# Detect Anything 3D in the Wild

## Supplementary Material

## 1. DA3D

DA3D is a unified 3D detection dataset, consists of 16 diverse datasets. It builds upon six datasets in Omni3D—Hypersim [7], ARKitScenes [2], Objectron [1], SUNRGBD [8], KITTI [5], and nuScenes [4]—while partially incorporating an additional 10 datasets to further enhance the scale, diversity, and generalization capabilities of 3D detection models. As shown in Figure 1, DA3D comprises 0.4 million frames (2.5× the scale of Omni3D), spanning 20 distinct camera configurations.

The dataset is standardized with the similar structure to Omni3D [3], including monocular RGB images, camera intrinsics, 3D bounding boxes, and depth maps. DA3D is designed to test 3D detection models across a wide variety of environments, camera configurations, and object categories, offering a more comprehensive evaluation setting.

### 1.1. Dataset Composition

We categorize the datasets in DA3D based on two aspects:

**Indoor *vs*. Outdoor.** As shown in Figure 2 (left), DA3D expands both indoor and outdoor datasets compared to Omni3D. Additionally, the ratio of indoor to outdoor data in DA3D is more balanced than in Omni3D, ensuring a more representative distribution for models trained across diverse environments.

**Supervision Types.** We also analyze DA3D in terms of the distribution of supervision types (See Figure 2 (right)):

- 35% data provides only depth supervision.
- 23% data provide only 3D bounding box annotations.
- 42% data contains both depth maps and 3D bounding boxes.
- Intrinsic parameters are available for all data.

### 1.2. Dataset Splits.

For training and evaluation, we follow the dataset splitting strategy used in prior works [3]. Specifically:

- We construct the training set by merging training subsets from the original datasets.
- We form the validation set by sampling from the original training data, ensuring balanced representation.
- We use the original validation sets of each dataset as the test set, allowing for direct comparison with previous benchmarks.

This setup ensures fair evaluation and maintains consistency with existing benchmarks while assessing both in-domain and zero-shot generalization capabilities.

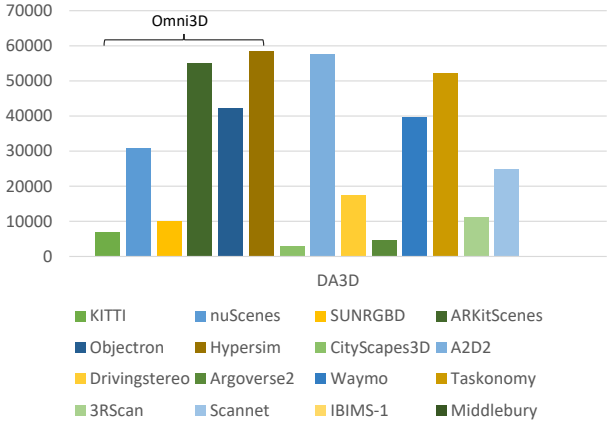


Figure 1. The composition of the DA3D dataset.

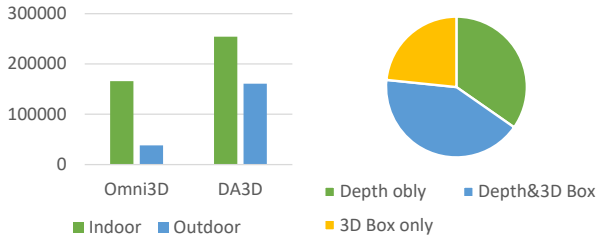


Figure 2. The data distribution of the DA3D dataset. (left): the statistics of indoor and outdoor data. (right): the statistics of data with different supervision categories.

### 1.3. Evaluation Setup

DA3D is designed to evaluate zero-shot generalization in both novel object categories and novel camera configurations. We define two evaluation settings:

**Zero-Shot Categories.** Following prior work [10], we select partial categories from KITTI, SUNRGBD, and ARKitScenes as unseen classes for zero-shot testing.

**Zero-Shot Datasets.**

- We use Cityscapes3D, Waymo, and 3RScan as unseen datasets with novel camera configurations.
- Cityscapes3D & Waymo introduce new intrinsics and image styles, challenging models to generalize across different camera setups.
- 3RScan not only introduces novel camera setups, but also contains unseen object categories, making it useful for testing both category and camera generalization.

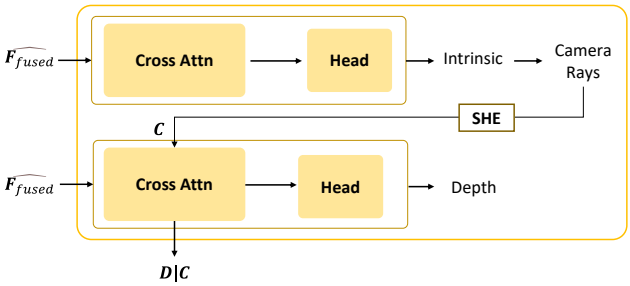


Figure 3. Detailed implementation of camera and depth module from UniDepth.

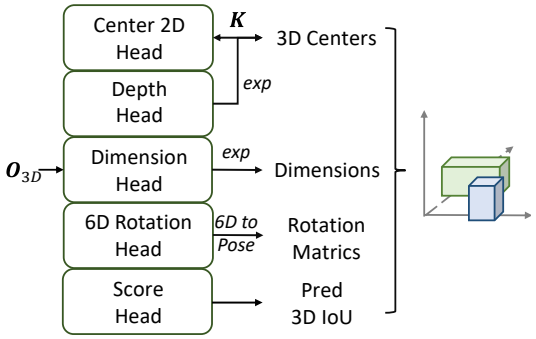


Figure 4. 3D Box head details.

## 2. Model Details

### 2.1. Camera and Depth Module Details

This section introduces how the camera module and depth module work, predicting intrinsic and camera-aware depth, also related feature.

As show in Figure 3, the fused feature $\hat{\mathbf{F}}_{\text{fused}}$ are input into the camera module, which uses a cross-attention mechanism and a to obtain the camera intrinsic parameters. These intrinsic parameters are then used to generate camera rays. The rays are defined as:

$$(r_1, r_2, r_3) = \mathbf{K}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

where $\mathbf{K}$ is the calibration matrix, $u$ and $v$ are the pixel coordinates, and 1 is a vector of ones. In this context, the homogeneous camera rays $(r_x, r_y)$ are derived from:

$$\left( \frac{r_1}{r_3}, \frac{r_2}{r_3} \right)$$

This dense representation of the camera rays undergoes Laplace Spherical Harmonic Encoding (SHE) [6] to produce the embeddings $\mathbf{C}$. These embeddings are then passed to the depth module using the cross-attention mechanism.

The depth feature conditioned on the camera embeddings, is computed as:

$$\mathbf{D}|\mathbf{C} = \texttt{MLP}(\texttt{CrossAttn}(\mathbf{D}, \mathbf{C}))$$

Subsequently, the depth feature is processed through an upsampling head to predict the final depth map.

### 2.2. 3D Box Head Details

This section introduces the details of the 3D box head. After the query $\mathbf{Q}$ passes through the Geometric Transformer and Two-Way Transformer, the model outputs $\mathbf{O}$. $\mathbf{O}$ contains outputs corresponding to both 3D-related hidden states $\mathbf{O}_{3D}$ and prompt hidden states $\mathbf{O}_p$. We extract the 3D-related output $\mathbf{O}_{\text{3D}}$ for further processing.

Subsequently, $\mathbf{O}_{\text{3D}}$ is passed through a series of prediction heads as show in Figure 4.

We then transform these predictions into the final 3D bounding box parameters and obtain the 3D bounding box $(x, y, z, w, h, l, R, S)$ for each detected object, where $(x, y, z)$ denotes the 3D center, $(w, h, l)$ represent the dimensions, and $(R, S)$ describe the rotation and predicted 3D IoU score.

### 2.3. Loss Details

**Depth Loss.** The depth module is supervised using the Scale-Invariant Logarithmic (SILog) loss, defined as:

$$\mathcal{L}_{\text{depth}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \Delta d_i{}^2 - 0.15 \cdot \left( \frac{1}{N}\sum_{i=1}^{N} \Delta d_i \right)^2} \quad (1)$$

where $\Delta d_i = \log(d_i^{\text{pred}}) - \log(d_i^{\text{gt}})$, and $N$ is the number of valid depth pixels.

**Camera Intrinsic Loss.** The camera error is computed with the dense camera rays. For an image with height $H$ and width $W$, the intrinsic loss is formulated as:

$$\mathcal{L}_{\text{cam}} = \sqrt{\frac{1}{HW}\sum_{i=1}^{HW} \Delta r_i{}^2 - 1 \cdot \left( \frac{1}{HW}\sum_{i=1}^{HW} \Delta r_i \right)^2} \quad (2)$$

where $\Delta r_i = r_i^{\text{pred}} - r_i^{\text{gt}}$.

**Detection Loss.** The detection loss consists of three components:

- Smooth L1 loss for box regression, covering the prediction of center, depth, and dimensions.
- Chamfer loss for rotation matrix prediction, ensuring accurate orientation estimation.

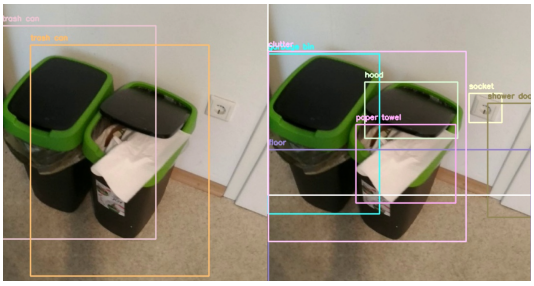


Figure 5. An example on 3RScan. The left image shows the original 3RScan annotations, while the right image presents the detection results from Grounding DINO after feeding in all the 3RScan labels. Severe naming ambiguities (e.g., "trash can" *vs*. "rubbish bin") and missing annotations lead to a substantial decrease in the detector's performance.

- Mean squared error (MSE) loss for 3D IoU score prediction, which optimizes the confidence estimates of detected objects.

Combining these terms, the total detection loss is:

$$\mathcal{L}_{\text{det}} = \mathcal{L}_{\text{box}} + \mathcal{L}_{\text{rot}} + \mathcal{L}_{\text{iou}}, \tag{3}$$

## 3. Target-aware Metrics

In our work, we evaluate both traditional metrics and the target-aware metrics proposed by OVMono3D [10]. Under the target-aware paradigm, rather than prompting the model with all possible classes from an entire dataset, we only prompt it with the classes present in the *current* image during inference. This is designed to address two key challenges encountered:

- **Missing annotations:** Comprehensive 3D annotation is often impractical or prohibitively expensive, leading to incomplete ground-truth annotations.
- **Naming ambiguity:** Datasets may label the same objects with inconsistent category names or annotation policies, creating confusion when merging datasets.

As illustrated in Figure 5, these issues are especially pronounced in the 3RScan [9] dataset. The left side shows the official 3RScan annotations, while the right side shows detections from Grounding DINO, which are largely misaligned with the dataset's labeling conventions. Consequently, traditional evaluation metrics may yield misleading or inconsistent results, whereas target-aware metrics help mitigate these mismatches by restricting the evaluated classes to those actually present in the scene.

## 4. More Ablation Study

### 4.1. Various Prompts Performance

In this section, we evaluate different types of prompts, including box prompts, point prompts, and text prompts, both with and without intrinsic prompts. The results on Omni3D are presented in Table 1. Each prompt type demonstrates its effectiveness in guiding 3D detection. Besides, on the zero-shot datasets, we observe that omitting intrinsic prompts leads to a significant performance drop (even approaching zero), which further highlights the critical role of intrinsic prompts for reliable depth calibration in unseen scenarios.

Table 1. Various Prompt Performance.

| Prompt Type | Box | Point | Text |
|---|---|---|---|
| w/ Intrinsic Prompt | 34.38 | 25.19 | 22.31 |
| w/o Intrinsic Prompt | 32.16 | 24.0 | 21.02 |

Table 2. Ablation on different backbones. The table reports $\text{AP}_{\text{3D}}$ scores. We verify the effectiveness of SAM and DINO along two dimensions: (1) whether or not we use the pretrained SAM parameters, and (2) whether adopt the pretrained DINO backbone or ConvNeXt for the depth module.

| Backbone | w/ SAM | w/o SAM |
|---|---|---|
| DINO | 25.80 | 19.12 |
| ConvNeXt | 23.11 | 18.27 |

### 4.2. Ablation on Different Backbones

In this section, we investigate our choice of backbone by comparing the use of *SAM* and *DINO* backbones. For DINO, we replace it with ConvNeXt and adopt the same pretraining method proposed by UniDepth. For SAM, we examine its effect by removing the SAM-pretrained weights and training from scratch. As shown in Table 2, SAM's pretrained parameters prove crucial for boosting performance. Meanwhile, compared to ConvNeXt, DINO offers richer geometric representations, resulting in stronger 3D detection performance.

### 4.3. Ablation on DA3D Dataset

We ablate the impact of the DA3D dataset in Tab. 3. The additional data in DA3D primarily improves generalization to novel cameras, as Omni3D contains only two distinctive intrinsics for outdoor scenes.

Table 3. Ablation on training datasets. Unless specified, all models are trained on the Omni3D dataset. For the in-domain setting, prompts are provided by Cube R-CNN, while prompts for novel classes and novel datasets are generated by Grounding DINO.

| **Method** | **In-domain** | **Novel Class** | | **Novel Camera** | |
|---|---|---|---|---|---|
| | $\text{AP}_{\text{3D}}^{\text{omni3d}}$ | $\text{AP}_{\text{3D}}^{\text{kit}}$ | $\text{AP}_{\text{3D}}^{\text{sun}}$ | $\text{AP}_{\text{3D}}^{\text{city}}$ | $\text{AP}_{\text{3D}}^{\text{3rs}}$ |
| Cube R-CNN | 23.26 | - | - | 8.22 / - | - |
| OVMono3D | 22.98 | 4.71 / 4.71 | 4.07 / 16.78 | 5.88 / 10.98 | 0.37 / 8.48 |
| DetAny3D | **24.33** | **23.75 / 23.75** | **7.63 / 20.87** | **8.31 / 11.68** | **0.64 / 9.56** |
| $\text{DetAny3D}_{\text{DA3D}}$ | 24.92 | 25.73 / 25.73 | 7.63 / 21.07 | 11.05 / 15.71 | 0.65 / 9.58 |

### 4.4. Ablation on Inference Speed

We compare the inference speed of DetAny3D with prior methods in Table 4. DetAny3D runs at 1.5 FPS on a single KITTI image, which is slower than Cube R-CNN (33.3 FPS) and OVMono3D (7.1 FPS). This is a trade-off for stronger generalization across novel categories and cameras, as DetAny3D is designed as a foundation model rather than for real-time deployment.

Table 4. Inference speed comparison on KITTI.

| Method | Cube R-CNN | OVMono3D | DetAny3D |
|---|---|---|---|
| FPS ↑ | 33.3 | 7.1 | 1.5 |

### 4.5. Per-category Performance on Novel Classes

As shown in Table 5, we provide a detailed comparison of per-category $AP_{3D}$ on novel classes from the KITTI, SUN-RGBD, and ARKitScenes datasets between our DetAny3D and the baseline OVMono3D. DetAny3D shows consistent improvements across most categories.

## 5. Limitations

**Text Prompt Process.** Our method leverages open-vocabulary 2D detectors such as Grounding DINO to convert text prompts into 2D box prompts. While effective, this strategy may cause semantic loss, as textual nuances are not directly injected into the 3D detection pipeline. Moreover, 2D detectors are known to perform poorly under heavy occlusion or partial visibility, introducing a domain gap when transferring their outputs to 3D tasks.

**Inference Efficiency.** Although DetAny3D achieves strong generalization across novel categories and camera settings, its inference speed (1.5 FPS) is significantly slower than existing lightweight 3D detectors. This limits its applicability in latency-sensitive scenarios such as real-time robotics or autonomous driving.

**Lack of Temporal Modeling.** Our current design operates on single-frame inputs and does not utilize temporal information from video sequences. Incorporating motion cues and enforcing temporal consistency could potentially improve detection accuracy and enable better integration into downstream video-based tasks, such as video knowledge distillation and temporal grounding.

## 6. Licenses and Privacy

All data used in this work are obtained from publicly available datasets and are subject to their respective licenses.

## References


[1] Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *CVPR*, 2021. 1

[2] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Yuri Feigin, Peter Fu, Thomas Gebauer, Daniel Kurz, Tal Dimry, Brandon Joffe, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. In *NeurIPS Datasets*, 2021. 1

[3] Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. Omni3d: A large benchmark and model for 3d object detection in the wild. In *CVPR*, 2023. 1

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 1

[5] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013. 1


Table 5. Per-category target-aware $AP_{3D}$ comparison on novel classes between DetAny3D and OVMono3D.

| Category | OVMono3D | DetAny3D |
|---|---|---|
| Board | 4.83 | 6.02 |
| Printer | 16.23 | 60.22 |
| Painting | 2.80 | 5.11 |
| Microwave | 30.31 | 57.21 |
| Tray | 10.11 | 6.70 |
| Podium | 48.37 | 73.65 |
| Cart | 47.31 | 33.46 |
| Tram | 4.71 | 27.90 |
| *Easy Categories* | 20.58 | 33.79 |
| Monitor | 9.44 | 15.95 |
| Bag | 15.61 | 17.69 |
| Dresser | 29.08 | 41.75 |
| Keyboard | 9.13 | 9.52 |
| Drawers | 43.04 | 40.80 |
| Computer | 7.44 | 12.37 |
| Kitchen Pan | 9.98 | 8.70 |
| Potted Plant | 6.66 | 26.34 |
| Tissues | 12.45 | 12.95 |
| Rack | 10.21 | 9.04 |
| Toys | 5.24 | 16.14 |
| Phone | 3.89 | 4.42 |
| Soundsystem | 13.22 | 6.21 |
| Fireplace | 13.16 | 30.75 |
| *Hard Categories* | 13.47 | 18.05 |
| *All Categories* | 16.05 | 23.77 |

[6] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. In *CVPR*, 2024. 2

[7] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *ICCV*, 2021. 1

[8] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, 2015. 1

[9] Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. Rio: 3d object instance re-localization in changing indoor environments. In *ICCV*, 2019. 3

[10] Jin Yao, Hao Gu, Xuweiyi Chen, Jiayun Wang, and Zezhou Cheng. Open vocabulary monocular 3d object detection. *arXiv preprint arXiv:2411.16833*, 2024. 1, 3