAE, a novel monocular 3D detection framework that adopts the idea of MAE by first masking certain object regions in the feature space (for simulating object occlusions) and then reconstructing the masked object features (for learning occlusion-tolerant representations). MonoMAE consists of a depth-aware masking module and a lightweight completion network. The depth-aware masking simulates object occlusions by masking the features of non-occluded objects adaptively according to the object depth information. It generates pairs of non-occluded and masked (i.e., occluded) object representations that can be directly applied to train the lightweight completion network, aiming for completing the occluded objects and learning occlusion-tolerant representations. Note that MonoMAE introduces little computational overhead in inference time as it requires no object masking in the inference stage.

The contributions of this work can be summarized in three major aspects. *First*, we design MonoMAE, a MAE-inspired monocular 3D detection framework that tackles object occlusions effectively by masking and reconstructing object regions in the feature space. To the best of our knowledge, this is the first work that explores masking-reconstructing for the task of monocular 3D object detection. *Second*, we design adaptive image masking and a lightweight completion network that mask non-occluded objects adaptively according to the object depth (for simulating object occlusions) and reconstruct the masked object regions (for learning occlusion-tolerant representations), respectively. *Third*, extensive experiments over KITTI 3D and nuScenes show that MonoMAE outperforms the state-of-the-art consistently and it can generalize to new domains as well.

2 Related Work

2.1 Monocular 3D Object Detection

Monocular 3D detection aims for the identification and 3D localization of objects from a single-view image. Most existing work can be broadly classified into two categories. The first employs convolutional neural networks, where most methods follow conventional 2D detectors [12, 23]. The standard approach learns monocular 3D detectors from single-view images only [17, 1, 8, 70, 66, 34]. In addition, several studies explore to leverage extra training data, such as LiDAR point clouds [39, 55, 6, 48, 46, 45], depth maps [11, 47, 45, 22, 38], and 3D CAD models [7, 35, 42] to acquire more depth information. Beyond that, several studies exploit the geometry relation between 2D and 3D spaces in different ways. For example, M3D-RPN [1] applies the powerful 2D detector FPN [49] for 3D detection. MonoDLE [40] aligns the centers of 2D and 3D boxes for better 3D localization. GUPNet [37] leverages uncertainty modeling to estimate the height of 3D boxes from the 2D boxes.

The second introduces powerful visual transformers [72, 21, 4, 65] for more accurate monocular 3D detection [19, 66, 71, 57, 56]. For example, MonoDTR [19] integrates context- and depth-aware features and injects depth positional hints into transformers. MonoDETR [66] modifies the transformer to be depth-aware and guides the detection process by contextual depth cues. However, most existing methods neglect object occlusions that exist widely in natural images and often degrade the performance of monocular 3D object detection clearly. We adopt the transformer architecture to learn occlusion-tolerant representations that can handle object occlusion effectively without requiring any extra training data or annotations.

2.2 Occlusions in 3D Object Detection

Object occlusion is pervasive in scene images and it has been investigated in several 2D and 3D vision tasks [63, 52, 10, 28, 26, 29, 30]. One typical approach learns to estimate the complete localization of occluded objects. For example, Mono-3DT [18] estimates complete 3D bounding boxes by re-identifying occluded vehicles from a sequence of 2D images. BtcDet [59] leverages object shape priors to learns to estimate the complete shapes of partially occluded objects. Several studies consider the degree of occlusions in training. For example, MonoPair [8] exploits the relation of paired samples and encodes spatial constraints of occluded objects from their neighbors. HMF [31] introduces an anti-occlusion loss to focus on occluded samples. Different from existing methods, the proposed MonoMAE learns enriched and occlusion-tolerant representations by masking and completing object parts in the feature space.

2.3 Masked Autoencoders in 3D Tasks

Masked Autoencoders (MAE) [15] learn visual representations by masking image patches and reconstructing them, and it has been explored in several point cloud pre-training studies. For outdoor point cloud pre-training, Occupancy-MAE [41] exploits range-aware random masking that employs three masking levels to deal with the sparse voxel occupancy structures of LiDAR point clouds. GD-MAE [62] introduces a Generative Decoder to merge the surrounding context to restore the masked tokens hierarchically. For indoor point cloud pre-training, Point-MAE [43] adopts MAE to directly reconstruct the 3D coordinates of masked tokens. I2P-MAE [67] introduces 2D pre-trained models, and it enhances 3D pre-training with diverse 2D semantics. PiMAE [5] learns cross-modal representations with MAE by interactively handling point clouds and RGB images. Different from existing studies, the proposed MonoMAE handles monocular 3D detection from single-view images and it focuses on object occlusions by learning to complete occluded object regions in the feature space.

3 Proposed Method

3.1 Problem Definition

Monocular 3D detection takes a single RGB image as input, aiming to classify objects and predict their 3D bounding boxes. The prediction of each object is composed of the object category C, a 2D bounding box B_{2D}, and a 3D bounding box B_{3D}. The 3D bounding box B_{3D} can be decomposed to the object 3D location (x_{3D}, y_{3D}, z_{3D}), the object dimensions in object height, width and length (h_{3D}, w_{3D}, l_{3D}), as well as the object orientation θ.

3.2 Overall Framework

Figure 2 shows the framework of the proposed MonoMAE. Given an input image I, the 3D Backbone first generates a sequence of 3D object queries $Q = [q_1, q_2, \cdots, q_K]$ (K denotes query number), and the Non-Occluded Query Grouping then classifies the queries into two groups including non-occluded queries $Q^{NO} = [q_1^{NO}, q_2^{NO}, \cdots, q_U^{NO}]$ and occluded queries $Q^O = [q_1^O, q_2^O, \cdots, q_V^O]$ (U and V are the number of non-occluded and occluded queries). The Non-Occluded Query Masking then masks Q^{NO} to produce masked queries according to their depth $D = [d_1, d_2, \cdots, d_U]$, leading to the masked queries $Q^M = [q_1^M, q_2^M, \cdots, q_U^M]$. The Query Completion further reconstructs Q^M to produce the completed queries $Q^C = [q_1^C, q_2^C, \cdots, q_U^C]$. Finally, the occluded queries Q^O and the completed queries Q^C are concatenated and fed to the Monocular 3D Detection for 3D detection

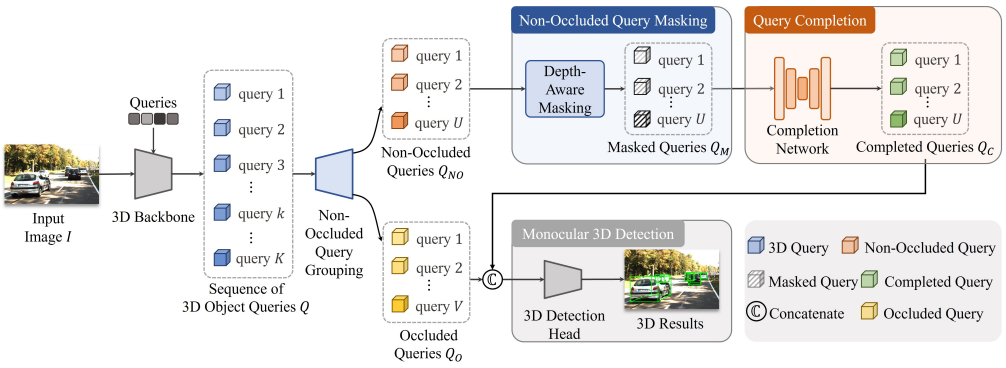


Figure 2: The framework of MonoMAE training: Given a single-view image, the 3D Backbone extracts 3D object query features which are grouped into non-occluded query features and occluded query features by the Non-Occluded Query Grouping. The Depth-Aware Masking then masks the non-occluded query features to simulate object occlusions adaptively based on the object depth, and the Completion Network then learns to reconstruct the masked queries. Finally, the completed and the occluded query features are concatenated to train the 3D Detection Head for 3D predictions.

predictions. Note the inference does not involve the Non-Occluded Query Masking, and it just concatenates the completion of occluded queries Q^O (i.e., Q^C) with the non-occluded queries Q^{NO} and feeds the concatenated queries to the 3D Detection Head for 3D predictions.

3.3 Non-Occluded Query Masking

Queries predicted by the 3D Backbone are either occluded or non-occluded, depending on whether the corresponding objects are occluded in the input image. In MonoMAE, we mask the non-occluded queries in the feature space to simulate occlusions, aiming to generate pairs of non-occluded and masked (i.e., occluded) queries for learning occlusion-tolerant object representations.

Specifically, we design a Non-Occluded Query Grouping module to identify non-occluded queries and then feed them into a Depth-Aware Masking module to synthesize occlusions, with more detail to be elaborated in the following subsections.

Non-Occluded Query Grouping. The Non-Occluded Query Grouping classifies the queries based on whether their corresponding objects are occluded or non-occluded. With no information about whether the input queries are occluded, we design an occlusion classification network Φ_{O} to predict the occlusion conditions $O^p = [o_1^p, o_2^p, \cdots, o_K^p]$ of queries $Q = [q_1, q_2, \cdots, q_K]$, where for the i-th query $o_i^p = \Phi_{\mathrm{O}}(q_i)$. The Non-Occluded Query Grouping can be formulated by:

$$\begin{cases} q_i \in Q^{NO} & \text{if } o_i^p = 0 \\ q_i \in Q^{O} & \text{if } o_i^p = 1 \end{cases}, \tag{1}$$

where $o_i^p = 0$ denotes the query is non-occluded, and $o_i^p = 1$ denotes the query is occluded. The occlusion classification network is trained with the occlusion classification loss L_{occ} as follows:

$$L_{occ} = CE(O^p, O^{gt}), \tag{2}$$

where CE is the $Cross\ Entropy$ loss. We adopted the bipartite matching [4] to match the predicted queries and objects in the image, where only matched queries have ground truth O^{gt} of KITTI 3D [13] about whether they are occluded or not.

Depth-Aware Masking. We design depth-aware masking to adaptively mask non-occluded query features to simulate occlusions in the feature space, aiming to create non-occluded and occluded (i.e., masked) pairs for learning occlusion-tolerant representations. As illustrated in Figure 3, the depth-aware masking determines the mask ratio according to the object depth - the closer the object, the larger the mask ratio, thereby compensating the information deficiency of distant objects. In addition, we simulate occlusions by masking in the feature space, as masking and reconstructing at the image level is complicated and computationally intensive.

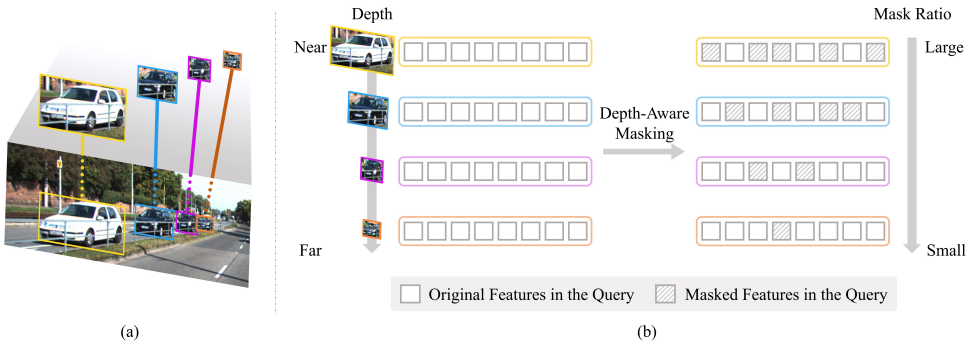


Figure 3: Illustration of the Depth-Aware Masking. (a) Objects farther away are usually smaller capturing less visual information. (b) The Depth-Aware Masking determines the mask ratio of an object according to its depth - the closer the object is, the larger the mask ratio is applied, thereby compensating the information deficiency for objects that have larger distances from the camera.

The depth-aware masking first obtains the query depth before query masking. Without backward gradient propagation, it adopts the 3D Detection Head to obtain the depth $D = [d_1, d_2, \cdots, d_U]$ for non-occluded queries. With the predicted depth, each non-occluded query is randomly masked as illustrated in Figure 3. Specifically, objects that are more distant from the camera are usually captured with less visual information. The depth-aware masking accommodates this by assigning a smaller masking ratio to them, thereby keeping more visual information for distant objects for proper visual representation learning.

The mask ratio r of each query is determined by:

$$r = 1.0 - d_i / D_{max}, \tag{3}$$

where r is the applied mask ratio for each query, d_i is the depth for the i-th query, and D_{max} is the maximum depth in datasets. The masks $M = [m_1, m_2, \cdots, m_U]$ generated for queries obey a Bernoulli Distribution.

Finally, the query masking is formulated by:

$$q_i^M = q_i^{NO} * m_i, \tag{4}$$

where q_i^M is the masked query, q_i^{NO} is the non-occluded query, and m_i is the generated mask.

3.4 Query Completion

The query completion learns to reconstruct the adaptively masked queries, aiming to produce completed queries whereby the network learns occlusion-tolerant representations that are helpful in detecting occluded objects. We design a completion network Φ_{C} to reconstruct the masked queries. The Completion Network has an hourglass structure consisting of three conv-bn-relu blocks and one conv-bn block for 3D query completion. The completed query q_i^C is obtained by:

$$q_i^C = \Phi_{\mathrm{C}}(q_i^M), \tag{5}$$

where q_i^M is the masked query. The Completion Network is trained under the supervision of the non-occluded queries before masking, where a completion loss L_{com} is formulated as follows:

$$L_{com} = L_1^s(Q^{NO}, Q^C), \tag{6}$$

where L_1^s denotes the SmoothL1 loss [14], Q^{NO} denotes the non-occluded queries, and Q^C denotes the queries completed by the Completion Network.

3.5 Loss Functions

The overall objective consists of three losses including L_{occ}, L_{com}, and L_{base} where L_{occ} and L_{com} are defined in Equation 2 and Equation 6, and L_{base} denote losses for supervising the 3D box

Table 1: Benchmarking on the KITTI 3D *test* set. All experiments adopt $AP|_{R_{40}}$ metric with an IoU threshold of 0.7. Best in **bold**, second underlined.

Method	Venue	Extra Data	$AP_{3D}(IoU=0.7)\|_{R_{40}}$			$AP_{BEV}(IoU=0.7)\|_{R_{40}}$		
			Easy	Moderate	Hard	Easy	Moderate	Hard
MonoRUn [6]	CVPR 21	LiDAR	19.65	12.30	10.58	27.94	17.34	15.24
MonoDTR [19]	CVPR 22		21.99	15.39	12.73	28.59	20.38	17.14
MonoDistill [9]	ICLR 22		22.97	16.03	13.60	31.87	22.59	19.72
DID-M3D [45]	ECCV 22		24.40	16.29	13.75	32.95	22.76	19.83
MonoNeRD [58]	ICCV 23		22.75	17.13	15.63	31.13	23.46	20.97
D4LCN [11]	CVPR 20	Depth	16.65	11.72	9.51	22.51	16.02	12.55
DDMP-3D [53]	CVPR 21		19.71	12.78	9.80	28.08	17.89	13.44
DD3D [44]	ICCV 21		23.22	16.34	14.20	30.98	22.56	20.03
Kinematic3D [2]	ECCV 20	Video	19.07	12.72	9.17	26.69	17.52	13.10
AutoShape [35]	ICCV 21	CAD	22.47	14.17	11.36	30.66	20.08	15.59
MonoFlex [68]	CVPR 21	None	19.94	13.89	12.07	28.23	19.75	16.89
MonoRCNN [50]	ICCV 21		18.36	12.65	10.03	25.48	18.11	14.10
GUPNet [37]	ICCV 21		20.11	14.20	11.77	-	-	-
DEVIANT [24]	ECCV 22		21.88	14.46	11.89	29.65	20.44	17.43
MonoCon [32]	AAAI 22		22.50	16.46	13.95	31.12	22.10	19.00
MonoDETR [66]	ICCV 23		25.00	16.47	13.58	33.60	22.11	18.60
MonoUNI [20]	NeurIPS 23		24.75	16.73	13.49	-	-	-
MonoCD [60]	CVPR 24		25.53	16.59	14.53	33.41	22.81	19.57
Ours	-	None	**25.60**	**18.84**	**16.78**	**34.15**	**24.93**	**21.76**

predictions. Specifically, L_{base} includes losses for supervising the 3D box predictions including each object's 3D locations, height, width, length and orientation. We set the weight for each loss item to 1.0, and the overall loss function is formulated as follows:

$$L = L_{occ} + L_{com} + L_{base}. \tag{7}$$

4 Experiments

4.1 Experimental Settings

Datasets. We benchmark our method over two public datasets in monocular 3D object detection.

• KITTI 3D [13] comprises 7,481 training images and 7,518 testing images, with training-data labels publicly available and test-data labels stored on a test server for evaluation. Following [7], we divide the 7,481 training samples into a new train set with 3,712 images and a validation set with 3,769 images for ablation studies.

• NuScenes [3] comprises 1,000 video scenes, including RGB images captured by 6 surround-view cameras. The dataset is split into a training set (700 scenes), a validation set (150 scenes), and a test set (150 scenes). Following [1, 50, 37, 24, 20], the performance on the validation set of nuScenes is reported.

In addition, we perform evaluations on the most representative Car category of KITTI 3D and nuScenes datasets as in prior studies [51, 50, 54, 66]

Evaluation Metrics. For KITTI 3D, we follow [51] and adopt $AP|_{R_{40}}$, the average of the AP of 40 recall points as the evaluation metric. We report the average precision on BEV and 3D object detection by $AP_{BEV}|_{R_{40}}$ and $AP_{3D}|_{R_{40}}$ with a threshold of 0.7 for both test and validation sets. For the nuScenes dataset, we adopt the mean absolute depth errors [50] in evaluations.

Implementation Details. We conduct experiments on one NVIDIA V100 GPU and train the framework for 200 epochs with a batch size of 16 and a learning rate of 2×10^{-4}. We use the AdamW [36] optimizer with weight decay 10^{-4}. We employ ResNet-50 [16] as the Transformer-based backbone and adopt the 3D detection head from [66] as our detection framework.

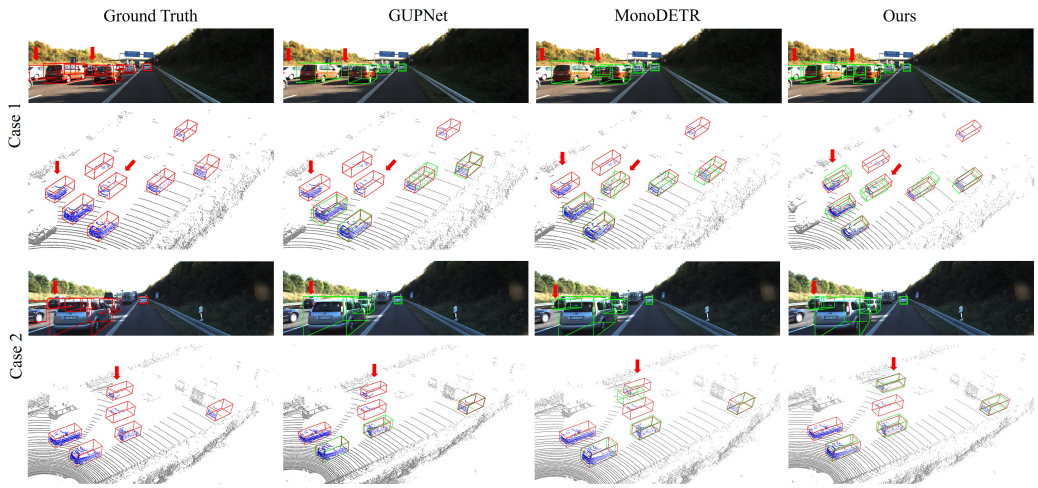


Figure 4: Detection visualization over the KITTI *val* set. Ground-truth annotations are highlighted by red boxes, and predictions by MonoMAE and two state-of-the-art methods are highlighted by green boxes. Red arrows highlight objects that have very different predictions across the compared methods. The ground truth of LiDAR point clouds is provided for visualization only, and they are not used in MonoMAE training. Best viewed in color and zoom-in.

Table 2: Ablation study of technical designs in MonoMAE on the KITTI 3D *val* set. 'NOQG', 'DAM', and 'CN' denote Non-Occluded Query Grouping, Depth-Aware Masking, and Completion Network, respectively. The symbol * indicates the baseline. The best results are highlighted in **bold**.

Index	NOQG	DAM	CN	$AP_{3D}(IoU=0.7)\|_{R_{40}}$ Easy	Moderate	Hard	$AP_{BEV}(IoU=0.7)\|_{R_{40}}$ Easy	Moderate	Hard
1*	✓			24.85	16.21	14.74	34.53	23.99	18.84
2		✓		23.33	15.09	13.20	32.68	21.80	17.43
3			✓	27.33	18.52	14.95	36.51	24.21	19.12
4	✓	✓		24.69	15.71	13.57	33.46	23.03	18.19
5	✓		✓	27.25	18.76	15.45	36.81	25.18	20.05
6		✓	✓	28.39	19.35	15.87	37.59	26.27	21.33
7	✓	✓	✓	**30.29**	**20.90**	**17.61**	**40.26**	**27.08**	**23.14**

4.2 Benchmarking with the State-of-the-Art

We benchmark MonoMAE with state-of-the-art monocular 3D object detection methods both quantitatively and qualitatively.

Quantitative Benchmarking. Table 1 shows quantitative experiments on the test set of dataset KITTI 3D, where all evaluations were performed on the official online test server [13] for fairness. We can see that MonoMAE achieves superior detection performance consistently across all metrics, without using any extra training data such as image depths, video sequences, LiDAR points, and CAD 3D models. In addition, MonoMAE outperforms more for the Moderate and Hard categories where various occlusions happen much more frequently than the Easy category. The superior performance is largely attributed to our designed depth-aware masking and completion network, which masks queries to simulate object occlusions in the feature space and reconstructs the masked queries to learn occlusion-tolerant visual representations, respectively.

Qualitative Benchmarking. Figure 4 shows qualitative benchmarking on the KITTI 3D val set. It can be observed that compared with two state-of-the-art methods GUPNet and MonoDETR, the proposed MonoMAE produces more accurate 3D detection consistently for both non-occluded and occluded objects, even for challenging scenarios like distant objects. Specifically, GUPNet and MonoDETR tend to miss the detection of highly occluded object in Cases 1 and 2 as highlighted by red arrows. As a comparison, MonoMAE performs clearly better by detecting those challenging objects successfully, demonstrating its superior capability on handling object occlusions.

Table 3: Ablation study of masking strategies on the KITTI 3D *val* set. The best results are in **bold**.

Index	Masking Strategy	$AP_{3D}(IoU=0.7)\|_{R_{40}}$			$AP_{BEV}(IoU=0.7)\|_{R_{40}}$		
		Easy	Moderate	Hard	Easy	Moderate	Hard
1	Image Masking	20.51	15.03	13.24	27.76	19.74	16.71
2	Query Masking (w/o Depth-Aware)	27.14	18.47	15.02	36.98	25.52	20.64
3	Query Masking (w/ Depth-Aware)	**30.29**	**20.90**	**17.61**	**40.26**	**27.08**	**23.14**

Table 4: Ablation study of the loss functions on the KITTI 3D *val* set. L_{occ} and L_{com} refer to the occlusion classification loss and the completion loss, respectively. The best results are in **bold**.

Index	L_{occ}	L_{com}	$AP_{3D}(IoU=0.7)\|_{R_{40}}$			$AP_{BEV}(IoU=0.7)\|_{R_{40}}$		
			Easy	Moderate	Hard	Easy	Moderate	Hard
1	✓		28.37	19.61	16.01	37.48	26.55	21.50
2		✓	26.36	19.15	15.88	36.76	26.49	22.62
3	✓	✓	**30.29**	**20.90**	**17.61**	**40.26**	**27.08**	**23.14**

4.3 Ablation Study

We conduct extensive ablation studies to examine the proposed MonoMAE. Specifically, we examine MonoMAE from the aspect of the technical designs, query masking strategies, as well as loss functions.

Network Designs. We examine the effectiveness of two key designs in MonoMAE, namely, the Depth-Aware Masking module (DAM) and the Completion Network (CN) (on the validation set of KITTI 3D), as shown in Table 2. We formulate the baseline by including the Non-Occluded Query Grouping module (NOQG), which does not affect the network training as both identified occluded and non-occluded queries are fed to train 3D detectors. When CN is not used in Rows 2 and 4, the 3D detection degrades as queries are masked but not reconstructed which leads to further information loss. While not incorporating DAM in Rows 3 and 5, the detection improves clearly compared with the baseline, as the completion helps learn better representations for naturally occluded queries. In addition, incorporating DAM and CN on top of NOQG in Row 7 performs clearly better than incorporating DAM and CN alone in Row 6, as the former applies masking and completion to non-occluded queries only. It also shows that masking naturally occluded queries to train the completion network is harmful to the learned representations.

Masking Strategies. We examine how different masking strategies affect monocular 3D detection. We studied three masking strategies as shown in Table 3. The first strategy masks the *input images* randomly, aiming to assess the value of masking and completing in the feature instead of image space. We can observe that the image-level masking yields clearly lower performance as compared with query masking in the feature space, largely due to the complication in masking and reconstructing images with a lightweight completion network. The second strategy masks query features randomly without considering object depths, aiming to evaluate the importance of object depths in query masking. The experiments show that random query masking outperforms the image-level masking significantly. The third strategy performs the proposed depth-aware query masking. It outperforms the feature-space random masking consistently, demonstrating the value of object depths for query masking.

Loss Functions. We examine the impact of the occlusion classification loss L_{occ} and the completion loss L_{com} in Equations 2 and 6, where L_{occ} supervises the occlusion classification network (in Non-Occluded Query Grouping) to predict whether the queries are occluded and L_{com} supervises the Completion Network to reconstruct the masked queries. As Table 4 shows, while implementing L_{occ} alone, the occlusion prediction is supervised while the query reconstruction is unsupervised. The network under such an objective does not learn well as the Completion Network cannot reconstruct object queries well without sufficient supervision. While implementing L_{com} alone, the occlusion classification network cannot identify occluded and non-occluded queries accurately where many occluded queries are fed for masking, leading to more query occlusion and poor detection performance. While employing both losses concurrently, the performance improves significantly as non-occluded queries can be identified for masking and reconstruction, leading to occlusion-tolerant representations.

Table 5: Comparison on inference speed of several monocular 3D detection methods. Ours* denotes the proposed MonoMAE without including the Completion Network.

Method	GUPNet [37]	MonoDTR [19]	MonoDETR [66]	Ours*	Ours
Inference Time (ms)	40	37	43	36	38

Table 6: Cross-dataset evaluations that perform training on the KITTI train set, and testing on the KITTI val and nuScenes val sets. We adopt the evaluation metric mean absolute error of the depth (↓). Best is highlighted in **bold**, and second underlined.

Method	KITTI Val				nuScenes frontal Val			
	0-20	20-40	40-∞	All	0-20	20-40	40-∞	All
M3D-RPN [1]	0.56	1.33	2.73	1.26	0.94	3.06	10.36	2.67
MonoRCNN [50]	0.46	1.27	2.59	1.14	0.94	2.84	8.65	2.39
GUPNet [37]	0.45	1.10	1.85	0.89	0.82	1.70	6.20	1.45
DEVIANT [24]	0.40	1.09	1.80	0.87	0.76	1.60	**4.50**	**1.26**
MonoUNI [20]	0.38	0.92	1.79	0.87	0.72	1.79	4.98	1.43
MonoMAE (Ours)	**0.36**	**0.91**	**1.74**	**0.86**	**0.71**	**1.57**	4.95	1.40

4.4 Discussions

Efficiency Comparison. We compare the inference time of several representative monocular 3D detection methods on the KITTI val set, where all compared methods are evaluated with one NVIDIA V100 GPU under the same computational environment for fairness. As Table 5 shows, GUPNet, MonoDTR, and MonoDETR have an average inference time of 40ms, 37ms, and 43ms for each image, respectively. As a comparison, the proposed MonoMAE takes the shortest inference time, demonstrating its good efficiency in monocular 3D detection. Further, we analyzed the Completion Network in terms of network parameters and floating-point operations per second (FLOPs), showing it has very limited 2.22G parameters and 0.08M in FLOPs.

Generalization Ability. We examine the generalization capability of the proposed MonoMAE by directly applying the KITTI-trained MonoMAE model to the car Category of the nuScenes validation set without additional training. The detection performance on the KITTI validation set is also reported for reference. Table 6 shows that MonoMAE attains the highest or second-highest detection performance across various metrics on the nuScenes frontal validation set. This indicates that despite the domain shift from KITTI to nuScenes, MonoMAE still maintains satisfactory performance. Since DEVIANT [24] is equivariant to the depth translations, it sometimes has higher performance.

5 Conclusion

This paper presents MonoMAE, a novel method inspired by the Masked Autoencoders (MAE) to deal with the pervasive occlusion problem in the monocular 3D object detection task. MonoMAE consists of two key designs. The first is a depth-aware masking module, which simulates the occlusion for non-occluded object queries in the feature space during training. The second is a lightweight completion network, which reconstructs and completes the masked object queries. Quantitative and qualitative experiment results show that MonoMAE learns enhanced 3D representations and achieves superior monocular 3D detection performance for both occluded and non-occluded objects. Moving forward, we plan to investigate generative approaches to simulate natural occlusion patterns for various 3D detection tasks.

Limitations. MonoMAE leverages depth-aware masking to mask non-occluded queries to simulate object occlusions in the feature space. However, the masked queries may have different patterns as compared with the features of naturally occluded object queries. Such a gap could affect the reconstruction of complete queries and monocular 3D detection performance. This issue could be mitigated by introducing generative networks that learn distributions from extensive real-world data for generating occlusion patterns that are more similar to natural occlusions.

Acknowledgments and Disclosure of Funding

This study is supported under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

References

[1] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9287–9296, 2019.

[2] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele. Kinematic 3d object detection in monocular video. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pages 135–152. Springer, 2020.

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[5] Anthony Chen, Kevin Zhang, Renrui Zhang, Zihan Wang, Yuheng Lu, Yandong Guo, and Shanghang Zhang. Pimae: Point cloud and image interactive masked autoencoders for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5291–5301, 2023.

[6] Hansheng Chen, Yuyao Huang, Wei Tian, Zhong Gao, and Lu Xiong. Monorun: Monocular 3d object detection by reconstruction and uncertainty propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10379–10388, 2021.

[7] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2147–2156, 2016.

[8] Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li. Monopair: Monocular 3d object detection using pairwise spatial relationships. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12093–12102, 2020.

[9] Zhiyu Chong, Xinzhu Ma, Hong Zhang, Yuxin Yue, Haojie Li, Zhihui Wang, and Wanli Ouyang. Monodistill: Learning spatial features for monocular 3d object detection. *International Conference on Learning Representations*, 2022.

[10] Huazhen Chu, Lisha Mo, Rongquan Wang, Tianyu Hu, and Huimin Ma. Visibility of points: Mining occlusion cues for monocular 3d object detection. *Neurocomputing*, 502:48–56, 2022.

[11] Mingyu Ding, Yuqi Huo, Hongwei Yi, Zhe Wang, Jianping Shi, Zhiwu Lu, and Ping Luo. Learning depth-guided convolutions for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1000–1001, 2020.

[12] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6569–6578, 2019.

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

[14] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1440–1448, 2015.

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[17] Tong He and Stefano Soatto. Mono3d++: Monocular 3d vehicle detection with two-scale 3d hypotheses and task priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8409–8416, 2019.

[18] Hou-Ning Hu, Qi-Zhi Cai, Dequan Wang, Ji Lin, Min Sun, Philipp Krahenbuhl, Trevor Darrell, and Fisher Yu. Joint monocular 3d vehicle detection and tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5390–5399, 2019.

[19] Kuan-Chih Huang, Tsung-Han Wu, Hung-Ting Su, and Winston H Hsu. Monodtr: Monocular 3d object detection with depth-aware transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4012–4021, 2022.

[20] Jinrang Jia, Zhenjia Li, and Yifeng Shi. Monouni: A unified vehicle and infrastructure-side monocular 3d object detection network with sufficient depth clues. In *Advances in Neural Information Processing Systems*, 2023.

[21] Xueying Jiang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Domain generalization via balancing training difficulty and model capability. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[22] Xueying Jiang, Sheng Jin, Lewei Lu, Xiaoqin Zhang, and Shijian Lu. Weakly supervised monocular 3d detection with a single-view image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[23] Sheng Jin, Xueying Jiang, Jiaxing Huang, Lewei Lu, and Shijian Lu. Llms meet vlms: Boost open vocabulary object detection with fine-grained descriptors. *International Conference on Learning Representations*, 2024.

[24] Abhinav Kumar, Garrick Brazil, Enrique Corona, Armin Parchami, and Xiaoming Liu. Deviant: Depth equivariant network for monocular 3d object detection. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pages 664–683. Springer, 2022.

[25] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019.

[26] Ke Li and Jitendra Malik. Amodal instance segmentation. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pages 677–693. Springer, 2016.

[27] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pages 1–18. Springer, 2022.

[28] Zhixuan Li, Weining Ye, Tingting Jiang, and Tiejun Huang. 2D amodal instance segmentation guided by 3D shape prior. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pages 165–181, 2022.

[29] Zhixuan Li, Weining Ye, Tingting Jiang, and Tiejun Huang. GIN: Generative invariant shape prior for amodal instance segmentation. In *IEEE Transactions on Multimedia*, pages 3924–3936, 2023.

[30] Zhixuan Li, Weining Ye, Juan Terven, Zachary Bennett, Ying Zheng, Tingting Jiang, and Tiejun Huang. MUVA: A new large-scale benchmark for multi-view amodal instance segmentation in the shopping scenario. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23504–23513, 2023.

[31] He Liu, Huaping Liu, Yikai Wang, Fuchun Sun, and Wenbing Huang. Fine-grained multilevel fusion for anti-occlusion monocular 3d object detection. *IEEE Transactions on Image Processing*, 31:4050–4061, 2022.

[32] Xianpeng Liu, Nan Xue, and Tianfu Wu. Learning auxiliary monocular contexts helps monocular 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1810–1818, 2022.

[33] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pages 531–548. Springer, 2022.

[34] Zechen Liu, Zizhang Wu, and Roland Tóth. Smoke: Single-stage monocular 3d object detection via keypoint estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 996–997, 2020.

[35] Zongdai Liu, Dingfu Zhou, Feixiang Lu, Jin Fang, and Liangjun Zhang. Autoshape: Real-time shape-aware monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15641–15650, 2021.

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

[37] 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*, pages 3111–3121, 2021.

[38] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking pseudo-lidar representation. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pages 311–327. Springer, 2020.

[39] Xinzhu Ma, Zhihui Wang, Haojie Li, Pengbo Zhang, Wanli Ouyang, and Xin Fan. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6851–6860, 2019.

[40] 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*, pages 4721–4730, 2021.

[41] Chen Min, Liang Xiao, Dawei Zhao, Yiming Nie, and Bin Dai. Occupancy-mae: Self-supervised pre-training large-scale lidar point clouds with masked occupancy autoencoders. *IEEE Transactions on Intelligent Vehicles*, 2023.

[42] J Krishna Murthy, GV Sai Krishna, Falak Chhaya, and K Madhava Krishna. Reconstructing vehicles from a single image: Shape priors for road scene understanding. In *International Conference on Robotics and Automation*, pages 724–731. IEEE, 2017.

[43] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pages 604–621. Springer, 2022.

[44] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3142–3152, 2021.

[45] Liang Peng, Xiaopei Wu, Zheng Yang, Haifeng Liu, and Deng Cai. Did-m3d: Decoupling instance depth for monocular 3d object detection. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pages 71–88. Springer, 2022.

[46] Liang Peng, Junkai Xu, Haoran Cheng, Zheng Yang, Xiaopei Wu, Wei Qian, Wenxiao Wang, Boxi Wu, and Deng Cai. Learning occupancy for monocular 3d object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[47] Zengyi Qin, Jinglu Wang, and Yan Lu. Monogrnet: A general framework for monocular 3d object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5170–5184, 2021.

[48] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8555–8564, 2021.

[49] 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.

[50] Xuepeng Shi, Qi Ye, Xiaozhi Chen, Chuangrong Chen, Zhixiang Chen, and Tae-Kyun Kim. Geometry-based distance decomposition for monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15172–15181, 2021.

[51] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1991–1999, 2019.

[52] Yongzhi Su, Yan Di, Guangyao Zhai, Fabian Manhardt, Jason Rambach, Benjamin Busam, Didier Stricker, and Federico Tombari. Opa-3d: Occlusion-aware pixel-wise aggregation for monocular 3d object detection. *IEEE Robotics and Automation Letters*, 8(3):1327–1334, 2023.

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

[54] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 913–922, 2021.

[55] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019.

[56] Zizhang Wu, Yuanzhu Gan, Lei Wang, Guilian Chen, and Jian Pu. Monopgc: Monocular 3d object detection with pixel geometry contexts. *International Conference on Robotics and Automation*, 2023.

[57] Zizhang Wu, Yunzhe Wu, Jian Pu, Xianzhi Li, and Xiaoquan Wang. Attention-based depth distillation with 3d-aware positional encoding for monocular 3d object detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[58] Junkai Xu, Liang Peng, Haoran Cheng, Hao Li, Wei Qian, Ke Li, Wenxiao Wang, and Deng Cai. Mononerd: Nerf-like representations for monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6814–6824, 2023.

[59] Qiangeng Xu, Yiqi Zhong, and Ulrich Neumann. Behind the curtain: Learning occluded shapes for 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2893–2901, 2022.

[60] Longfei Yan, Pei Yan, Shengzhou Xiong, Xuanyu Xiang, and Yihua Tan. Monocd: Monocular 3d object detection with complementary depths. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[61] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17830–17839, 2023.

[62] Honghui Yang, Tong He, Jiaheng Liu, Hua Chen, Boxi Wu, Binbin Lin, Xiaofei He, and Wanli Ouyang. Gd-mae: generative decoder for mae pre-training on lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9403–9414, 2023.

[63] Hongdou Yao, Jun Chen, Zheng Wang, Xiao Wang, Pengfei Han, Xiaoyu Chai, and Yansheng Qiu. Occlusion-aware plane-constraints for monocular 3d object detection. *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[64] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11784–11793, 2021.

[65] Jingyi Zhang, Jiaxing Huang, Xueying Jiang, and Shijian Lu. Black-box unsupervised domain adaptation with bi-directional atkinson-shiffrin memory. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11771–11782, 2023.

[66] Renrui Zhang, Han Qiu, Tai Wang, Xuanzhuo Xu, Ziyu Guo, Yu Qiao, Peng Gao, and Hongsheng Li. Monodetr: Depth-aware transformer for monocular 3d object detection. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[67] Renrui Zhang, Liuhui Wang, Yu Qiao, Peng Gao, and Hongsheng Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21769–21780, 2023.

[68] 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*, pages 3289–3298, 2021.

[69] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.

[70] Yunsong Zhou, Yuan He, Hongzi Zhu, Cheng Wang, Hongyang Li, and Qinhong Jiang. Monoef: Extrinsic parameter free monocular 3d object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):10114–10128, 2021.

[71] Yunsong Zhou, Hongzi Zhu, Quan Liu, Shan Chang, and Minyi Guo. Monoatt: Online monocular 3d object detection with adaptive token transformer. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[72] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *International Conference on Learning Representations*, 2021.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The contributions are summarized at the end of the introduction section to accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The discussion of the limitations of the work is presented in Conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when the image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The results provided in this paper are not theoretical, since the results are practical results tested on datasets.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.

- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The datasets used for experiments and the implementation details are introduced in Section 4.1 to ensure the reproducibility of this work. Moreover, a detailed introduction to the architecture of the proposed approach is presented in Section 3 of the paper and Section C.1 of the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The used datasets are publicly available. We will consider releasing the code upon acceptance.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: They are provided in the implementation details in Section 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Following previous papers in the same field, the statistical significance is not provided.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The information on the computer resources is provided in Section 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper conforms to the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The original papers of the used datasets are cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.