Difficulty-Aware Label-Guided Denoising for Monocular 3D Object Detection

Soyul Lee[1*] **Seungmin Baek**[1*] **Dongbo Min**[1†]

[1]School of AI and Software, Ewha Womans University, South Korea
{230aig, min100kim, dbmin}@ewha.ac.kr

Abstract

Monocular 3D object detection is a cost-effective solution for applications like autonomous driving and robotics, but remains fundamentally ill-posed due to inherently ambiguous depth cues. Recent DETR-based methods attempt to mitigate this through global attention and auxiliary depth prediction, yet they still struggle with inaccurate depth estimates. Moreover, these methods often overlook instance-level detection difficulty, such as occlusion, distance, and truncation, leading to suboptimal detection performance. We propose MonoDLGD, a novel **D**ifficulty-Aware **L**abel-**G**uided **D**enoising framework that adaptively perturbs and reconstructs ground-truth labels based on detection uncertainty. Specifically, MonoDLGD applies stronger perturbations to easier instances and weaker ones into harder cases, and then reconstructs them to effectively provide explicit geometric supervision. By jointly optimizing label reconstruction and 3D object detection, MonoDLGD encourages geometry-aware representation learning and improves robustness to varying levels of object complexity. Extensive experiments on the KITTI benchmark demonstrate that MonoDLGD achieves state-of-the-art performance across all difficulty levels.

Code — https://github.com/lsy010857/MonoDLGD

Introduction

Monocular 3D object detection aims to estimate the 3D location, size, and orientation of objects using a single RGB image. Owing to its low cost, ease of deployment, and compatibility with high-resolution images, it has become an attractive solution for applications such as autonomous driving, robotics, and augmented reality. However, unlike LiDAR-based (Liu et al. 2024a; Wang et al. 2023; Lang et al. 2019) or stereo-based (Chen et al. 2022; Guo et al. 2021; Li, Chen, and Shen 2019) methods, monocular approaches inherently suffer from a lack of depth cues, rendering 3D geometry estimation fundamentally ill-posed.

Recent advances in monocular 3D object detection have been driven by the adaptation of Transformer-based architectures, particularly the DEtection TRansformer (DETR) (Carion et al. 2020), originally developed for 2D object detection. MonoDETR (Zhang et al. 2023) first introduces DETR into monocular 3D detection, moving beyond CenterNet-based approaches (Ma et al. 2021; Yang et al. 2022; Zhang, Lu, and Zhou 2021) that focus on local features. By leveraging the global attention mechanism of Transformers, it effectively captures spatial and depth relationships between objects. To compensate for the absence of explicit depth cues, MonoDETR (Zhang et al. 2023) and MonoDGP (Pu et al. 2025) incorporate auxiliary depth prediction heads, injecting geometric priors into the detection pipeline. However, since the depth estimation relies solely on a single image, these methods remain constrained by the ill-posed nature of monocular images, which limits the accuracy of depth prediction. As illustrated in Figure 1, this inherent limitation results in considerable errors in 3D object localization.

MonoMAE (Jiang et al. 2024) attempt to improve robustness to occlusion by masking and reconstructing object features based on their depth levels, thereby enabling better 3D representation learning for partially visible objects. While

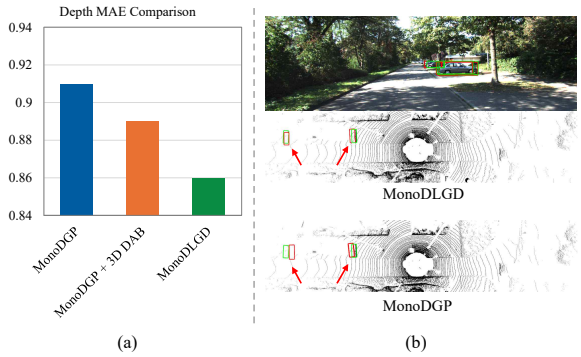


Figure 1: **Depth-Centric Detection: Elevating Depth and BEV Accuracy via MonoDLGD.** (a) Depth estimation accuracy using mean absolute error (MAE) on the KITTI validation set for MonoDGP (Pu et al. 2025), MonoDGP with our 3D-DAB, and ours. (b) Bird's-eye view (BEV) visualization. MonoDLGD achieves more accurate and robust detection across varying object distances, highlighting its improved geometric understanding.

*These authors contributed equally.
†Corresponding author.

effective to some extent, its difficulty modeling is limited to isolated factors such as occlusion status or depth range. In monocular settings, however, detection difficulty arises from a combination of factors including object scale, distance, truncation, and occlusion. Ignoring this multi-factor complexity can degrade both training stability and representation quality (Zhang, Lu, and Zhou 2021; Jiang et al. 2024).

To address the fundamental limitations of monocular 3D detection, we propose Difficulty-Aware Label-Guided Denoising (MonoDLGD), a novel framework that injects adaptive perturbations into ground-truth labels and learns to reconstruct them, providing explicit geometric supervision during training. Unlike prior DETR-based methods (Zhang et al. 2023; Pu et al. 2025; Jiang et al. 2024), our approach directly operates on 3D ground truth labels, enabling more stable and geometry-aware representation learning. MonoDLGD introduces two key components: (1) a denoising strategy that perturbs and reconstructs ground-truth labels containing rich 3D information, and (2) a difficulty-aware perturbation (DAP) mechanism that modulates the strength of perturbations based on instance-level detection difficulty. Together, these components guide the model to learn robust geometric representations across objects with diverse complexities, with only a marginal inference-time overhead.

Specifically, MonoDLGD perturb ground-truth labels such as projected bounding boxes and depths during training and learns to reconstruct them via a shared decoder, as illustrated in Fig. 2 (b). This denoising process provides strong supervision signals that help the model better understand 3D structure from monocular cues. To further enhance this effect, we introduce 3D Dynamic Anchor Box (3D-DAB), which embeds spatial priors (object projections and depths) into the queries, tightly aligning it with the perturbed label representations in the decoder. The reconstruction of these perturbed labels enables the decoder to transfer geometric signals into the detection pipeline more effectively.

Crucially, not all objects are equally difficult to detect in monocular settings. Small, distant, or occluded instances present greater ambiguity, and applying uniform perturbations may degrade their structural signals. To address this, MonoDLGD estimates instance-wise uncertainty as a proxy for detection difficulty and adaptively scales perturbation strength. Hard instances receive weaker perturbations to preserve their geometry, while easier ones are perturbed more aggressively. This difficulty-aware strategy promotes stable and discriminative feature learning across varying levels of complexity, ultimately improving 3D detection accuracy.

Our main contributions are as follows:

- We propose **Difficulty-Aware Label-Guided Denoising (MonoDLGD)**, which introduces label perturbation and reconstruction guided by prediction uncertainty, effectively leveraging explicit geometric supervision.
- We show that **modeling instance-level uncertainty** alone substantially improves detection accuracy, highlighting the importance of uncertainty-aware denoising in monocular 3D object detection.
- Our method achieves **state-of-the-art performance** on the KITTI benchmark without any additional inference overhead, as the difficulty-aware pertubation and reconstruction are confined to the training phase.

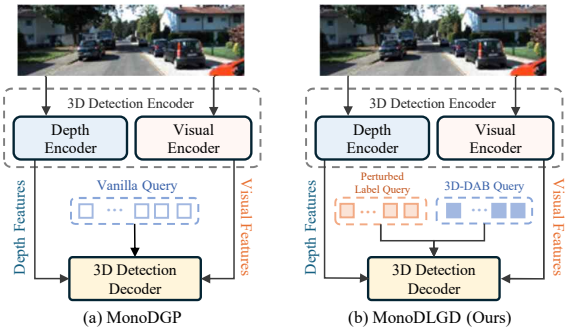


Figure 2: **Structural Comparison with MonoDGP (Pu et al. 2025).** Our method introduces 3D-DAB queries to encode spatial priors and explicitly provides 3D geometric supervision by reconstructing perturbed label queries within a shared decoder.

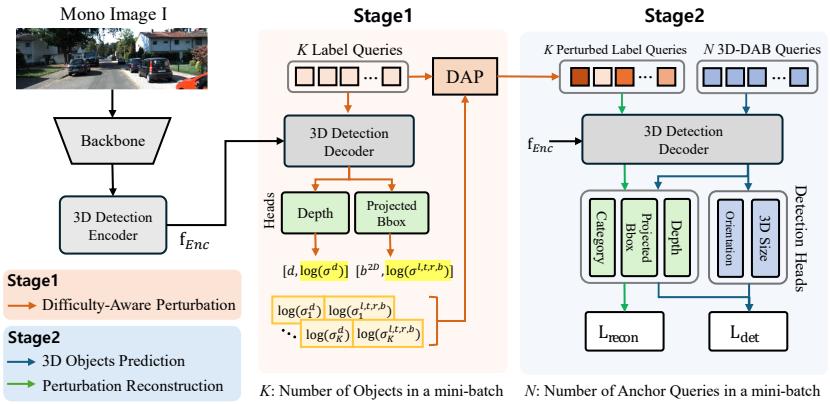
Related Work

Monocular 3D Detection with Transformers

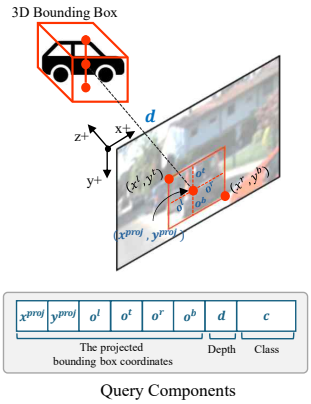
Recent advances in monocular 3D object detection have been largely driven by the adoption of Transformer architectures, owing to their superior ability to capture global context and long-range dependencies compared to CNN-based approaches (Ma et al. 2021; Yang et al. 2022; Zhang, Lu, and Zhou 2021). MonoDETR (Zhang et al. 2023) introduced the first DETR-style monocular 3D detector with a depth-guided transformer, using a predicted depth map and depth-aware decoder to incorporate global context. MonoDGP (Pu et al. 2025) further improved transformer detectors by decoupling 2D and 3D query streams and introducing perspective-invariant geometry error priors to refine depth estimation. Meanwhile, MonoMAE (Jiang et al. 2024) tackled occlusion via a depth-aware masked autoencoder: it masks parts of object queries and learns to reconstruct them, thereby handling heavily occluded objects. However, these methods inherently suffer from ill-posed geometric constraints, as depth estimation relies solely on a single image. Moreover, they tend to overlook instance-level challenges such as occlusion, distance, and truncation, leading to unstable training and degraded 3D representation quality.

Denoising Strategies for Object Detection

Denoising has recently emerged as an effective technique to stabilize training and enhance detection performance, especially within Transformer-based detection frameworks. DN-DETR (Li et al. 2022a) introduced denoising to DETR by perturbing ground-truth boxes and training the model to reconstruct them, significantly accelerating training convergence and reducing instability during bipartite matching. This approach has been further improved by DINO (Zhang et al. 2022), which introduced contrastive denoising by leveraging noisy queries to explicitly model positive and negative query pairs, further enhancing detection accuracy



(a) Overall architecture of MonoDLGD



(b) Query Component Structure

Figure 3: Overview of the proposed MonoDLGD: (a) MonoDLGD adopts a two-stage architecture after extracting the encoder feature f_{Enc} containing depth and visual features. **Stage 1** (red arrows) performs Difficulty-Aware Perturbation (DAP) by first estimating the uncertainty of bounding box (σ^l, σ^t, σ^r, σ^b) and depth (σ^d) attributes and then adaptively perturbing the label queries based on the estimated uncertainties. **Stage 2** (blue and green arrows) feeds both the perturbed label queries and the 3D-DAB queries into the decoder. (b) illustrates the internal components of label queries and 3D-DAB queries, all of which share a common structure.

and training stability. Denoising techniques have also been extended to 3D detection. ConQueR (Zhu et al. 2023) applies denoising to LiDAR-based detectors by perturbing and reconstructing queries in the voxel space to achieve sparse predictions and SEED (Liu et al. 2024b) adopts denoising training in point cloud-based DETR frameworks, enhancing detection accuracy and robustness for point cloud data. In this paper, we leverage a denoising strategy to explicitly incorporate geometric supervision. Unlike previous methods that apply uniform denoising, our approach directly injects adaptive perturbations into ground-truth labels based on instance-level detection difficulty, enabling the model to stably learn robust 3D representations.

Uncertainty Estimation

(Kendall and Gal 2017) formally categorize uncertainty into aleatoric uncertainty, which captures noise inherent in observations, and epistemic uncertainty, which accounts for uncertainty in model parameters. The aleatoric uncertainty has been actively explored in the context of object detection (Choi et al. 2019; Chen et al. 2020; He et al. 2019; Zhang, Lu, and Zhou 2021). For example, Gaussian YOLOv3 (Choi et al. 2019) models bounding boxes as Gaussian distributions to quantify localization uncertainty, which helps to rectify detection scores and reduce false positives. (He et al. 2019) predict bounding boxes as Gaussian distributions and utilize KL divergence as the regression loss, explicitly accounting for aleatoric uncertainty to achieve accurate object detection. MonoPair (Chen et al. 2020) leverages uncertainty to weight pair-wise geometric constraints during post-processing optimization, effectively improving detection stability and accuracy. MonoFlex (Zhang, Lu, and Zhou 2021) further models uncertainties of depth estimates from multiple depth predictors, leveraging aleatoric uncertainty to adaptively combine depth predictions from direct regression and keypoint-based geometry, significantly improving localization accuracy. In this paper, we leverage aleatoric uncertainty to enhance both training stability and representation quality in monocular 3D object detection. We incorporate this uncertainty into the denoising process for both 2D bounding box and depth supervision and use it to guide the adaptive perturbation strength in our difficulty-aware label-guided denoising framework.

Method

Motivation and Overview

Existing DETR-based monocular 3D object detectors, such as MonoDETR (Zhang et al. 2023) and MonoDGP (Pu et al. 2025), leverage auxiliary foreground depth maps to alleviate depth ambiguity but still fundamentally suffer from the ill-posed nature of monocular geometry. These methods also uniformly treat all objects during training, overlooking key difficulty factors such as object size, distance, occlusion, and truncation. MonoMAE (Jiang et al. 2024) partially addresses these issues via occlusion-aware masking, but its consideration of detection difficulty remains limited to occlusion alone, neglecting other critical complexity factors.

We propose MonoDLGD, a novel framework that leverages rich geometric information from ground-truth labels and explicitly models instance-level detection difficulty. Fig. 3 illustrates the overall architecture of MonoDLGD, consisting of a backbone, 3D detection encoder, and 3D detection decoder. Following prior DETR-based architectures (Pu et al. 2025; Zhang et al. 2023), the encoder layer is composed of self-attention layers followed by feedforward layers, while the decoder conducts self-attention across queries and cross-attention between encoder-generated features and queries.

MonoDLGD adopts a two-stage architecture utilizing the

3D detection decoder. In Stage 1, label queries are passed through the decoder and two prediction heads to estimate detection uncertainty for projected bounding boxes and depth attributes. Based on the estimated uncertainty, Difficulty-Aware Perturbations (DAP) are applied to generate perturbed label queries.

This strategy facilitates robust learning by adapting perturbation strength to instance-level difficulty. In Stage 2, the perturbed label queries from Stage 1 and the 3D-DAB queries, which explicitly embed spatial priors, are jointly fed to the decoder. The decoder simultaneously performs perturbation reconstruction and 3D object prediction, effectively leveraging instance difficulty and geometric priors to enhance monocular 3D detection.

3D-Dynamic Anchor Box (3D-DAB)

MonoDLGD initializes queries in the detection decoder as 3D-DAB, which explicitly encodes spatial priors rather than using arbitrary learnable embeddings. Inspired by DAB-DETR (Liu et al. 2022), our 3D-DAB extends dynamic anchor boxes to monocular 3D detection by incorporating projected geometry and class semantics. Each query in the 3D-DAB set $Q_{DAB} = \{q_i | i = 1, ..., N\}$, where N denotes the number of 3D-DAB queries in a mini-batch, is defined as

$$q_i = [b_i^{proj}, d_i, c_i] \in \mathbb{R}^{7+C}, \tag{1}$$

where $b^{proj} = [x^{proj}, y^{proj}, o^l, o^t, o^r, o^b] \in \mathbb{R}^6$ denotes the bounding box projected onto the normalized 2D image plane, $d \in \mathbb{R}$ is the depth, and $c \in \mathbb{R}^C$ is the class embedding for C object categories. The projected bounding box b^{proj} consists of the center coordinates (x^{proj}, y^{proj}) and the distances (o^l, o^t, o^r, o^b) from this center to each of the four sides. Here, the superscripts l, t, r, b correspond to the direction from the projected center to each side (left, top, right, bottom) of the bounding box.

By directly encoding the geometric correspondence between the 2D image plane and 3D object space through projected bounding boxes, 3D-DAB constrains the search space to geometrically meaningful regions, rather than relying on arbitrary learnable embeddings. This significantly reduces detection ambiguity and facilitates more accurate and robust monocular 3D object detection (Liu et al. 2022). By embedding these explicit spatial priors into the query representation, 3D-DAB enables the model to localize objects in 3D space more effectively.

Difficulty-Aware Perturbation (DAP)

To address the limited geometric cues and diverse instance difficulty in monocular 3D detection, MonoDLGD introduces the DAP strategy that computes instance-wise difficulty scores based on detector-estimated uncertainty and adaptively scales the perturbation strength for each label query, as shown in Fig. 3. Harder objects with higher uncertainty receive smaller perturbations to preserve essential geometric information, while easier objects are perturbed more strongly to effectively regularize training. This results in difficulty-adaptive perturbed label queries, which explicitly guide the model to learn geometry-aware representations through a reconstruction process. Since perturbation and reconstruction are applied only during training, DAP introduces negligible additional inference-time cost.

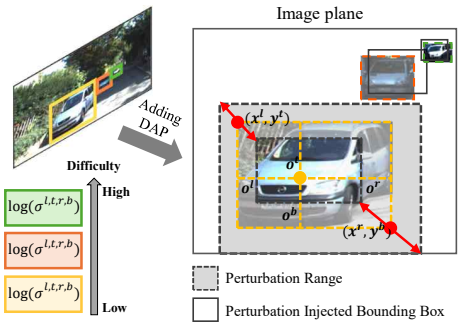


Figure 4: Difficulty-Aware Perturbation (DAP) for projected bounding boxes. Objects with lower uncertainty (lower difficulty scores) receives larger perturbations (*e.g.*, yellow box).

The perturbed label query set is denoted by $Q_{LP} = \{\tilde{q}_i | i = 1, ..., K\}$, where K denotes the number of objects in a mini-batch. The proposed DAP consists of two stages: (i) Difficulty score estimation and (ii) Difficulty-aware label perturbation.

Difficulty Score Estimation Difficulty scores are computed based on the estimation uncertainty of depth and projected bounding box for each instance. We estimate uncertainty using ground-truth labels rather than label queries corrupted with uniform noise, which often require careful hyperparameter tuning and may lead to unstable training dynamics. Ground-truth labels inherently encode object-level geometry and supervision fidelity, offering a more stable and reliable signal for uncertainty estimation. A detailed comparison with uniformly noised label queries is presented in the supplementary material.

Specifically, as shown in Fig. 3, label queries, which consist of the projected bounding box coordinates b^{proj}, depth, and one-hot class vector are first processed by the 3D detection decoder in Stage 1. The resulting features are then fed into two separate detection heads, the projected bounding box head and depth head, to estimate the log-variance uncertainties $\log(\sigma^v)$, where $v \in \{d, l, t, r, b\}$. The subscript i is omitted here for notational simplicity. Note that the uncertainties of the projected bounding box are estimated for (x^l, y^t, x^r, y^b), where (x^l, y^t) and (x^r, y^b) are the top-left and bottem-right coordinates, instead of $b^{proj} = (x^{proj}, y^{proj}, o^l, o^t, o^r, o^b)$ used in 3D-DAB. This is because the latter would require additional uncertainty estimation for the projection center x^{proj}, y^{proj}, complicating the perturbation design.

To convert uncertainty into a certainty score, we compute the inverse of the log-variance:

$$c^v = \exp(-\log(\sigma^v)), \quad v \in \{d, l, t, r, b\}. \tag{2}$$

The resulting certainty values are then min-max normalized to obtain a relative difficulty score $\hat{c} \in [0, 1]$:

$$\hat{c}^v = \frac{c^v - c^v_{\text{min}}}{c^v_{\text{max}} - c^v_{\text{min}}}, \tag{3}$$

where c^v_{min} and c^v_{max} represent the minimum and maximum of the certainty c over entire training dataset. A higher $\hat{c}$ indicates greater prediction certainty. To ensure that normalization captures the global distribution of prediction difficulties throughout training, the minimum and maximum certainty values are updated at each batch using an exponential moving average (EMA):

$$c^v_{min,t} \leftarrow \beta c^v_{min,t-1} + (1-\beta)c^v_{min,t}, \tag{4}$$

where $c^v_{min,t}$ is the minimum certainty at iteration t, and β is the momentum coefficient. $c^v_{max,t}$ is computed in the same way. $c^v_{min,0}$, $c^v_{max,0}$ are initialized from the first mini-batch.

Difficulty-Aware Label Perturbation The computed instance-wise difficulty score $\hat{c}^v_i$ for $v \in \{d, l, t, r, b\}$ is subsequently used as a perturbation scale factor. This perturbation is independently applied to the depth d and the projected bounding box coordinates $b^{2D} = [x^l, y^t, x^r, y^b]$.

(a) Projected Bounding box Perturbation:

Fig. 4 shows the overall procedure of the bounding box perturbation. We inject perturbations into the bounding box coordinates $b^{2D} = [x^l, y^t, x^r, y^b]$ using the computed difficulty scores. Specifically, to calculate each coordinates perturbation scale, we independently sample a random sign $s \sim U\{-1, 1\}$ and multiply it by the difficulty score $\hat{c} \in [0, 1]$, the boundary distance $[o^l, o^t, o^r, o^b]$ and a bounding box perturbation scaling factor $\gamma^b \in (0, 1)$. The resulting perturbed coordinates $\tilde{b}^{2D}$ are computed as follows:

$$\begin{aligned} \tilde{x}^v &= \text{CLIP}_{(0,1)}(x^v + o^v \cdot \hat{c}^v \cdot s^v \cdot \gamma^b), \quad v \in \{l, r\}, \\ \tilde{y}^v &= \text{CLIP}_{(0,1)}(y^v + o^v \cdot \hat{c}^v \cdot s^v \cdot \gamma^b), \quad v \in \{t, b\}. \end{aligned} \tag{5}$$

$\text{CLIP}_{a,b}(x) = min(max(x, a), b)$ ensures that the perturbed bounding box remains within the normalized image plane coordinates range $[0, 1]$.

The perturbation is constrained within the range $-1 < \hat{c}^v \cdot s^v \cdot \gamma^b < 1$, so the perturbed coordinates satisfy $\tilde{x}^l < \tilde{x}^r$ and $\tilde{y}^t < \tilde{y}^b$. Through this approach, we ensure that the perturbation scale of each b^{2D} remains within the boundary distances and satisfies the geometric constraints $0 \leq \tilde{x}^l < \tilde{x}^r \leq 1$ and $0 \leq \tilde{y}^t < \tilde{y}^b \leq 1$ of a valid bounding box. Alg. 1 shows the process of perturbing projected bounding boxes.

(b) Depth Perturbation: The depth perturbation is performed similarly to the bounding box perturbation. To determine the depth perturbation scale, we multiply the depth d by a randomly sampled sign $s^d \sim U\{-1, 1\}$, the depth difficulty score $\hat{c}^d \in (0, 1)$, and a depth perturbation scaling factor $\gamma^d \in (0, 1)$ as follows:

$$\tilde{d} = d + d \cdot \hat{c}^d \cdot s^d \cdot \gamma^d. \tag{6}$$

(c) Class Perturbation: In monocular 3D object detection, class information serves as a strong prior to constraining object size and aspect ratio. Therefore, perturbing the class label during training can act as a useful form of regularization. We adopt a label-flipping strategy where class labels are randomly switched to another class with equal probability. Unlike depth or bounding box perturbations, class perturbation is difficulty-agnostic and applied uniformly across all instances. The final perturbed label query is defined as:

$$\tilde{q}_i = [\tilde{b}_i, \tilde{d}_i, \tilde{c}_i]. \tag{7}$$

Algorithm 1: DAP for Projected Bounding Box

Input: $b^{proj} = (x^{proj}, y^{proj}, o^l, o^t, o^r, o^b)$, $(\hat{c}^l, \hat{c}^t, \hat{c}^r, \hat{c}^b)$
Reparameterize
$b^{proj} \rightarrow b^{2D} = (x^l, y^t, x^r, y^b)$
$x^l = x^{proj} - o^l, \quad y^t = y^{proj} - o^t$
$x^r = x^{proj} + o^r, \quad y^b = y^{proj} + o^b$
Difficulty-Aware Label Perturbation
for each coordinate $v \in \{l, t, r, b\}$ **do**
  Sample random sign $s^v \sim U\{-1, 1\}$
  Compute perturbation scale: $\Delta = o^v \cdot \hat{c}^v \cdot s^v \cdot \gamma^b$
  if $v \in \{l, r\}$ **then**
    Update coordinate: $\tilde{x}^v = \text{CLIP}_{(0,1)}(x^v + \Delta)$
  else if $v \in \{t, b\}$ **then**
    Update coordinate: $\tilde{y}^v = \text{CLIP}_{(0,1)}(y^v + \Delta)$
  end if
end for
Obtain perturbed box: $\tilde{b}^{2D} = (\tilde{x}^l, \tilde{y}^t, \tilde{x}^r, \tilde{y}^b)$
Inverse-Reparameterize
$\tilde{b}^{2D} \rightarrow \tilde{b}^{proj} = (\tilde{x}^{proj}, \tilde{y}^{proj}, \tilde{o}^l, \tilde{o}^t, \tilde{o}^r, \tilde{o}^b)$,
where $(\tilde{x}^{proj}, \tilde{y}^{proj})$ are the center coordinates of the *reparameterized* perturbed bounding box $\tilde{b}^{2D}$.
Output: Perturbed projected bounding box $\tilde{b}^{proj}$

Difficulty-Aware Reconstruction

The perturbed label queries $Q_{LP} = \{\tilde{q}_i | i = 1, ..., K\}$, generated through DAP, are fed into the 3D detection decoder along with the 3D-DAB queries $Q_{DAB} = \{q_i | i = 1, ..., N\}$. K and N are the number of objects in a mini-batch and the number of 3D anchor queries, respectively. Both query sets share the same detection heads (projected Bbox, depth, category), as shown in Fig. 3. Additionally, both the projected bounding box head and the depth head estimate the uncertainty of each corresponding attribute.

The decoder is supervised by (i) reconstructing original labels from Q_{LP} and (ii) detecting objects from Q_{DAB}. Since the perturbed label query $\tilde{q}_i$ has a known corresponding ground truth label, Hungarian matching is not required for reconstruction loss (L_{recon}). For reconstructing the projected bounding box and depth, we employ the Laplacian aleatoric uncertainty loss, enabling uncertainty-adaptive training:

$$L^d_{recon} = \sum_{i=1}^{K} \left(\frac{\sqrt{2}}{\sigma^d_i} ||d_{gt,i} - d_{recon,i}||_1 + log(\sigma^d_i) \right), \tag{8}$$

$$\begin{aligned} L^{bbox}_{recon} = \sum_{i=1}^{K} \Bigg(& \sum_{v \in \{l,r\}} \left(\frac{\sqrt{2}}{\sigma^v_i} \left\| x^v_{gt,i} - x^v_{recon,i} \right\|_1 + \log(\sigma^v_i) \right) \\ & + \sum_{v \in \{t,b\}} \left(\frac{\sqrt{2}}{\sigma^v_i} \left\| y^v_{gt,i} - y^v_{recon,i} \right\|_1 + \log(\sigma^v_i) \right) \Bigg) \end{aligned} \tag{9}$$

where (x_{gt}, y_{gt}) and (x_{recon}, y_{recon}) represent ground truth and reconstructed coordinates of *reparameterized* projected

bounding boxes, respectively. Class reconstruction utilizes standard cross-entropy loss.

The overall reconstruction loss is defined as:

$$L_{recon} = \lambda_{bbox} L^{bbox}_{recon} + \lambda_d L^{d}_{recon} + \lambda_{cls} L^{cls}_{recon}, \quad (10)$$

where λ denotes the weighting factor to balance losses. Our proposed module can be easily integrated as a plug-in into existing DETR-based 3D object detectors, introducing negligible additional inference-time computational overhead as perturbation and reconstruction occur only during training.

Loss Function

The overall training objective comprises the label reconstruction loss L_{recon} and the detection loss L_{det} adopted from baseline methods. For predictions from the 3D-DAB queries, we perform Hungarian matching with ground truth labels, and then apply the same loss function as MonoDGP (Pu et al. 2025) for orientation, 3D size, 2D projected bounding box, depth, and class:

$$L = L_{recon} + L_{det}. \quad (11)$$

Experiments

Experimental Setup

Dataset We evaluated our method on the KITTI 3D object detection benchmark (Geiger, Lenz, and Urtasun 2012), a widely used dataset in monocular 3D detection. The dataset contains 7,481 training images and 7,518 testing images, and provides annotations for three object categories (Car, Pedestrian, and Cyclist). Each object is further classified into three difficulty levels (Easy, Moderate, Hard). Following the common protocol established in (Chen et al. 2015), we split the 7,481 training images into 3,712 for training and 3,769 for validation.

Evaluation Metrics We reported results across the three difficulty levels (Easy, Moderate, Hard) using Average Precision (AP) for 3D bounding boxes AP_{3D} and bird eye view projection AP_{BEV}. These metrics are computed over 40 recall positions in accordance with the official KITTI protocol (Simonelli et al. 2019). All methods were ranked based on the Moderate AP_{3D} score for the Car category.

Implementation Details Our method is implemented on top of MonoDGP (Pu et al. 2025), which employs ResNet-50 (He et al. 2016) as the backbone network. The loss for detection L_{det} follow the MonoDGP setup. We conducted both main and ablation experiments based on the MonoDGP-based implementation. The model was trained for 250 epochs using the Mixup3D (Li, Jia, and Shi 2024) strategy following (Pu et al. 2025). The batch size and initial learning rate were set to 8 and 2×10^{-4}, respectively. The AdamW optimizer (Loshchilov and Hutter 2018) was used with a weight decay of 10^{-4} and the learning rate decayed by a factor of 0.5 at epochs 85, 125, 165, and 225. Training was conducted on an NVIDIA RTX A6000. During inference, we discarded queries with category confidence scores fall below 0.2. In addition, we applied our method to a MonoDETR-based implementation. Training follows the same setup as MonoDGP.

Main Results

We evaluated our proposed method, implemented on top of the MonoDGP (Pu et al. 2025) architecture. Table 1 reports results on the KITTI 3D test set, evaluated using the official online server (Geiger, Lenz, and Urtasun 2012) for fair comparison. To facilitate better learning of 3D geometric information from objects with varying levels of difficulty, our method introduces a difficulty-aware label-guided denoising strategy. As shown, MonoDLGD achieves state-of-the-art performance across all difficulty levels, without relying on any additional training data. Compared to the MonoDGP baseline, MonoDLGD significantly improves AP^{R40}_{3D} by +2.76 (Easy), +1.15 (Moderate), and +1.77 (Hard) on the test set. These consistent gains demonstrate that MonoDLGD effectively enhances 3D geometric reasoning in monocular settings, especially under challenging conditions such as occlusion, truncation, and depth ambiguity.

To further evaluate the versatility of our approach, we integrated MonoDLGD into the MonoDETR (Zhang et al. 2023) architecture without modifying its core design. Specifically, we applied our label denoising strategy in conjunction with 3D-DAB queries. As shown in Table 2, this integration yields consistent improvements. These results suggest that our method can serve as a complementary component to existing DETR-based monocular 3D detection pipelines, potentially benefiting a broader range of architectures with similar formulations.

Ablation Study

Efficiency Comparison Table 2 compares the inference time on the KITTI validation set. All methods were evaluated under the same computational environment using a single NVIDIA Titan RTX GPU with a batch size of 1 to ensure fair comparison. The average inference time per image for MonoDGP (Pu et al. 2025) and MonoDETR (Zhang et al. 2023) is 42.4 ms and 35.2 ms, respectively. When integrated with the proposed method, the inference time remains nearly unchanged. The training time increases slightly due to the added perturbation and reconstruction operations in Stage 1 (see Fig. 3), which are applied only during training. A detailed analysis is provided in the supplementary.

Effectiveness of Label-Guided Denoising with 3D-DAB and DAP Table 3 presents an ablation study evaluating the core components of MonoDLGD framework. Starting from the MonoDGP baseline (a), simply replacing anchor queries 3D-DAB (b) slightly degrades performance due to the lack of accompanying supervision despite encoding spatial priors. However, combining 3D-DAB with uniform label perturbation (c) yields immediate performance gains, demonstrating that label-guided denoising enhances 3D geometric learning by providing explicit supervision to the anchor queries. Further gains are achieved by introducing Difficulty-Aware Perturbation (DAP) based on predictive uncertainty, *i.e.* (d) *vs.* (e). Unlike the uniform noise scheme of DN-DETR (Li et al. 2022a), DAP adaptively regularizes easy instances while preserving the geometric structure of hard examples. Overall, while 3D-DAB provides strong geometric supervision for monocular 3D detection, DAP en-

Table 1: **Comparisons on the KITTI test and validation sets (Car category)**. We **bold** the best results and underline the second-best results.

Methods	Extra data	Reference	Test, $AP_{BEV\|R40}$			Test, $AP_{3D\|R40}$			Val, $AP_{BEV\|R40}$			Val, $AP_{3D\|R40}$		
			Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
MonoDTR (Huang et al. 2022)	LiDAR	CVPR 2022	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)		ECCV 2022	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)		CVPR 2024	35.38	24.18	21.37	25.55	17.02	14.79	35.72	26.60	23.68	26.87	19.96	17.15
MonoPGC (Wu et al. 2023)	Depth	ICRA 2023	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)		RAL 2023	33.54	22.53	19.22	24.60	17.05	14.25	33.80	25.51	22.13	24.97	19.40	16.59
DEVIANT (Kumar et al. 2022)	None	ECCV 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. 2022b)		CVPR 2022	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)		NeurIPS 2023	-	-	-	24.75	16.73	13.49	-	-	-	24.51	17.18	14.01
MonoDETR (Zhang et al. 2023)		ICCV 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
MonoCD (Yan et al. 2024)		CVPR 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
FD3D (Wu et al. 2024)		AAAI 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)		NeurIPS 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
MonoDGP (Pu et al. 2025)		CVPR 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
Ours	None		**36.63**	**25.3**	**23.13**	**29.11**	**19.87**	**17.74**	**41.68**	**30.53**	**27.76**	**34.89**	**25.19**	**21.78**
Improvement over Second-Best Method	-		+1.39	+0.07	+1.11	+2.76	+1.03	+0.96	+1.42	+2.33	+3.34	+4.13	+2.85	+2.76
Improvement over MonoDGP Baseline	-		+1.39	+0.07	+1.11	+2.76	+1.15	+1.77	+2.28	+2.33	+3.34	+4.13	+2.85	+2.76

Table 2: **More results with computational cost.** The proposed method was implemented on top of MonoDETR (Zhang et al. 2023) and MonoDGP (Pu et al. 2025). * indicates results reproduced from the authors' official code. Performance was evaluated on the KITTI validation set for Car category.

Method	Val, $AP_{BEV\|R40}$			Val, $AP_{3D\|R40}$			GFLOPs↓	Time (ms)↓
	Easy	Mod.	Hard	Easy	Mod.	Hard		
MonoDETR*	36.38	26.19	22.29	27.34	19.33	16.04	59.7	35.2
+Ours	38.59	27.65	23.62	29.79	21.63	18.17	59.8	35.5
MonoDGP	39.40	28.20	24.42	30.76	22.34	19.02	69.0	42.4
+Ours	41.68	30.53	27.76	34.89	25.19	21.78	69.3	42.7

Table 3: **Ablation on the KITTI val set (Car)**. **(a)** serves as our baseline with the same architecture as MonoDGP (Pu et al. 2025). **UN** indicates uniform noise ignores instance-level detection difficulty, equivalent to DN-DETR (Li et al. 2022a). L_{recon}^{bbox} means the type of projected bounding box reconstruction loss. **LU** is Laplacian Uncertainty Loss adopted in our method.

Idx	3D-DAB	Perturb.	L_{recon}^{bbox}	Val, $AP_{BEV\|R40}$			Val, $AP_{3D\|R40}$		
				Easy	Mod.	Hard	Easy	Mod.	Hard
(a)	×	×	×	39.40	28.20	24.42	30.76	22.34	19.02
(b)	O	×	×	36.85	26.72	23.21	27.82	20.64	17.78
(c)	O	UN	L1	40.32	30.13	26.53	31.99	23.82	20.65
(d)	O	UN	LU	41.16	30.31	26.54	33.82	24.7	21.19
(e): Ours	O	DAP	LU	41.68	30.53	27.76	34.89	25.19	21.78

Table 4: **Ablation study on denoising target configurations** on the KITTI validation set (Car category).

Denoising Setup	Val, $AP_{BEV\|R40}$			Val, $AP_{3D\|R40}$		
	Easy	Mod.	Hard	Easy	Mod.	Hard
Baseline (MonoDGP)	39.40	28.20	24.42	30.76	22.34	19.02
Bbox (proj.) + Class	40.58	29.75	26.18	30.93	23.36	20.30
Bbox (proj.) + Class + Depth	41.68	30.53	27.76	34.89	25.19	21.78

ables more robust learning by adapting to instance-level difficulty.

Effectiveness of Uncertainty in Denoising We extend the use of aleatoric uncertainty to the bounding box denoising process. As shown in Table 3, this extension yields a notable performance improvement, *i.e.* (c) *vs.* (d). Incorporating uncertainty-aware estimation into the denoising process helps the model downweight unreliable supervision from hard or ambiguous objects, allowing it to focus on more confident and informative signals. These results suggest that aleatoric uncertainty provides a meaningful signal for modeling instance-level detection difficulty in label denoising.

Effectiveness of Depth Information To evaluate the contribution of depth information in our method, we conducted an ablation study by excluding depth attributes from both the label queries and the denoising process. As shown in Table 4, applying our denoising strategy solely to the projected bounding boxes and class labels improves the moderate AP_{3D}^{R40} from 22.34% to 23.36%, validating the effectiveness of denoising core detection components. More importantly, further incorporating depth into the denoising process yields a significant performance boost to 25.19%, highlighting the importance of depth supervision in enhancing geometric representation.

Conclusion

We have presented MonoDLGD, a novel framework for monocular 3D object detection that introduces difficulty-aware label-guided denoising via label perturbation and reconstruction. By adaptively modulating perturbation strength based on reconstruction uncertainty, our method explicitly incorporates geometric supervision and effectively mitigates the ill-posed nature of monocular 3D object detection. Furthermore, our uncertainty-aware estimation strategy leads to consistent performance gains, highlighting the importance of modeling instance-level uncertainty. Extensive experiments on the KITTI benchmark demonstrate that MonoDLGD consistently improves 3D detection performance across all difficulty levels.

Acknowledgments

This work was supported by the NRF of Korea grant (RS-2025-24803204), the Bio & Medical Technology Development Program of the NRF of Korea (RS-2022-NR068424), and the BK21 FOUR funded by the Ministry of Education (MOE, Korea) and the NRF of Korea (2120251015523).

References

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, 213–229. Springer.

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.

Chen, Y.; Huang, S.; Liu, S.; Yu, B.; and Jia, J. 2022. Dsgn++: Exploiting visual-spatial relation for stereo-based 3d detectors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4): 4416–4429.

Chen, Y.; Tai, L.; Sun, K.; and Li, M. 2020. Monopair: Monocular 3d object detection using pairwise spatial relationships. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12093–12102.

Choi, J.; Chun, D.; Kim, H.; and Lee, H. J. 2019. Gaussian YOLOv3: An Accurate and Fast Object Detector Using Localization Uncertainty for Autonomous Driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 502–511. IEEE.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3354–3361. IEEE.

Guo, X.; Shi, S.; Wang, X.; and Li, H. 2021. LIGA-Stereo: Learning LiDAR Geometry Aware Representations for Stereo-based 3D Detector. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3153–3163. IEEE.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 770–778.

He, Y.; Zhu, C.; Wang, J.; Savvides, M.; and Zhang, X. 2019. Bounding Box Regression with Uncertainty for Accurate Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2888–2897. IEEE.

Huang, K.-C.; Wu, T.-H.; Su, H.-T.; and Hsu, W. H. 2022. Monodtr: Monocular 3d object detection with depth-aware transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4012–4021.

Jiang, X.; Jin, S.; Zhang, X.; Shao, L.; and Lu, S. 2024. MonoMAE: Enhancing Monocular 3D Detection through Depth-Aware Masked Autoencoders.

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.

Kendall, A.; and Gal, Y. 2017. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 30.

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

Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12697–12705.

Li, F.; Zhang, H.; Liu, S.; Guo, J.; Ni, L. M.; and Zhang, L. 2022a. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13619–13627.

Li, P.; Chen, X.; and Shen, S. 2019. Stereo r-cnn based 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7644–7652.

Li, Z.; Jia, J.; and Shi, Y. 2024. MonoLSS: Learnable sample selection for monocular 3D detection. In *International Conference on 3D Vision*, 1125–1135. IEEE.

Li, Z.; Qu, Z.; Zhou, Y.; Liu, J.; Wang, H.; and Jiang, L. 2022b. 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.

Liu, S.; Li, F.; Zhang, H.; Yang, X.; Qi, X.; Su, H.; Zhu, J.; and Zhang, L. 2022. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*.

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.; Hou, J.; Wang, X.; Ye, X.; Wang, J.; Zhao, H.; and Bai, X. 2024a. Lion: Linear Group RNN for 3D Object Detection in Point Clouds. *Advances in Neural Information Processing Systems*, 37: 13601–13626.

Liu, Z.; Hou, J.; Ye, X.; Wang, T.; Wang, J.; and Bai, X. 2024b. Seed: A Simple and Effective 3D DETR in Point Clouds. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, 110–126. Springer.

Loshchilov, I.; and Hutter, F. 2018. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

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.

Ma, X.; Zhang, Y.; Xu, D.; Zhou, D.; Yi, S.; Li, H.; and Ouyang, W. 2021. Delving into localization errors for

monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4721–4730.

Peng, L.; Wu, X.; Yang, Z.; Liu, H.; and Cai, D. 2022. Didm3d: Decoupling instance depth for monocular 3d object detection. In *Proceedings of the IEEE/CVF 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

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.

Simonelli, A.; Bulo, S. R.; Porzi, L.; López-Antequera, M.; and Kontschieder, P. 2019. Disentangling monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1991–1999.

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.

Wang, H.; Shi, C.; Shi, S.; Lei, M.; Wang, S.; He, D.; and Wang, L. 2023. DSVT: Dynamic Sparse Voxel Transformer with Rotated Sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13520–13529. IEEE.

Wu, Z.; Gan, Y.; Wang, L.; Chen, G.; and Pu, J. 2023. Monopgc: Monocular 3d object detection with pixel geometry contexts. In *International Conference on Robotics and Automation*, 4842–4849. IEEE.

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, F.; Xu, X.; Chen, H.; Guo, Y.; Han, J.; Ni, K.; and Ding, G. 2022. Ground Plane Matters: Picking Up Ground Plane Prior in Monocular 3D Object Detection. *arXiv preprint arXiv:2211.01556*.

Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L. M.; and Shum, H.-Y. 2022. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*.

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.

Zhang, Y.; Lu, J.; and Zhou, J. 2021. Objects are different: Flexible monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3289–3298.

Zhu, B.; Wang, Z.; Shi, S.; Xu, H.; Hong, L.; and Li, H. 2023. Conquer: Query Contrast Voxel-DETR for 3D Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9296–9305.

Appendix

Analyzing the Relationship Between Uncertainty and Difficulty

The proposed Difficulty-Aware Perturbation (DAP) module computes a difficulty score for each instance based on the predicted aleatoric uncertainty of its depth and *reparameterized* projected bounding box in Eq. (8) and (9) of the main paper. This score is then used to modulate the perturbation magnitude in Eq. (5) and (6) of the main paper. Instances with higher uncertainty (*i.e.*, greater difficulty) are assigned smaller perturbations to preserve critical geometric cues, and *vice versa*.

To empirically validate that the predicted uncertainty reflects the instance-level difficulty, we analyze the uncertainty distribution across difficulty levels provided by the KITTI benchmark, which manually annotates each object as *Easy* (Level 1), *Moderate* (Level 2), or *Hard* (Level 3) based on occlusion, truncation, and distance. As shown in Fig. 5, the uncertainty for depth and horizontal box coordinates (x^l, x^r) increases with difficulty level, confirming that our predicted uncertainty reflects instance-level detection difficulty. In contrast, vertical coordinates (y^t, y^b) exhibit relatively stable uncertainty across difficulty levels.

This asymmetry stems from the inherent nature of monocular 3D detection: horizontal localization is more affected by occlusion and truncation, whereas vertical positioning is less influenced by these difficulty factors due to by stable geometric contexts such as the ground plane and camera height. Nonetheless, vertical cues remain essential to defining the overall 3D box geometry and provide complementary information, particularly in scenes with sloped terrain or uneven ground.

To further validate this, Tab. 5 presents a quantitative analysis that compares the impact of applying DAP to horizontal and/or vertical coordinates of the *reparameterized* projected bounding box. Note that these results extend the analysis of Tab. 4 in the main paper. The results show that perturbing horizontal coordinates leads to greater performance gains, consistent with their higher sensitivity to instance-level difficulty. Nevertheless, incorporating perturbations along the vertical axis further yields meaningful improvements, supporting our choice to model uncertainty across both axes. This comprehensive treatment enables more robust perturbation and reconstruction under diverse scene geometries.

Table 5: **Effect of Different Geometric Supervision Components on Validation Performance.** The last row corresponds to our method using all geometric components.

Class	Depth	Bbox (Proj.)		Val, $AP_{BEV\|R40}$			Val, $AP_{3D\|R40}$		
		Horizon	Vertical	Easy	Mod.	Hard	Easy	Mod.	Hard
O	×	×	O	40.21	29.36	25.71	31.17	22.42	19.14
O	×	O	×	39.27	29.35	25.76	32.08	23.65	20.39
O	×	O	O	40.06	29.65	27.03	31.52	23.74	20.61
O	O	×	O	40.92	29.92	26.37	30.34	23.00	20.04
O	O	O	×	40.82	29.95	27.13	32.35	24.05	20.78
O	O	O	O	**41.68**	**30.53**	**27.76**	**34.89**	**25.19**	**21.78**

Geometric Constraints on Projected Bounding Box Perturbation

In this section, we explicitly demonstrate that the *reparameterized* Projected Bounding Box Perturbation defined in Eq. (5) of the main paper consistently satisfies valid geometric constraints. Specifically, we prove that the perturbed coordinates $\tilde{b}^{2D} = (\tilde{x}^l, \tilde{y}^t, \tilde{x}^r, \tilde{y}^b)$ always preserve the correct positional order with respect to the projected bounding box center (x^{proj}, y^{proj}), such that:

$$\tilde{x}^l < x^{proj} < \tilde{x}^r, \quad \tilde{y}^t < y^{proj} < \tilde{y}^b \tag{12}$$

Before perturbaton, the *reparameterized* projected bounding box $b^{2D} = (x^l, y^t, x^r, y^b)$ satisfies the following geometric constraints:

$$\begin{aligned} &0 \le x^l = x^{proj} - o^l < x^{proj} < x^r = x^{proj} + o^r \le 1, \\ &0 \le y^t = y^{proj} - o^t < y^{proj} < y^b = y^{proj} + o^b \le 1. \end{aligned} \tag{13}$$

The perturbation scale Δ^v injected into each coordinate is defined as:

$$\Delta^v = o^v \cdot \hat{c}^v \cdot s^v \cdot \gamma^b, \quad v \in \{l, t, r, b\} \tag{14}$$

where $s^v \sim U\{-1, 1\}$, $0 \le \hat{c}^v \le 1$, and $0 < \gamma^b < 1$. Thus, the absolute value of the perturbation is upper bounded by:

$$|\Delta^v| = o^v \cdot \hat{c}^v \cdot \gamma^b < o^v \tag{15}$$

Here, o^v for $v \in \{l, t, r, b\}$ denotes the distance from the projected center coordinates (x^{proj}, y^{proj}) to each side of the bounding box. This inequality indicates that the perturbation magnitude $|\Delta^v|$ is always less than the corresponding boundary distance o^v, ensuring that the perturbed coordinates remain within valid geometric limits.

Specifically, the perturbed coordinates satisfy the following conditions. For the horizontal coordinates x^l, the perturbation in the positive direction yields:

$$\tilde{x}^l = x^l + \Delta^l < x^l + o^l = x^{proj} \tag{16}$$

Conversely, perturbing negatively gives:

$$\tilde{x}^l = x^l + \Delta^l > x^l - o^l = x^{proj} - 2o^l \tag{17}$$

Here, $x^{proj} - 2o^l$ could potentially yield a negative value; however, the `CLIP` in Eq. (5) of the main paper operation enforces a lower bound of 0, thereby guaranteeing that $0 \le \tilde{x}^l < x^{proj}$ always holds.

The same logic applies to the right boundary coordinate x^r. For the maximal negative perturbation:

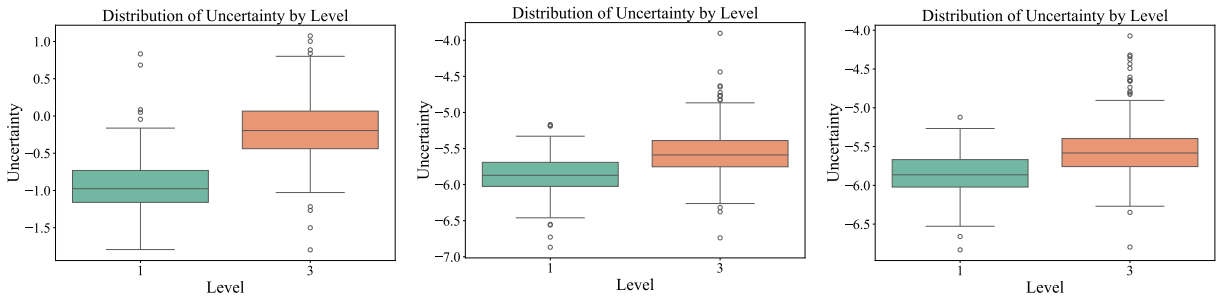
$$\tilde{x}^r = x^r + \Delta^r > x^r - o^r = x^{proj} \tag{18}$$

and for the maximal positive perturbation:

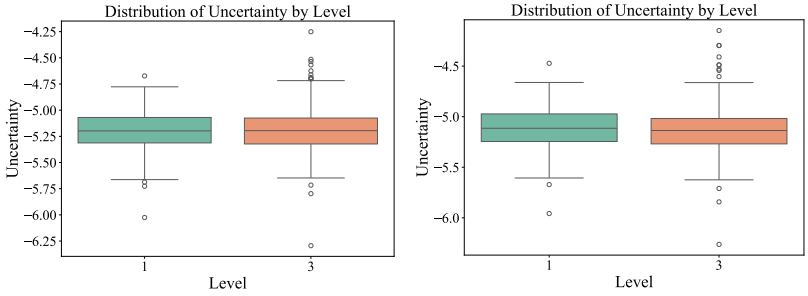
$$\tilde{x}^r = x^r + \Delta^r < x^r + o^r = x^{proj} + 2o^r \tag{19}$$

$x^{proj} + 2o^r$ may also exceed 1, the `CLIP` operation restricts the upper bound to 1, ensuring $x^{proj} < \tilde{x}^r \le 1$.

Similarly, the top and bottom boundary coordinates, $\tilde{y}^t$ and $\tilde{y}^b$, are guaranteed to satisfy the geometric constraints $0 \le \tilde{y}^t < y^{proj}$ and $y^{proj} < \tilde{y}^b \le 1$, respectively.



(a) Depth (b) Projected Bounding Box (x^l) (c) Projected Bounding Box (x^r)



(d) Projected Bounding Box (y^t) (e) Projected Bounding Box (y^b)

Figure 5: **Distribution of predicted uncertainty across difficulty levels on the KITTI validation set.** The x-axis represents the difficulty levels defined by the KITTI benchmark, where level 1 corresponds to *Easy* and level 3 corresponds to *Hard*. The y-axis shows the distributios and mean values of uncertainties predicted by the depth and projected bounding box detection heads.

Consequently, the perturbed bounding box coordinates are guaranteed to satisfy the following geometric constraints:

$$\begin{aligned} &0 \leq \tilde{x}^l < x^{proj} < \tilde{x}^r \leq 1 \rightarrow 0 \leq \tilde{x}^l < \tilde{x}^r \leq 1 \\ &0 \leq \tilde{y}^t < y^{proj} < \tilde{y}^b \leq 1 \rightarrow 0 \leq \tilde{y}^t < \tilde{y}^b \leq 1 \end{aligned} \tag{20}$$

These inequalities ensure that the perturbed box remains a valid bounding box with properly ordered and normalized coordinates.

Implementation Details

Implementation Details of Perturbed Label Query Following the approaches introduced by DN-DETR (Li et al. 2022a) and DINO (Zhang et al. 2022), we construct multiple perturbation groups when generating perturbed label queries. Within each perturbation group, both positive and negative queries are created.

Specifically, each query is perturbed by randomly sampling a perturbation sign $s^v \sim U\{-1, 1\}$, and multiplying it by a group agnostic perturbation scale $o^v \cdot \hat{c}^v \cdot \gamma^b$, ensuring each query group to have diverse perturbations signs while sharing the same perturbation scale. Consequently, the queries within each perturbation group reflect varied perturbations relative to the original labels. The effect of the number of perturbation groups on model performance is analyzed in Tab. 10.

Additionally, following DINO (Zhang et al. 2022), we generate negative query groups corresponding to each positive perturbation group. Specifically, for negative queries, we increase the perturbation magnitude by adding 1 to the perturbation scale of the corresponding positive queries, preserving the original perturbation direction defined by the sign s^v.

During training, these negative queries are explicitly assigned to the "no object" class in the class reconstruction loss, effectively encouraging the model to distinguish foreground objects from background regions more clearly.

Implementation Differences between MonoDGP and MonoDETR In the main paper, we demonstrated that our proposed method can be effectively integrated with DETR-based detectors such as MonoDETR (Zhang et al. 2023) and MonoDGP (Pu et al. 2025). Although both methods share the same DETR architecture, they differ in how the final depth prediction is derived from the outputs of their respective depth heads. These differences lead to distinct strategies for constructing the label queries for the depth component. Below, we explicitly outline these implementation differences:

- **Depth Predictoin in MonoDETR:** MonoDETR computes the final predicted depth d_{pred} as an average of three components: the direct output from the depth head (d_{reg}), the geometric depth derived from perspective projection formula (d_{geo}), and a foreground depth map estimation (d_{map}). Since the depth head directly predicts the absolute regressed depth value, we use the ground-

truth depth as the depth component when generating label queries.

- **Depth Prediction in MonoDGP:** MonoDGP treats the output of the depth head as a residual error term (d_{err}) rather than an absolute depth value. The final predicted depth d_{pred} is calculated by adding this error term to the geometric depth d_{geo}:

$$d_{pred} = d_{gro} + d_{err} \tag{21}$$

where d_{err} indicates the discrepency between the actual ground truth depth and geometric depth. Given that the MonoDGP depth head outputs this error term, we construct the label query for the depth head by using the residual error computed from the ground truth depth and the geometric depth ground truth.

Additional Ablation Studies

EMA cofficients for difficulty score min/max normalization. In Eq. (4) of the main paper, we apply EMA to update the minimum and maximum values used for normalizing the difficulty scores, effectively capturing the global distribution of data. Tab. 6 shows the ablation results for different EMA momentum coefficients. The baseline corresponds to batch-wise min/max normalization without EMA, resulting in performance degradation due to its inability to reflect the global distribution. The best performance was achieved with an momentum coefficient of 0.8, which we adopt as the default setting.

Table 6: **Ablation study of the EMA momentum coefficient.** The baseline corresponds to the case without EMA.

Momentum	Val, $AP_{BEV\|R40}$			Val, $AP_{3D\|R40}$		
	Easy	Mod.	Hard	Easy	Mod.	Hard
Baseline	41.44	31.33	27.61	32.22	24.32	20.94
$\beta = 0.7$	41.41	31.37	27.84	33.24	24.23	21.78
$\beta = 0.8$	**41.68**	**30.53**	**27.76**	**34.89**	**25.19**	**21.78**
$\beta = 0.9$	41.71	30.55	28.06	33.2	24.29	21.16

Stage 1 Input: Ground-Truth vs. Uniformly-Corrupted Label Queries. In Tab. 7, we compare the detection performance using two types of label queries for Stage 1 of Fig. 3 in the main paper: ground-truth labels and uniformly-corrupted labels. These label queries serve as decoder inputs for the Difficulty-Aware Perturbation (DAP) module in Stage 1. The first row (*Label (Ours)*) corresponds to using noise-free ground-truth labels to estimate instance-level detection uncertainty. The second row (*Uniform Noised Label*) denotes queries constructed by injecting uniform noise into the labels. Experimental results show that using clean labels yields consistently better performance than the noisy counterpart. This is becuase ground-truth labels inherently encode precise object-level geometry and supervision fidelity, thereby providing a more stable and reliable signal for accurate uncertainty estimation. Additionally, our approach avoids the need for extensive hyperparameter tuning (e.g., uniform noise scale selection), resulting in a more stable and efficient training process.

Table 7: **Impact of Noise Injection on Label Queries.**

Method	Val, $AP_{BEV\|R40}$			Val, $AP_{3D\|R40}$		
	Easy	Mod.	Hard	Easy	Mod.	Hard
Label (Ours)	41.68	30.53	27.76	34.89	25.19	21.78
Uniform Noised Label	39.17	29.57	25.96	31.68	23.85	20.55

Ablation Study of Perturbation Scaling Factors We analyze the effects of two perturbation scale factors used during the Difficulty-Aware Perturbation (DAP) process in Stage 1: γ^b , which controls the perturbation magnitude of the projected bounding box, and γ^d, which determine the perturbation magnitude applied to the depth label. Tab. 8 and Tab. 9 show the effects of these scale factors. The best performance is achieved when γ^d and γ^b are set to 0.8 and 0.4, respectively. This suggests that reconstructing perturbed bounding box coordinates is inherently more challenging than reconstructing perturbed depth values.

Moreover, our depth perturbation strategy employs the depth error—defined as the discrepancy between geometric depth and actual ground-truth depth—as the target depth label (detailed in Supplementary Section D). Perturbing this relative error, rather than the absolute depth, inherently provides greater numerical stability. Specifically, depth errors typically exhibit small numerical magnitudes (approximately within the range of -2 to 2 in practice), thus allowing appropriate perturbation scales without significantly distorting the original error distribution. As a result, applying a relatively larger perturbation scale factor to depth labels enhances training effectiveness and contributes positively to overall detection performance and model robustness.

Table 8: **Ablation study of the Bounding box Scale Factor γ^b.**

γ^b	Val, $AP_{BEV\|R40}$			Val, $AP_{3D\|R40}$		
	Easy	Mod.	Hard	Easy	Mod.	Hard
0.2	39.32	29.94	27.4	32.02	24.22	21.81
0.4	**41.68**	**30.53**	**27.76**	**34.89**	**25.19**	**21.78**
0.6	42.38	31.88	28.23	33.28	24.44	21.17
0.8	40.24	29.77	26.17	32	23.85	20.59

Table 9: **Ablation study of the Depth Scale Factor γ^d.**

γ^d	Val, $AP_{BEV\|R40}$			Val, $AP_{3D\|R40}$		
	Easy	Mod.	Hard	Easy	Mod.	Hard
1.0	40.68	30.34	27.85	32	24.29	21.21
0.8	**41.68**	**30.53**	**27.76**	**34.89**	**25.19**	**21.78**
0.6	42.59	30.59	26.59	34.03	24.68	21.15
0.4	43.39	30.56	26.51	33.41	24	20.54

Ablation Study of on the Number of Perturbation Groups In Tab. 10, we analyze the influence of the number of perturbation groups used to construct perturbed label queries. Here, the number of perturbation groups refers to M, with a fixed number of original label queries K. Specifically, when the group number $N = 1$, the total number of perturbed label queries is $2 \times K$, consisting of one positive and one negative set of queries. In general, the total number of perturbed label queries is calculated as $2 \times N \times K$, pro-

portional to the selected perturbation groups N. Each perturbation group is constructed based on the randomness of perturbation sign s^v, which enables the diverse perturbation patterns. Experimental results show that using 7 perturbation groups yields the best performance. This suggests that the randomness of perturbation signs is effective in producing sufficiently diverse perturbed label queries for robust learning.

Table 10: **The Number of Perturbed Label Query Groups.**

# of Perturbed Label Query Groups	Val, $AP_{BEV\|R40}$			Val, $AP_{3D\|R40}$		
	Easy	Mod.	Hard	Easy	Mod.	Hard
1	40.63	29.66	26.05	32.58	23.75	20.56
3	41.27	30.13	27.45	32.88	24.37	21.09
5	41.08	29.83	26.40	33.21	24.02	20.94
7	**41.68**	**30.53**	**27.76**	**34.89**	**25.19**	**21.78**
8	42.16	31.17	27.59	31.91	23.93	20.76

Training Complexity

In Tab. 11, we analyze the training time and memory usage overhead introduced by our MonoDLGD framework. When applied to the baseline model, MonoDGP (Pu et al. 2025), the training time increases by approximately 17.4 %, and memory usage increased by approximately 5.99GB. These increases primarily stem from the additional computational steps introduced by Difficulty-Aware Perturbation (DAP) module, including difficulty score computation, perturbation injection, and the subsequent label reconstruction processes. Additionally, memory usage grows due to the utilization of multiple perturbed label query groups, in this case, seven groups, as discussed in Section E.4.

However, as demonstrated by the computational cost analysis in Tab. 2 of the main paper, these operations are only required during training. The inference process remains unaffected by DAP, resulting in negligible inference-time overhead. Therefore, the proposed MonoDLGD framework delivers substantial accuracy improvements without sacrificing inference efficiency.

Table 11: **Training Complexity.**

Method	Val, $AP_{3D\|R40}$			Training time (min/epoch)↓	Memory Usage (GB)↓
	Easy	Mod.	Hard		
MonoDGP	30.76	22.34	19.02	4.55	12.35
MonoDGP+Ours	34.89	25.19	21.78	5.51	18.34

More Results

Experiments on Other Categories

Tab. 12 presents additional experimental results on other object categories (Pedestrian and Cyclist) from the KITTI dataset. Compared to MonoDGP, MonoDLGD consistently improves performance on the Cyclist category across all difficulty levels, achieving notalbe AP improvements of 4.01 (Easy), 1.86 (Moderate), and 1.28 (Hard). However, for the Pedestrian category, MonoDLGD provide 0.02 AP improvement at the Moderate level, while MonoDGP achieves slightly better performance in Easy and Hard levels. Tab. 13 presents the performance on the KITTI test set. MonoDLGD consistently achieves competitive results across all difficulty levels for the Cyclist category. For the Pedestrian category, MonoDLGD demonstrates the best performance among compared methods, but the performance improvements are relatively modest compared to those observed for Cyclist. This discrepancy in performance between categories can be explained by the uncertainty analysis presented in Fig. 5. Specifically, the difficulty levels defined by the KITTI benchmark primarily reflect challenges related to horizontal localization, including occlusion and truncation. Consequently, our approach, which explicitly models horizontal localization uncertainty, excels on object categories such as Car and Cyclist that exhibit larger horizontal dimensions. Conversely, objects like Pedestrians, which typically possess a vertical-elongated structure, experience relatively stable vertical localization uncertainty. In summary, the proposed MonoDLGD effectively enhances performance for object categories heavily influenced by horizontal localization uncertainty. However, its impact is relatively limited for vertically elongated or upright objects. This observation suggests future research directions toward category-specific perturbation strategies that account for the distinctive geometric properties of different object types.

Table 12: Ablation study of the pedestrian and cyclist categories on the KITTI val set.

Methods	Val, IoU=0.5, $AP_{3D\|R40}$					
	Pedestrian			Cyclist		
	Easy	Mod.	Hard	Easy	Mod.	Hard
MonoDGP (Pu et al. 2025)	**13.77**	10.06	**7.96**	12.21	6.61	5.95
MonoDLGD (Ours)	13.04	**10.08**	7.56	**16.22**	**8.47**	**7.23**

Table 13: Comparisons of the pedestrian and cyclist categories on the KITTI test set.

Methods	Extra data	Test, IoU=0.5, $AP_{3D\|R40}$					
		Pedestrian			Cyclist		
		Easy	Mod.	Hard	Easy	Mod.	Hard
CaDDN (Reading et al. 2021)	LiDAR	12.87	8.14	6.76	7.00	3.41	3.30
OccupancyM3D (Peng et al. 2024)		14.68	9.15	7.80	**7.37**	3.56	2.84
MonoPGC (Wu et al. 2023)	Depth	14.16	9.67	8.26	5.88	3.30	2.85
GUPNet (Lu et al. 2021)	None	14.72	9.53	7.87	4.18	2.65	2.09
MonoCon (Liu, Xue, and Wu 2022)		13.10	8.41	6.94	2.80	1.92	1.55
DEVIANT (Kumar et al. 2022)		13.43	8.65	7.69	5.05	3.13	2.59
MonoDDE (Li et al. 2022b)		11.13	7.32	6.67	5.94	3.78	3.33
MonoDETR (Zhang et al. 2023)		12.65	7.19	6.72	5.12	2.74	2.02
MonoDGP (Pu et al. 2025)		15.04	9.89	8.38	5.28	2.82	2.65
MonoDLGD (Ours)	None	**15.99**	**10.44**	**8.82**	**6.62**	**4.39**	**3.35**

Qualitative Results

To intuitively evalutate the effectiveness of our model under vairous difficulty conditions, we visualize 3D detection results and bird's-eye view (BEV) representations on the KITTI validation set. In Fig. 6, we compare the detection results of previous DETR-based baselines; (a) MonoDETR, (b) MonoDGP, and (c) MonoDLGD (Ours). MonoDLGD achieves more accurate and consistent detection performance, even in challenging environments involving distant or occluded objects.

(a) MonoDETR (b) MonoDGP (c) MonoDLGD (Ours)

Figure 6: **Quality Analysis on KITTI val dataset.** (a) MonoDETR (Zhang et al. 2023) (b) MonoDGP (Pu et al. 2025) (c) MonoDLGD (Ours). For each image, Green box represents ground truth box and Red box represents predicted value.