MonoCLUE : Object-Aware Clustering Enhances Monocular 3D Object Detection

Sunghun Yang, Minhyeok Lee, Jungho Lee, Sangyoun Lee

Yonsei University
Seoul, Republic of Korea
sunghun98@yonsei.ac.kr, hydragon516@yonsei.ac.kr, 2015142131@yonsei.ac.kr, syleee@yonsei.ac.kr

Abstract

Monocular 3D object detection offers a cost-effective solution for autonomous driving but suffers from ill-posed depth and limited field of view. These constraints cause a lack of geometric cues and reduced accuracy in occluded or truncated scenes. While recent approaches incorporate additional depth information to address geometric ambiguity, they overlook the visual cues crucial for robust recognition. We propose MonoCLUE, which enhances monocular 3D detection by leveraging both local clustering and generalized scene memory of visual features. First, we perform K-means clustering on visual features to capture distinct object-level appearance parts (e.g., bonnet, car roof), improving detection of partially visible objects. The clustered features are propagated across regions to capture objects with similar appearances. Second, we construct a generalized scene memory by aggregating clustered features across images, providing consistent representations that generalize across scenes. This improves object-level feature consistency, enabling stable detection across varying environments. Lastly, we integrate both local cluster features and generalized scene memory into object queries, guiding attention toward informative regions. Exploiting a unified local clustering and generalized scene memory strategy, MonoCLUE enables robust monocular 3D detection under occlusion and limited visibility, achieving state-of-the-art performance on the KITTI benchmark.

Code — https://github.com/SungHunYang/MonoCLUE

1 Introduction

3D object detection is a cornerstone of autonomous driving, enabling the estimation of the location, size, and depth of objects in a scene. To this end, LiDAR-based, multi-view, and monocular approaches have emerged in 3D object detection research. In particular, the simplicity of using a single image and its cost-effectiveness have made the monocular approach a topic of significant current interest. However, the lack of viewpoint disparity in monocular images leads to a loss of relative geometric cues. As a result, the model faces difficulty in projecting 2D bounding boxes into accurate 3D positions referred to as the ill-posed depth problem. Moreover, relying on a single image limits the observable field of view. Without alternative viewpoints, the model is required to infer occluded objects based solely on partial observations, which leads to reduced prediction accuracy.

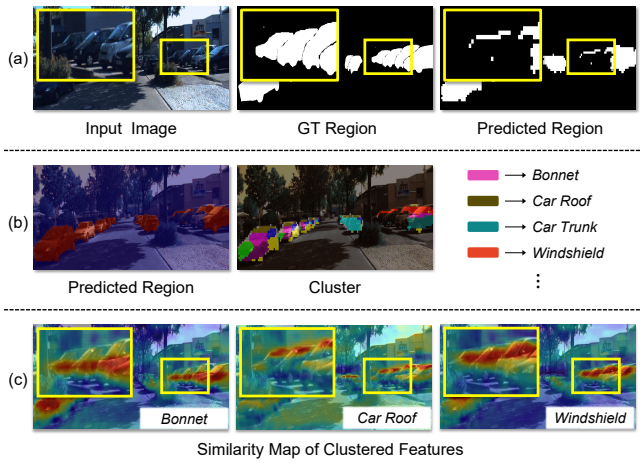

Figure 1: (a) Comparison between predicted and ground-truth regions, showing prediction errors on occluded objects. (b) Local clustering results, revealing that clusters reflect object parts and orientation, even in incomplete segmentation. (c) Activation maps of cluster features propagated across the image, capturing visually similar regions beyond initial segmentation.

To address the ill-posed depth problem, previous works (Zhang et al. 2023; Pu et al. 2025; Yan et al. 2024) have explored the use of depth cues to enhance geometric reasoning. MonoDETR (Zhang et al. 2023) introduces a foreground depth map to provide effective depth cues and improve geometric performance. In addition, MonoDGP (Pu et al. 2025) enhances depth accuracy by incorporating geometric depth differences. These approaches enrich 3D-aware representations and mitigate geometric ambiguities using enhanced depth map.

However, these methods overlook the importance of visual cues that are fundamental to object detection. The aforementioned issue of limited observable field of view has not been the focus of previous methods. In monocular settings, key factors such as object center, spatial position, and orientation must be inferred solely from appearance. This becomes especially problematic in cases of occlusion, trunca-

tion, or complex scenes with overlapping objects. In such cases, relying on depth is insufficient to separate instances or capture their complete shape. As a result, depth-focused strategy monocular object detection remains highly challenging under these conditions. Therefore, it is crucial to emphasize diverse object-level appearance cues, which overlooked in prior works, to support reliable detection in such cases.

To solve these problems, we propose MonoCLUE. First, we apply K-means clustering (Hartigan and Wong 1979) to regions of the feature map corresponding to object locations. This process aims to separate object-level appearance patterns and is referred to as local clustering. To guide this process, we utilize object segment masks generated by SAM (Kirillov et al. 2023), which ensure that clustering is performed within object regions. Consequently, we encourage the clustered features to represent diverse object-level visual patterns(e.g., bonnet, car roof) through training. This enables the detector to reinforce similar instances and encode object features more effectively. This method is robust to hard samples where objects are only partially visible. As shown in Figure 1, similar appearances are captured when the clustered features from the segmented region are propagated across the entire scene. Second, clustering within a single image lacks the ability to capture consistent visual patterns across scenes. To complement this, we aggregate local cluster features from multiple scenes to construct a generalized scene memory that encodes commonly occurring appearance cues. This memory provides common visual patterns by reducing sensitivity to image-specific variability. Moreover, it serves as a stable reference, supporting reliable predictions when local cluster features are insufficient or ambiguous. As a result, this process enhances the overall stability of detection performance, especially in less complex scenes. Lastly, MonoCLUE adopts the DETR (Carion et al. 2020) architecture for object detection. It further enhances the framework by integrating local cluster features and generalized scene memory into the object queries, improving object-level reasoning. This guidance helps the queries focus on relevant regions and capture more informative object-level features. Consequently, the combined effect of local clustering and generalized scene memory leads to more robust object understanding in monocular settings.

We demonstrate through various ablation studies that our model achieves both efficiency and strong performance. This results in state-of-the-art performance in monocular settings on the KITTI benchmark (Geiger, Lenz, and Urtasun 2012).

Our main contributions are summarized as follows:

- We cluster local regions to capture diverse visual cues, enabling robust detection of partially visible instances and improving performance on hard samples.
- We construct a generalized scene memory from aggregated local cluster features, providing consistent appearance patterns that enhance generalization across scenes.
- We integrate local cluster features and generalized scene memory to the query to enhance object decoding. This leads to state-of-the-art performance on the KITTI dataset.

2 Related Work

2.1 Muti-View 3D object detection

Multi-view 3D object detection leverages images from multiple cameras to estimate object locations and shapes in 3D space. DETR3D (Wang et al. 2022) introduces 3D object queries projected onto multi-view images to aggregate relevant features. BEVFormer (Li et al. 2022b) uses learnable BEV queries and a spatiotemporal transformer for efficient multi-view feature aggregation. Subsequent studies enhance performance with techniques like cross-modal distillation (Huang et al. 2022b). These methods benefit from diverse viewpoints and combine geometric and appearance cues for strong performance. In contrast, monocular methods lack spatial diversity and show lower accuracy. To address this, we extract rich visual signals from a single image. MonoCLUE applies object-level clustering to enhance monocular features and strengthen object-level priors without relying on multi-view cues.

2.2 Monocular 3D object detection

Monocular 3D object detection estimates 3D location, dimensions, and orientation of objects from a single RGB image. Compared to multi-view–based methods, it offers a cost-efficient alternative without additional sensors. CNNs have been widely used for extracting local features and building spatial context (Li et al. 2022a; Liu, Wu, and Tóth 2020). Many methods extend 2D detectors (Wang et al. 2021b) and incorporate geometric constraints or auxiliary supervision, such as LiDAR and depth maps (Ma et al. 2019; Reading et al. 2021; Wang et al. 2021a), to compensate for missing depth. Recently, Transformer-based models (Dosovitskiy et al. 2020) have gained attention for capturing long-range dependencies and global context. DETR-style frameworks (Carion et al. 2020) treat detection as set prediction using object queries. MonoDETR (Zhang et al. 2023) introduces depth-aware queries, while MonoDGP (Pu et al. 2025) improves context with segment embeddings and decoupled 2D–3D decoding.

MonoDGP uses segment embeddings to enhance context but overlooks regions outside masks and lacks feature diversity. We replace limited cues with clustering-based features to build robust representations that address monocular limitations. Our model retains query-based decoding and geometric reasoning, while enhancing appearance cues via local clustering and scene memory to improve detection.

2.3 K-means Cluster

K-means (Hartigan and Wong 1979) has been widely used in vision tasks to group features with similar patterns or semantics (Guo et al. 2017; Caron et al. 2018). In object detection, it captures underlying structures within feature representations, facilitating region grouping and instance understanding. Prior works apply clustering to discover object parts (Wang et al. 2020) or for representation learning and prototype generation (Caron et al. 2020). Inspired by this, we leverage K-means to extract diverse object-level features and enhance visual reasoning in monocular 3D detection.

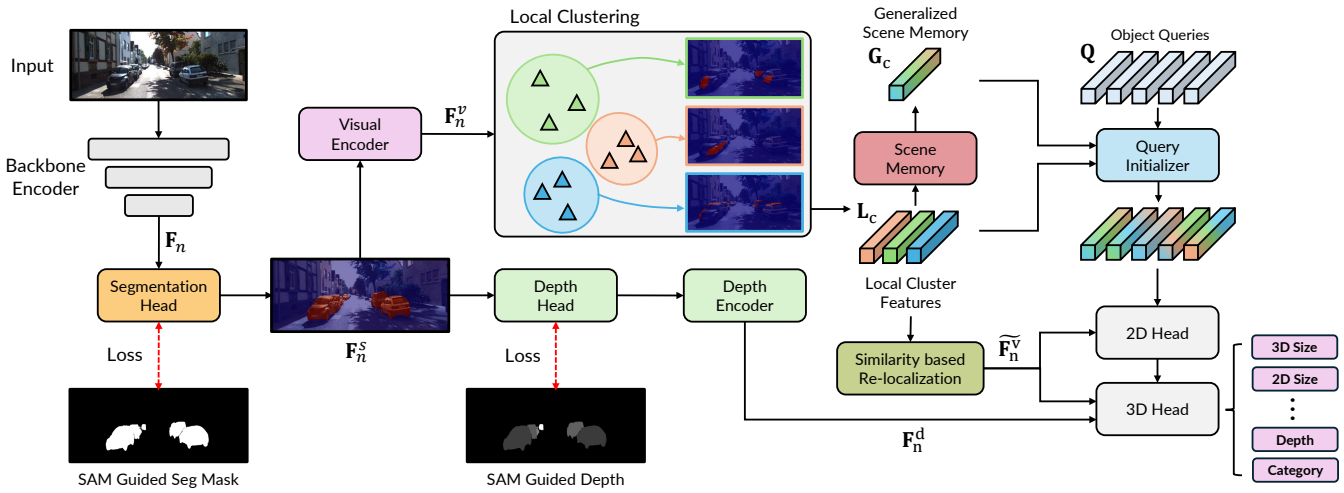

Figure 2: Overall architecture of the proposed MonoCLUE. Our core components are local clustering, similarity based re-localization, and query initialization. We perform clustering on the visual encoder features to extract local cluster features from specific regions. The local cluster features are then used for re-localization, generalized scene memory, and query initialization.

3 Methods

3.1 Overall Architecture

Figure 2 presents the overall architecture of MonoCLUE. A backbone encoder extracts multi-scale features $\mathbf{F}_n \in \mathbb{R}^{C \times \frac{H}{n} \times \frac{W}{n}}$, where $n \in \{2^3, 2^4, 2^5, 2^6\}$. Each $\mathbf{F}_n$ is processed by the region segmentation head. Its output is supervised using SAM (Kirillov et al. 2023)-guided segmentation masks. The seg-embedded features $\mathbf{F}_n^s$ produced by the region segmentation head are subsequently passed through both the visual and the depth encoder, following the structure of MonoDETR (Zhang et al. 2023).

Firstly, we obtain visual features $\mathbf{F}_n^v$, which are the output of the visual encoder. Afterward, these features are clustered by K-means clustering within object-region segmentation mask as shown in Figure 2. We define each clustered region, which captures diverse visual cues, as local cluster features $\mathbf{L}_c \in \mathbb{R}^{N_l \times C}$, where N_l is the number of clusters. To enhance region identification, we additionally perform similarity-based re-localization. This involves identifying regions that show high similarity with the $\mathbf{L}_c$. Such a process is utilized to assist the model in discovering object-like areas, including partially visible objects. Secondly, we gather all $\mathbf{L}_c$ and store them in a generalized scene memory $\mathbf{G}_c \in \mathbb{R}^{N_g \times C}$, where N_g represents the number of generalized scene memories. These memories, which are learned dataset-wide representations, capture common appearance patterns. Lastly, we use object queries $\mathbf{Q} \in \mathbb{R}^{N_q \times C}$, where N_q denotes the number of queries, to decode the learned information. $\mathbf{Q}$ is directly initialized with both $\mathbf{L}_c$ and $\mathbf{G}_c$ before decoding. This initialization allows $\mathbf{Q}$ to embed object-aware features in advance. Then, following the structure of MonoDGP (Pu et al. 2025), we decode initialized $\mathbf{Q}$ with separate 2D and 3D heads to perform 3D object detection. The detailed descriptions of each component are provided in the following sections.

3.2 Local Clustering

Figure 3(a) shows the process of our local clustering. The visual encoder feature $\mathbf{F}_n^v$ contains information required for both object classification and box regression. Especially in 3D object detection, since visual characteristics change depending on the orientation and depth of an object, $\mathbf{F}_n^v$ inevitably includes 2D and 3D-aware information. To make these representations more distinguishable, we cluster $\mathbf{F}_n^v$ to explicitly separate and emphasize distinct visual cues.

To this end, we replace the box-shaped masks used in MonoDGP with object-shaped masks to supervise the segmentation head. This allows clustering to focus exclusively on object regions. As a result, the quality of the clusters is significantly enhanced by eliminating background noise, leading to improved discrimination. Based on this strategy, we apply K-means clustering to $\mathbf{F}_n^v$ exclusively within the segmentation mask $M_n \in \mathbb{R}^{\frac{H}{n} \times \frac{W}{n}}$ predicted from the region segmentation head. We then apply masked average pooling to each of the N_l clusters to obtain the local cluster features $\mathbf{L}_c$. This process is expressed as follows:

$$\mathbf{L}_c^{(k)} = \frac{\sum_{i,j} M_n^{(k)}(i,j) \cdot \mathbf{F}_n^v(i,j)}{\sum_{i,j} M_n^{(k)}(i,j)} \quad \text{for } k = 1, \ldots, N_l, \tag{1}$$

where (i, j) are the pixel coordinates. Consequently, $\mathbf{L}_c$ exhibits enhanced reliability in specific regions of partially visible objects, as the separation of features enables accurate discrimination even under partial visibility conditions

3.3 Generalized Scene Memory

Since $\mathbf{L}_c$ comes from single-image clustering, it may lack generalization capability. To complement such image-specific features, it is necessary to capture general object priors. Therefore, we introduce a generalized scene memory. Figure 3(b) illustrates the structure of the generalized scene memory procedure. The generalized scene memories

$\mathbf{G}_c$ store common and recurring object features across the dataset. By collecting image-specific $\mathbf{L}_c$ from all images and extracting shared features, we obtain a generalized representation for the dataset.

To this end, we first create N_g embedding vectors as memory. We then incorporate $\mathbf{L}_c$ into them using a cross-attention mechanism (Vaswani et al. 2017), where the memory vectors $\mathbf{G}_c$ serve as query. We apply this process at every training iteration to progressively update the memory, encouraging $\mathbf{G}_c$ to store useful features that are commonly shared across the dataset. Cross-attention enables $\mathbf{G}_c$ to aggregate diverse $\mathbf{L}_c$ into general object-level cues in a balanced manner. In training stage, to ensure consistency across batches, the same memories $\mathbf{G}_c$ are shared for all inputs. Therefore, we flatten the batch dimension of $\mathbf{L}_c$, resulting in $\tilde{\mathbf{L}}_c \in \mathbb{R}^{(B\times N_l)\times C}$, and use it as the key and value in the cross-attention. This process is expressed as follows:

$$\mathbf{G}_c = \text{softmax}\left(\frac{w_q\mathbf{G}_c(w_k\tilde{\mathbf{L}}_c)^\top}{\sqrt{C}}\right)(w_v\tilde{\mathbf{L}}_c) + w_q\mathbf{G}_c, \quad (2)$$

where w_q,w_k and w_v are learnable projection matrices of query, key and value. By integrating a dataset-wide shared features into $\mathbf{G}_c$, the model obtains a generalized object representation that reflects common appearance patterns. These generalized priors remain effective even in unseen scenes and contribute to stabilizing predictions. In addition, $\mathbf{G}_c$ perform robustly on easy objects that resemble frequently seen prototypes during training.

3.4 Similarity-based Re-localization

In general, the segmentation head tends to produce inaccurate masks in cases of occlusion or small objects due to insufficient visual cues. Such errors degrade the final 3D object detection performance.

To address this, we utilize the $\mathbf{L}_c$ to re-localize and refine object regions based on high similarity. This approach enables broader discovery of objects that exhibit sparse visual cues. First, as shown in Figure 4(a), we calculate pixel-wise cosine similarity scores for $\mathbf{F}_n^v$ against all N_l instances of $\mathbf{L}_c$. Therefore, the resulting N_l local similarity maps has dimensions of $N_l \times \frac{H}{n} \times \frac{W}{n}$. Next, we take the maximum value along the N_l dimension of this local similarity maps to generate final similarity map $\mathbf{S} \in \mathbb{R}^{\frac{H}{n}\times\frac{W}{n}}$. This final $\mathbf{S}$ identifies object-like candidate regions based on high similarity to any of the N_l clustered features. This process is expressed as follows:

$$S(i,j) = \max_{N_l}\left(\frac{\mathbf{L}_c \cdot \mathbf{F}_n^v(i,j)}{\|\mathbf{L}_c\|\,\|\mathbf{F}_n^v(i,j)\|}\right). \quad (3)$$

The $\mathbf{S}$ helps re-localize the candidate positions of objects that are not detected due to insufficient visual cues. Second, the $\mathbf{S}$ is concatenated with $\mathbf{F}_n^v$ to inject candidate object location cues into the features, guiding the model to focus on likely object regions. In this method, we employ the Multiscale Deformable attention (MsDeform) (Zhu et al. 2020) for memory efficiency, similar to existing methods. As a result, we generate refined features $\tilde{\mathbf{F}}_n^v$, which enhances the model's localization performance. Lastly, we guide MS-Deform attention to focus on object-centric regions, which

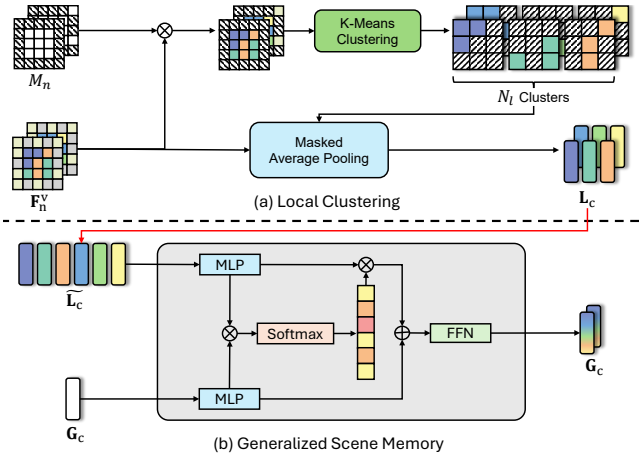

Figure 3: The process of local clustering and generalized scene memory. Each feature corresponds to a monocular image. (a) Local cluster features are obtained by independently extracting N_l features from the masked object regions. (b) Generalized scene memory integrates $\mathbf{L}_c$ from multiple images into a shared representation.

helps prevent attention weights from shifting toward the background due to randomly initialized offsets. To achieve this, we initialize reference point offsets using the similarity map $\mathbf{S}$, which serves as an auxiliary guide. In this process, the softmax correlation map is interpreted as a probability mass over the reference lattice r (Luvizon, Tabia, and Picard 2019). A weighted average of grid locations estimates the object center c, and subtracting r from c. This yields coarse offsets that bias sampling toward object-centric regions. This process is expressed as follows:

$$c = \sum_{i,j} \text{softmax}(\mathbf{S}(i,j)) \cdot r(i,j), \quad (4)$$

$$\Delta(i,j) = c - r(i,j). \quad (5)$$

The detailed MS-Deform attention weight results are provided in the supplementary material.

3.5 Query Initializer

Following baseline (Pu et al. 2025), we use object queries $\mathbf{Q}$ for decoding and process them through a 2D and a 3D head. These queries receive information from the refined visual feature $\tilde{\mathbf{F}}_n^v$ and the depth encoder output $\mathbf{F}_n^d$ via cross-attention, which assigns higher weights to more similar features during aggregation.

To guide this process more effectively, we pre-inject the diverse visual cues from $\mathbf{L}_c$ and $\mathbf{G}_c$ into the $\mathbf{Q}$. This initialization offers object-aware priors, allowing $\mathbf{Q}$ to better capture visual representations relevant to the target object. Since decoding requires not only object-level understanding but also contextual awareness, background features become essential. These features provide complementary cues that help refine object boundaries and improve 3D localization. To leverage this, we cluster the background excluding the object masks $(1 - M_n)$ into N_b groups to obtain background

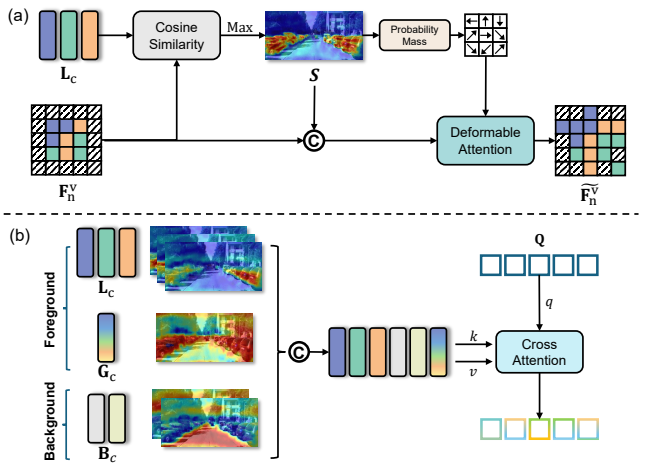

Figure 4: The process of re-localization and query initialization. (a) Local cluster similarity is used to compute **S**, which re-localizes features to object-like regions. (b) Query initialization using $\mathbf{L}_c$, $\mathbf{G}_c$, and $\mathbf{B}_c$.

feature $\mathbf{B}_c \in \mathbb{R}^{N_b \times C}$ using the same method as for the local clustering. As noted in (Yang et al. 2024, 2023), depth is closely related to ground surface, incorporating these cues contributes to accurate 3d object decoding. Lastly, as shown in Figure 4(b), we incorporate $\mathbf{B}_c$ alongside $\mathbf{L}_c$ and $\mathbf{G}_c$ during query initialization. Since these features already summarize meaningful object and scene-level information, there is no need to attend to the entire spatial feature map. Instead of applying cross-attention directly to the full feature map with spatial dimensions of $\frac{H}{n} \times \frac{W}{n}$ per level, we use a compact set of representative features. Specifically, we select $N_l + N_g + N_b$ features that effectively summarize the representation. This reduces memory consumption and enables more efficient attention computation.

As a result, embedding both object-aware and context-aware cluster information into **Q** leads to more robust and accurate object predictions.

3.6 Loss function

We follow the loss formulation of MonoDGP, which includes region segmentation loss $\mathcal{L}_{\text{region}}$, depth map regression loss $\mathcal{L}_{\text{depth}}$, 2D detection loss $\mathcal{L}_{\text{2D}}$, and 3D object estimation loss $\mathcal{L}_{\text{3D}}$. Here, $\mathcal{L}_{2D}$ covers classification, 2D bounding box regression, GIoU, and projected center losses. While, $\mathcal{L}_{3D}$ handles 3D size, orientation, and center depth. The total loss is expressed as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{2D}} + \mathcal{L}_{\text{3D}} + \lambda \mathcal{L}_{\text{depth}} + \lambda \sum_{i=0}^{4} \mathcal{L}^{i}_{\text{region}}, \quad (6)$$

where λ controls the depth and region segmentation loss weights.

4 Experiments

4.1 Dataset

KITTI We conduct experiments on the widely used KITTI benchmark (Geiger, Lenz, and Urtasun 2012), which is a standard dataset for 3D object detection in autonomous driving. It consists of 7,481 training and 7,518 test images, with annotations for three object categories, namely Car, Pedestrian, and Cyclist. Each object instance is assigned one of three difficulty levels Easy, Moderate and Hard based on factors such as occlusion and truncation. Following common practice (Chen et al. 2015), we split the training set into 3,712 training and 3,769 validation images for ablation and comparison. We report performance using Average Precision AP metrics for both 3D bounding boxes AP_{3D} and bird's eye view AP_{BEV}, evaluated at 40 recall positions for each difficulty level.

4.2 Implementation Details

We adopt ResNet-50 (He et al. 2016) as the backbone to extract multi-scale features. Following DETR-based designs, we use 50 object queries represented as learnable embeddings, and each attention module employs 8 heads. For memory efficiency, we apply multi-scale deformable attention (Zhu et al. 2020) in both the visual encoder and re-localization, using 4 sampling points. For depth prediction, the object-shaped depth range (0–60m) is uniformly quantized into 80 bins. To accelerate clustering, we use a CUDA-implemented K-means (Geiger, Lenz, and Urtasun 2012) algorithm for efficient GPU computation. We set N_l, N_g, and N_b to 10, the number of classes, and 3, respectively. Our model is trained on a single RTX 3090 GPU for 250 epochs with batch size 8, using AdamW (Loshchilov and Hutter 2017) with initial learning rate 2×10^{-4} and step decay schedule. During inference, queries with confidence below 0.2 are filtered, and no post-processing like NMS is applied.

4.3 Results on KITTI

We evaluate MonoCLUE on the KITTI benchmark and compare it with recent monocular 3D detection methods. As shown in Table 1, MonoCLUE achieves the best performance across all difficulty levels on both the test and validation sets for the car category, except for the hard case on the test set. This is achieved without using any extra information like depth or LiDAR. On the test set, it achieves 27.94 and 19.70 AP_{3D} for the easy and moderate cases, outperforming previous state-of-the-art methods by +1.59% and +0.86%, respectively. On the validation set, it further improves performance by +2.98% (easy) and +1.76% (moderate). These gains are attributed to our structured clustering approach and generalized memory, which together enhance object-level discrimination and improve localization accuracy. Table 1 highlights the effectiveness of MonoCLUE in capturing discriminative visual cues under monocular settings. Additional experiments on the KITTI dataset are included in the supplementary material.

4.4 Results on Other Categories

Table 2 shows that MonoCLUE achieves the best performance for Pedestrian and the second-best for Cyclist on the KITTI test set. This indicates the effectiveness of our method across other object types. Furthermore, both the detection and orientation metrics show substantial improvements over previous methods. These results show that our

Methods	Extra	Test						Validation					
		$AP_{BEV\|R40}$			$AP_{3D\|R40}$			$AP_{BEV\|R40}$			$AP_{3D\|R40}$		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
MonoPGC (Wu et al. 2023)	Depth	32.50	23.14	20.30	24.68	17.17	14.14	34.06	24.26	20.78	25.67	18.63	15.65
OPA-3D (Su et al. 2023)	Depth	33.54	22.53	19.22	24.60	17.05	14.25	33.80	25.51	22.13	24.97	19.40	16.59
MonoDTR (Huang et al. 2022a)	LiDAR	28.59	20.38	17.14	21.99	15.39	12.73	33.33	25.35	21.68	24.52	18.57	15.51
DID-M3D (Peng et al. 2022)	LiDAR	32.95	22.76	19.83	24.40	16.29	13.75	31.10	22.76	19.50	22.98	16.12	14.03
OccupancyM3D (Peng et al. 2024)	LiDAR	35.38	24.18	21.37	25.55	17.02	14.79	35.72	26.60	23.68	26.87	19.96	17.15
GUPNet (Lu et al. 2021)	-	30.29	21.19	18.20	22.26	15.02	13.12	31.07	22.94	19.75	22.76	16.46	13.72
MonoCon (Liu, Xue, and Wu 2022)	-	31.12	22.10	19.00	22.50	16.46	13.95	-	-	-	26.33	19.01	15.98
DEVIANT (Kumar et al. 2022)	-	29.65	20.44	17.43	21.88	14.46	11.89	32.60	23.04	19.99	24.63	16.54	14.52
MonoDDE (Li et al. 2022a)	-	33.58	23.46	20.37	24.93	17.14	15.10	35.51	26.48	23.07	26.66	19.75	16.72
MonoUNI (Jinrang, Li, and Shi 2023)	-	33.28	23.05	19.39	24.75	16.73	13.49	-	-	-	24.51	17.18	14.01
MonoDETR (Zhang et al. 2023)	-	33.60	22.11	18.60	25.00	16.47	13.58	37.86	26.95	22.80	28.84	20.61	16.38
FD3D (Wu et al. 2024)	-	34.20	23.72	20.76	25.38	17.12	14.50	36.98	26.77	23.16	28.22	20.23	17.04
MonoMAE (Jiang et al. 2024)	-	34.15	24.93	21.76	25.60	18.84	**16.78**	40.26	27.08	23.14	30.29	20.90	17.61
MonoCD (Yan et al. 2024)	-	33.41	22.81	19.57	25.53	16.59	14.53	34.60	24.96	21.51	26.45	19.37	16.38
MonoDGP (Pu et al. 2025)	-	35.24	25.23	22.02	26.35	18.72	15.97	39.40	28.20	24.42	30.76	22.34	19.02
MonoCLUE	-	**36.15**	**26.15**	**22.81**	**27.94**	**19.70**	16.69	**41.79**	**29.91**	**26.00**	**33.74**	**24.10**	**20.58**

Table 1: Comparisons with monocular methods on the KITTI validation and test for the car category. The best results are shown in bold, and the second-best are shown with underline.

Method	Detection	Orientation	Pedestrain $AP_{3D\|Mod.}$	Cyclist $AP_{3D\|Mod.}$
GUPNet	94.15	93.92	9.76	**3.21**
DEVIANT	94.42	94.01	8.65	3.13
MonoDGP	94.35	94.22	9.89	2.28
MonoCLUE	**95.82**	**95.54**	**10.45**	3.20

Table 2: Comparison of multi-category, 2D detection, and orientation on the test set. The detection and orientation results are evaluated on the car category.

enhanced visual cues contribute to improved 2D detection performance. Moreover, when geometric reasoning is incorporated in the 3D detection, these results further boost overall performance. In addition, since orientation in monocular settings relies heavily on visual cues, our diverse visual patterns prove effective in ensuring robustness to orientation.

4.5 Qualitative results

We compare the $3D$ and BEV detection performance of our method with two baseline models, MonoDETR and MonoDGP, on the KITTI validation set. As shown in Figure 5, MonoCLUE exhibits more reliable overall detection quality than the baselines. Specifically, our method demonstrates more robust detection on small and distant objects that are occluded. This improvement is attributed to the cluster features, which re-localize objects by matching them to easily detectable patterns across the image. This allows the model to infer missing object parts based on similar appearances observed in less challenging regions. Additionally, in the last example, a row of aligned cars shares similar orientation. This leads to similar appearances, which result in similar cluster features. Therefore, these similar cluster features lead to consistent detection in the BEV. As a result, MonoCLUE demonstrates robust performance in challenging scenarios such as occlusion, truncation, and overlapping objects. Additional activation map and qualitative results are provided in the supplementary material.

Architecture	$AP_{3D\|R40}$		
	Easy	Moderate	Hard
None	30.66	23.03	19.71
Codebook	31.77 (+1.11)	23.22 (+0.19)	19.75 (+0.04)
Cross attention	33.74 (+3.08)	24.10 (+1.07)	20.58 (+0.87)

Table 3: Comparison of generalized scene memory architecture on the validation set. Underlined score indicates the comparison without using the generalized scene memory.

SAM Guidance	Query Initializer	Similarity-based Re-localization	$AP_{3D\|R40}$		
			Easy	Moderate	Hard
-	-	-	29.61	22.06	18.75
✓	-	-	29.82	22.62	19.30
✓	✓	-	32.91	23.93	20.36
✓	-	✓	31.14	23.20	20.02
✓	✓	✓	**33.74**	**24.10**	**20.58**

Table 4: Comparison of core components on the validation set. Each result includes the checked component along with all other fixed components.

5 Ablation Analysis

5.1 Scene memory

Table 3 demonstrates the effectiveness of the proposed generalized scene memory. The results show that performance is lowest when the generalized scene memory is not used, and the results align with our intention, as the easy case shows the most significant improvement. Specifically, the gains are +3.08% and +1.11%, confirming that the cross-attention-based memory is the most effective. To explore optimal memory designs, we employ the codebook proposed

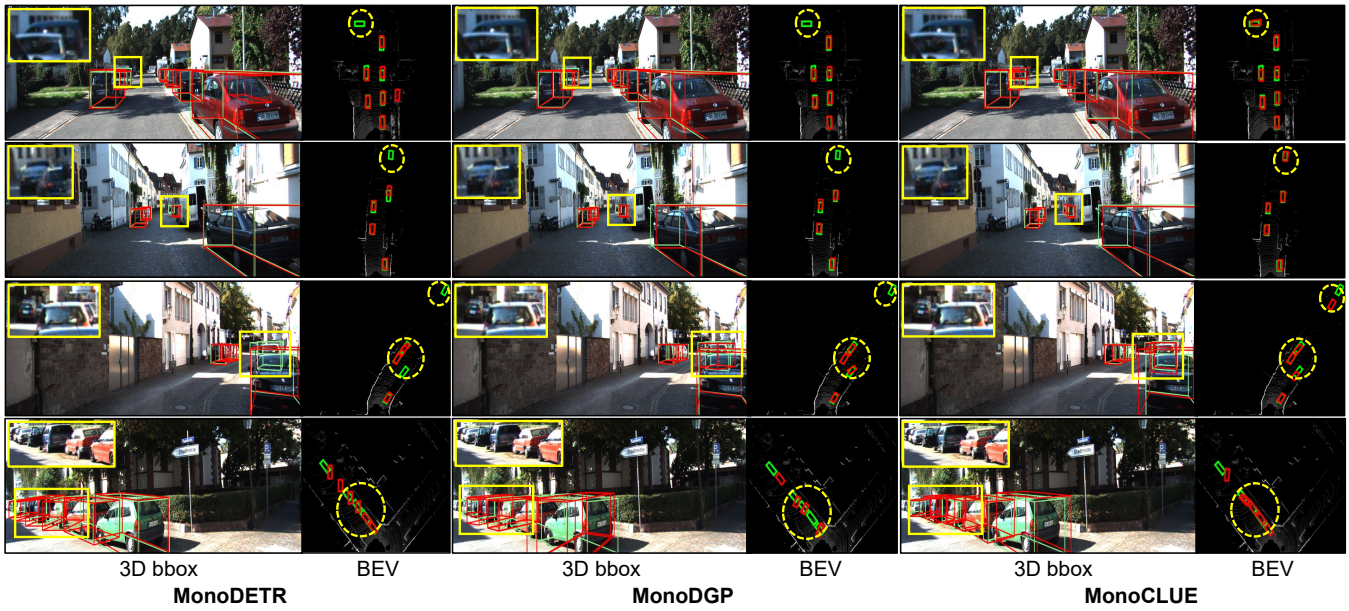

Figure 5: Qualitative comparison on the KITTI validation set. Ground-truth boxes (green) and predictions (red) are shown for both 3D bounding boxes and bird's-eye view (BEV).

Method	Params (M)	FLOPs ↓ (G)	$AP_{3D\|R40}$ ↑ (Moderate)	Runtime ↓ (ms)
MonoDETR	37.68	59.72	20.61	35
MonoDGP	42.16	68.99	22.34	42
MonoCLUE	44.17	72.71	24.10	52

Table 5: Comparison of computational complexity. FLOPs and Runtime are measured on a single RTX 3090 GPU with a batch size of 1.

in VQ-VAE (Van Den Oord, Vinyals et al. 2017), a commonly used memory structure. All memory types are configured with the same size for fair comparison. However, the codebook structure struggle to find representative features using only the loss as guidance. Additionally, since not all codebook vectors are updated during training, some memory slots remained unused, leading to lower performance. In contrast, the cross-attention structure applies weights to all memory entries and learns an common feature, resulting in the best performance.

5.2 Core components

Table 4 analyzes the contribution of each component in MonoCLUE. Removing SAM guidance corresponds to training with box-shaped masks, as in baseline methods (Pu et al. 2025). This hinders clustering by including background noise in object regions, which degrades performance. This highlights the importance of SAM for effective clustering. Re-localization improves occluded region representation by identifying candidate locations with similar appearances through local clustering. This leads to a performance gain of +0.7% in the hard case. The query initializer aggregates all clustering information, leading to consistent improvements across difficulty levels. Compared to using only SAM, it achieves gains of +3.9% in easy and +1.31% in moderate case, validating the effectiveness of combining background and generalized information. Finally, integrating all components yields the highest performance, demonstrating the complementary benefits of each component.

5.3 Efficiency and Performance Comparison

Table 5 presents a comparison of each method in terms of model complexity, computational complexity and moderate level performance AP_{3D} on the KITTI validation set. Compared to the baseline MonoDETR, MonoDGP improves the performance by +1.73% compared to our baseline, but at the cost of a significant increase in parameters 4.48M and FLOPs 9.27G. In contrast, our method MonoCLUE achieves a larger performance gain of +1.76%, while incurring a much smaller increase in parameters 2.01M and FLOPs 3.72G. This indicates that the clustering-based design enables performance improvements without incurring significant computational or parameter overhead. As a result, MonoCLUE achieves a better cost-performance tradeoff than previous state-of-the-art models.

6 Conclusion

We propose MonoCLUE, a monocular 3D object detection framework that improves object-level reasoning through localized clustering and scene priors. By clustering fine-grained features in object-shaped regions and generalizing them into a memory module, our method captures divers object patterns. Incorporating local, background, and scene-level features enables informative query initialization with strong contextual and geometric cues. Experiments on the KITTI benchmark validate the effectiveness of each component, with MonoCLUE achieving state-of-the-art performance among monocular methods.

Acknowledgements

This work was supported by the Korea Institute of Science and Technology (KIST) Institutional Program (Project No.2E33612-25-016) and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT)(No. RS-2024-00340745).

References

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, 213–229. Springer.

Caron, M.; Bojanowski, P.; Joulin, A.; and Douze, M. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, 132–149.

Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33: 9912–9924.

Chen, X.; Kundu, K.; Zhu, Y.; Berneshawi, A. G.; Ma, H.; Fidler, S.; and Urtasun, R. 2015. 3d object proposals for accurate object class detection. *Advances in neural information processing systems*, 28.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, 3354–3361. IEEE.

Guo, X.; Liu, X.; Zhu, E.; and Yin, J. 2017. Deep clustering with convolutional autoencoders. In *International conference on neural information processing*, 373–382. Springer.

Hartigan, J. A.; and Wong, M. A. 1979. Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1): 100–108.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Huang, K.-C.; Wu, T.-H.; Su, H.-T.; and Hsu, W. H. 2022a. Monodtr: Monocular 3d object detection with depth-aware transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4012–4021.

Huang, P.; Liu, L.; Zhang, R.; Zhang, S.; Xu, X.; Wang, B.; and Liu, G. 2022b. Tig-bev: Multi-view bev 3d object detection via target inner-geometry learning. *arXiv preprint arXiv:2212.13979*.

Jiang, X.; Jin, S.; Zhang, X.; Shao, L.; and Lu, S. 2024. MonoMAE: Enhancing monocular 3D detection through depth-aware masked autoencoders. *Advances in Neural Information Processing Systems*, 37: 11392–11411.

Jinrang, J.; Li, Z.; and Shi, Y. 2023. Monouni: A unified vehicle and infrastructure-side monocular 3d object detection network with sufficient depth clues. *Advances in Neural Information Processing Systems*, 36: 11703–11715.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4015–4026.

Kumar, A.; Brazil, G.; Corona, E.; Parchami, A.; and Liu, X. 2022. Deviant: Depth equivariant network for monocular 3d object detection. In *European Conference on Computer Vision*, 664–683. Springer.

Li, Z.; Qu, Z.; Zhou, Y.; Liu, J.; Wang, H.; and Jiang, L. 2022a. Diversity matters: Fully exploiting depth clues for reliable monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2791–2800.

Li, Z.; Wang, W.; Li, H.; Xie, E.; Sima, C.; Lu, T.; Qiao, Y.; and Dai, J. 2022b. BEVFormer: Learning Bird's-Eye-View Representation from Multi-Camera Images via Spatiotemporal Transformers. *arXiv preprint arXiv:2203.17270*.

Liu, X.; Xue, N.; and Wu, T. 2022. Learning auxiliary monocular contexts helps monocular 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 1810–1818.

Liu, Z.; Wu, Z.; and Tóth, R. 2020. Smoke: Single-stage monocular 3d object detection via keypoint estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 996–997.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Lu, Y.; Ma, X.; Yang, L.; Zhang, T.; Liu, Y.; Chu, Q.; Yan, J.; and Ouyang, W. 2021. Geometry uncertainty projection network for monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3111–3121.

Luvizon, D. C.; Tabia, H.; and Picard, D. 2019. Human pose regression by combining indirect part detection and contextual information. *Computers & Graphics*, 85: 15–22.

Ma, X.; Wang, Z.; Li, H.; Zhang, P.; Ouyang, W.; and Fan, X. 2019. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6851–6860.

Peng, L.; Wu, X.; Yang, Z.; Liu, H.; and Cai, D. 2022. Didm3d: Decoupling instance depth for monocular 3d object detection. In *European Conference on Computer Vision*, 71–88. Springer.

Peng, L.; Xu, J.; Cheng, H.; Yang, Z.; Wu, X.; Qian, W.; Wang, W.; Wu, B.; and Cai, D. 2024. Learning occupancy for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10281–10292.

Pu, F.; Wang, Y.; Deng, J.; and Yang, W. 2025. Monodgp: Monocular 3D object detection with decoupled-query and

geometry-error priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 6520–6530.

Reading, C.; Harakeh, A.; Chae, J.; and Waslander, S. L. 2021. Categorical depth distribution network for monocular 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8555–8564.

Su, Y.; Di, Y.; Zhai, G.; Manhardt, F.; Rambach, J.; Busam, B.; Stricker, D.; and Tombari, F. 2023. Opa-3d: Occlusion-aware pixel-wise aggregation for monocular 3d object detection. *IEEE Robotics and Automation Letters*, 8(3): 1327–1334.

Van Den Oord, A.; Vinyals, O.; et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, F.; Liu, H.; Guo, D.; and Fuchun, S. 2020. Unsupervised representation learning by invariance propagation. *Advances in Neural Information Processing Systems*, 33: 3510–3520.

Wang, L.; Du, L.; Ye, X.; Fu, Y.; Guo, G.; Xue, X.; Feng, J.; and Zhang, L. 2021a. Depth-conditioned dynamic message propagation for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 454–463.

Wang, T.; Zhu, X.; Pang, J.; and Lin, D. 2021b. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 913–922.

Wang, Y.; Guizilini, V. C.; Zhang, T.; Wang, Y.; Zhao, H.; and Solomon, J. 2022. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on robot learning*, 180–191. PMLR.

Wu, Z.; Gan, Y.; Wang, L.; Chen, G.; and Pu, J. 2023. Monopgc: Monocular 3d object detection with pixel geometry contexts. *arXiv preprint arXiv:2302.10549*.

Wu, Z.; Gan, Y.; Wu, Y.; Wang, R.; Wang, X.; and Pu, J. 2024. Fd3d: Exploiting foreground depth map for feature-supervised monocular 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 6189–6197.

Yan, L.; Yan, P.; Xiong, S.; Xiang, X.; and Tan, Y. 2024. Monocd: Monocular 3d object detection with complementary depths. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10248–10257.

Yang, L.; Zhang, X.; Yu, J.; Li, J.; Zhao, T.; Wang, L.; Huang, Y.; Zhang, C.; Wang, H.; and Li, Y. 2024. MonoGAE: Roadside monocular 3D object detection with ground-aware embeddings. *IEEE Transactions on Intelligent Transportation Systems*, 25(11): 17587–17601.

Yang, X.; Ma, Z.; Ji, Z.; and Ren, Z. 2023. Gedepth: Ground embedding for monocular depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 12719–12727.

Zhang, R.; Qiu, H.; Wang, T.; Guo, Z.; Cui, Z.; Qiao, Y.; Li, H.; and Gao, P. 2023. Monodetr: Depth-guided transformer for monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9155–9166.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2020. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.