WildDet3D

Scaling Promptable 3D Detection in the Wild

Weikai Huang[♥1,2] **Jieyu Zhang**[♥1,2]

Sijun Li[2] **Taoyang Jia**[2] **Jiafei Duan**[1,2] **Yunqian Cheng**[1] **Jaemin Cho**[1,2] **Matthew Wallingford**[1] **Rustin Soraki**[1,2] **Chris Dongjoo Kim**[1] **Shuo Liu**[1,2] **Donovan Clay**[1,2] **Taira Anderson**[1] **Winson Han**[1]

Ali Farhadi[1,2] **Bharath Hariharan**[3] **Zhongzheng Ren**[♥1,2,4] **Ranjay Krishna**[♥1,2]

[1]Allen Institute for AI, [2]University of Washington, [3]Cornell University, [4]UNC-Chapel Hill

♥ marks core contributors.

Models: WildDet3D

Data: WildDet3D-Data WildDet3D-Bench WildDet3D-Stereo4D-Bench

Code: https://github.com/allenai/WildDet3D

Demo: https://huggingface.co/spaces/allenai/WildDet3D

iPhone App: WildDet3D (App Store)

Websites: Technical-Website Blog

Abstract

Understanding objects in 3D from a single image is a cornerstone of spatial intelligence. A key step toward this goal is monocular 3D object detection—recovering the extent, location, and orientation of objects from an input RGB image. To be practical in the open world, such a detector must generalize beyond closed-set categories, support diverse prompt modalities, and leverage geometric cues when available. Progress is hampered by two bottlenecks: existing methods are designed for a single prompt type and lack a mechanism to incorporate additional geometric cues, and current 3D datasets cover only narrow categories in controlled environments, limiting open-world transfer. In this work we address both gaps. First, we introduce **WildDet3D**, a unified geometry-aware architecture that natively accepts text, point, and box prompts and can incorporate auxiliary depth signals at inference time. Second, we present **WildDet3D-Data**, the largest open 3D detection dataset to date, constructed by generating candidate 3D boxes from existing 2D annotations and retaining only human-verified ones, yielding over 1M images across 13.5K categories in diverse real-world scenes. WildDet3D establishes a new state-of-the-art across multiple benchmarks and settings. In the open-world setting, it achieves 22.6/24.8 AP_{3D} on our newly introduced **WildDet3D-Bench** with text and box prompts. On Omni3D, it reaches 34.2/36.4 AP_{3D} with text and box prompts, respectively. In zero-shot evaluation, it achieves 40.3/48.9 ODS on Argoverse 2 and ScanNet. Notably, incorporating depth cues at inference time yields substantial additional gains (+20.7 AP on average across settings).

1 Introduction

Understanding objects in 3D is fundamental to spatial intelligence. An agent cannot reliably navigate, manipulate, or reason about the physical world by knowing what objects are alone; it must also understand

Figure 1 Overview of WildDet3D. Given a single RGB image and an optional depth map, WildDet3D performs open-vocabulary monocular 3D object detection by accepting flexible prompt modalities—text queries, 2D point clicks, or 2D bounding boxes—and predicting full 3D bounding boxes for the specified objects. This unified framework enables interactive, open-world 3D perception across diverse scenes and thousands of object categories, supporting applications in mobile devices, robotics, and AR/VR. The model gracefully leverages additional geometric cues, *i.e.*, depth when available. To train WildDet3D for broad generalization, we also curate WildDet3D-Data, a large-scale in-the-wild dataset with approximately 1M human-verified samples spanning 13K categories.

where they are, how large they are, and how they are oriented in 3D space. This capability lies at the core of robotics, embodied AI, autonomous driving, and AR/VR use cases, where success depends on grounding perception in geometry rather than appearance alone. Despite rapid progress in open-vocabulary 2D object recognition driven by large-scale vision-language models, bringing the same flexibility to 3D remains difficult. Monocular 3D object detection—recovering the position, extent, and pose of objects from a single RGB image—is a core instance of this challenge, yet existing methods still lack the generality needed for open-world use.

What would it take to build a truly general-purpose monocular 3D detector? We argue that such a system must satisfy three requirements that are not well addressed by existing methods [6, 58, 63, 59]. First, it should generalize in the wild, where object categories are long-tailed, open-ended, and frequently unseen during training. Second, it should support multiple prompt modalities. Different downstream applications naturally call for different ways of specifying a target object: a robot may issue a language command such as "pick up the mug," an AR interface may let a user tap on a region of interest, and an upstream 2D detector may provide a bounding box to be lifted into 3D. A practical model should unify these interfaces—text, 2D points, and 2D boxes—within a single architecture rather than specialize to only one. Third, real deployments may sometimes provide extra geometric cues, such as sparse LiDAR or partial depth, which should be leveraged to improve 3D localization when available. Prior work typically tackles only a subset of these requirements. Existing open-vocabulary methods [58, 59] largely focus on text-based querying, whereas oracle-prompt methods [63, 49] assume fixed geometric inputs such as boxes; neither provides a flexible, unified framework for interactive open-world 3D perception, nor do they naturally accommodate additional depth signals at inference time.

Addressing these challenges requires advances on both the model and data sides. Handling flexible inputs

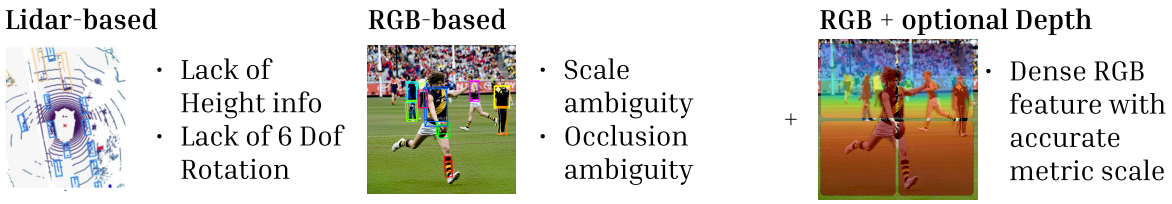


Figure 2 Input modality comparison for generalized 3D detection. LiDAR point clouds lack reliable height information and full 6-DoF rotation cues. RGB images provide dense appearance features but suffer from inherent scale and occlusion ambiguity. By combining RGB with *optional* depth, our approach retains the rich visual semantics needed for open-vocabulary recognition while reducing metric scale ambiguity when geometric signals are available.

demands a model that can unify text, points, and boxes within a single geometry-aware framework, while also incorporating partial depth cues when available. At the same time, generalization in the wild depends on training data that captures broader vocabularies, greater visual diversity, and more realistic open-world settings. We present contributions on both fronts, as highlighted in Figure 1.

On the model front, we introduce **WildDet3D**, a state-of-the-art open model for open-vocabulary monocular 3D detection. WildDet3D handles multiple prompt modalities—text, 2D points, and 2D boxes—within a single geometry-aware architecture, making it suitable for flexible and interactive open-world 3D perception. By combining strong open-vocabulary visual recognition with monocular geometry estimation, it predicts 3D bounding boxes from a single image, while also leveraging additional geometric cues such as partial depth when available. A key design question for a generalized 3D detector is the choice of input modality (Figure 2). LiDAR-based methods provide direct geometric measurements but produce sparse point clouds that lack reliable height information and full 6-DoF rotation cues, limiting their applicability to categories with well-defined upright priors. Pure RGB approaches offer dense visual features suitable for open-vocabulary recognition, yet they face inherent scale ambiguity—a small nearby object is indistinguishable from a large distant one—and occlusion ambiguity when objects overlap. We argue that *RGB with optional depth* strikes the best balance: dense appearance features support open-vocabulary, open-world recognition across arbitrary categories, while depth—when available from a LiDAR, stereo pair, or sensor—resolves metric scale without sacrificing visual richness. Crucially, by making depth *optional*, the model degrades gracefully to monocular mode rather than failing when geometric signals are absent (Section 2).

On the data front, we introduce **WildDet3D-Data**, a large-scale in-the-wild dataset for 3D detection that complements the model's generalization ability. We build it by applying existing models and methods to generate candidate 3D boxes for 2D annotations drawn from diverse 2D detection datasets [29, 44, 18, 50], and then asking human annotators to select the best qualified boxes, if any, from these candidates. This process produces a curated dataset of over 1M images covering 13.5K categories in diverse real-world scenes, substantially expanding vocabulary coverage and scene diversity for monocular 3D detection. The resulting human-verified supervision enables WildDet3D to generalize well beyond standard benchmark settings (Section 3).

Empirically, WildDet3D delivers strong performance across open-world in-the-wild evaluation, standard benchmarks, and zero-shot transfer. On WildDet3D-Bench, our in-the-wild benchmark spanning 700+ open-vocabulary categories, WildDet3D achieves 22.6 AP_{3D} with text prompts and 24.8 AP_{3D} with box prompts, far exceeding prior methods (2.3 AP for 3D-MOOD). When ground-truth depth is provided, performance reaches 41.6 AP (text) and 47.2 AP (box). On Omni3D [6], it surpasses prior methods in both text-prompt and box-prompt (oracle) settings, achieving 34.2 and 36.4 AP_{3D}, respectively, while training for only 12 epochs, compared with 80–120 epochs for competing approaches. The model also generalizes effectively across datasets: trained on Omni3D and evaluated zero-shot, it reaches 40.3 ODS on Argoverse 2 [54] and 48.9 ODS on ScanNet [12], with particularly large improvements on novel categories unseen during training. Moreover, when extra geometric cues such as sparse or ground-truth depth are available at inference time, the gains are substantial, highlighting the model's ability to flexibly benefit from richer 3D information (Section 4).

Finally, we demonstrate the versatility of WildDet3D across a range of real-world deployment scenarios, including an interactive web demo, on-device iPhone inference, AR/VR integration with Meta Quest and Meta

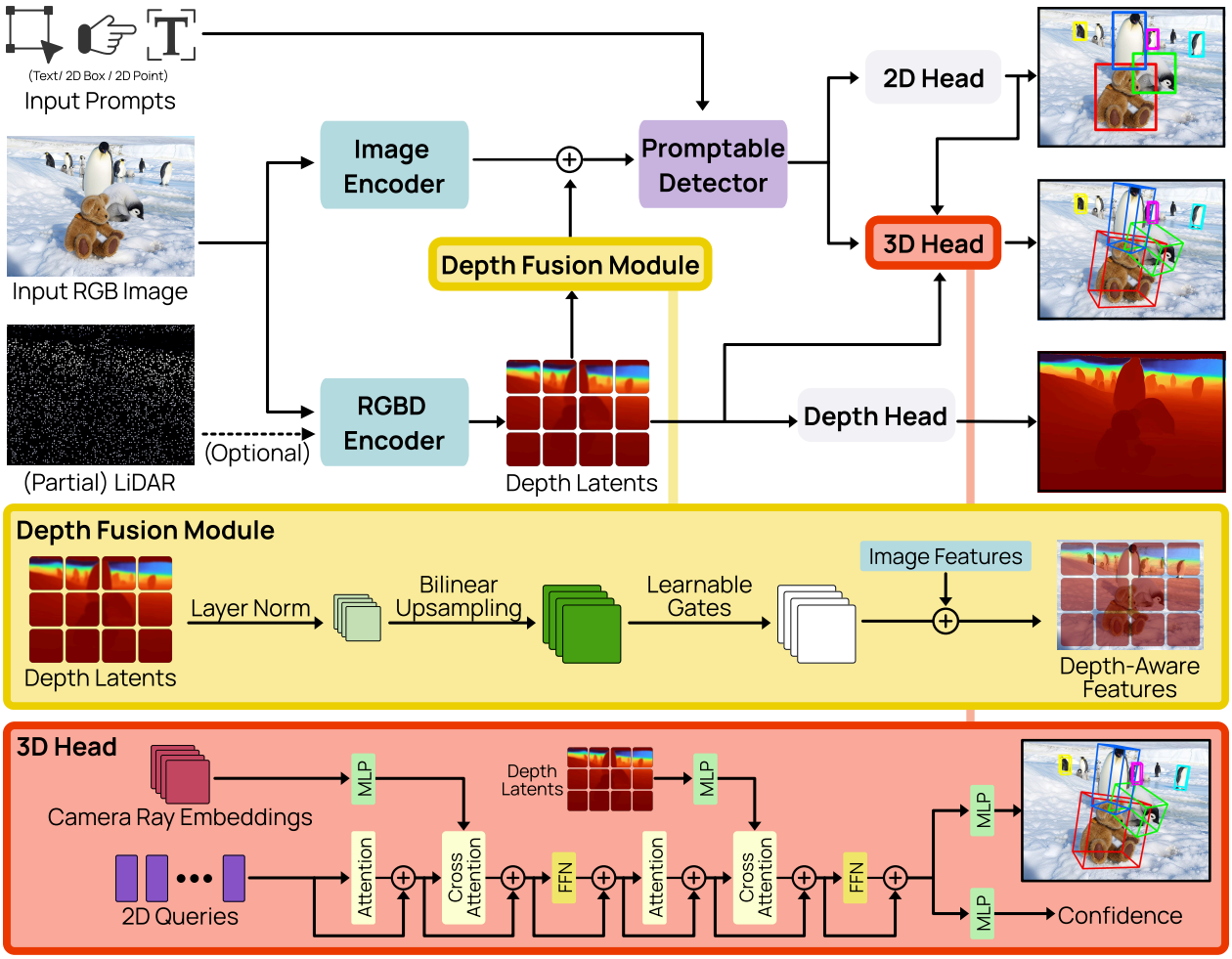


Figure 3 Overview of WildDet3D. Given an RGB image and optional depth input, dual-vision encoders (Image + RGBD) extract visual features and depth latents in parallel. The depth fusion module processes the depth latents generated by the RGBD encoder and combines it with image features from the image encoder via element-wise addition to produce enriched visual queries, which are then integrated with diverse input prompts via the promptable detector. The resulting outputs are passed through cascaded 2D and 3D Heads for open-vocabulary object detection, while the depth latents are separately decoded for depth estimation. Although our primary focus is 3D object detection, the auxiliary 2D detection and depth heads provide complementary supervision that improves overall performance.

Glasses, vision-language model integration for spatial reasoning, and robotic manipulation. These applications highlight that WildDet3D serves as a general-purpose 3D perception module that can be deployed across platforms and paired with diverse upstream systems in a plug-and-play fashion (Section 5).

2 WildDet3D

Given a single RGB image $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$, optional camera intrinsics $\mathbf{K} \in \mathbb{R}^{3\times 3}$, optional partial or full depth map $\mathbf{D} \in \mathbb{R}^{H\times W}$ (*e.g.*, from LiDAR, stereo, or ToF sensors), and a user-specified prompt $\mathcal{P}$, WildDet3D predicts a set of 3D bounding boxes $\{\mathbf{B}_i\}_{i=1}^{N}$ for all target objects in the scene. Each 3D box $\mathbf{B}_i = (\mathbf{c}_i, \mathbf{d}_i, \mathbf{R}_i, s_i)$ consists of a 3D center $\mathbf{c}_i \in \mathbb{R}^3$ in metric space (meters), physical dimensions $\mathbf{d}_i = (w, h, l) \in \mathbb{R}^3_+$ in meters, an orientation $\mathbf{R}_i \in \mathrm{SO}(3)$ with unambiguous rotation normalization, and a confidence score $s_i \in [0, 1]$. When intrinsics or depth are not provided, the model falls back to its built-in estimation modules.

An overview of the architecture of WildDet3D is shown in Figure 3. As illustrated, WildDet3D comprises three main components: (1) to accommodate optional geometric signals (*e.g.*, input depth available at inference

time), we introduce dual-vision encoders (blue blocks) and a depth fusion module (yellow block) that produce geometry-aware depth latents (Section 2.1); (2) to unify different forms of input prompts within a single architecture, we develop a promptable detector (purple block) that takes the fused vision features along with input prompts to produce unified query representations for detection heads (Section 2.2); (3) for 3D bounding box prediction, we propose a 3D detection head with unambiguous rotation normalization (red block) that aggregates multi-source information spanning depth, 2D spatial, and semantic features (Section 2.3). We further introduce auxiliary 2D detection and depth estimation heads (gray blocks) that substantially boost 3D performance while enabling broader downstream applications (Section 2.4) .

2.1 Dual-vision encoder

Accurate 3D detection from monocular images poses two intertwined challenges: the detection backbone must extract rich semantic features for recognition and localization, while the system simultaneously requires metrically grounded depth and camera-aware representations to reason in 3D. Naïvely coupling these two objectives inside a single encoder forces a trade-off—optimizing for depth can degrade detection features and vice versa—and tightly binds the architecture to one particular depth model.

We address this with a *dual-encoder* design comprising three components. First, an **image encoder** provides high-resolution, multi-scale semantic features. Second, an **RGBD encoder** operates as a pluggable geometry backend: it ingests the same image plus an optional partial or full depth map from a pretrained RGBD backbone and produces depth latents through a multi-level convolutional neck. Third, a **depth fusion module** merges the geometric cues from the RGBD encoder into the semantic feature maps of the image encoder, supplying the downstream detection head with a unified representation that is both semantically rich and metrically informed. By decoupling semantics from geometry at the encoder level and reuniting them through a dedicated fusion stage, the architecture remains modular—different depth models can be integrated without modifying the core detection pipeline.

Image encoder. The image encoder is a ViT-H [14] with a SimpleFPN neck, initialized from a segmentation-pretrained checkpoint that provides strong dense prediction features. Given an input image resized to $H \times W$ pixels with patch size p, the ViT generates $\frac{H}{p} \times \frac{W}{p}$ spatial tokens. The SimpleFPN projects these to 256-channel feature maps that are fed to the downstream detector. During training, the first 28 of 32 ViT blocks are frozen, with only the last 4 blocks fine-tuned. We adopt the image encoder architecture and weights from SAM 3 [8], which provides high-quality dense features for detection and segmentation.

RGBD encoder. The RGBD encoder is built on a DINOv2 ViT-L/14 [36] that accepts 4-channel RGBD input at 686×686 resolution (49×49 tokens), where the depth channel is optional—when no external depth is available, a zero-filled depth channel is used and RGBD encoder will generate depth feature solely based on RGB. The encoder features are passed through a ConvStack neck that produces a 5-level feature pyramid, from which we extract depth latents $\mathbf{Z}_d \in \mathbb{R}^{C_d \times 49 \times 49}$ with $C_d = 256$ via average pooling. During training, the first 21 of 24 DINOv2 blocks are frozen, with the last 3 blocks fine-tuned. To support optional depth input, training uses a stochastic strategy: 70% monocular (zero depth), 20% patch-masked depth, and 10% full depth copy-through. The RGBD encoder architecture and weights are adopted from LingBot-Depth [48], a model pretrained for metric depth estimation on large-scale RGBD data.

We deliberately use different backbones for the image and RGBD encoders: the image encoder is pretrained for segmentation, providing detection-oriented features, while the RGBD encoder is pretrained on depth completion, producing geometric features suited for metric depth estimation.

Depth fusion module. The depth fusion module (yellow block in Figure 3) injects depth latents into the image encoder's feature maps before they enter the transformer encoder, following a ControlNet-style [64] residual design. Given visual features $\mathbf{V} \in \mathbb{R}^{C \times H_v \times W_v}$ from the SimpleFPN and depth latents $\mathbf{Z}_d$, the module computes:

$$\mathbf{V}' = \mathbf{V} + \mathrm{Conv}_{1\times1}\big(\mathrm{LN}(\mathbf{Z}_d^{\uparrow})\big), \tag{1}$$

where $\mathbf{Z}_d^{\uparrow}$ denotes depth latents bilinearly interpolated to match the visual feature resolution, LN is LayerNorm that normalizes the depth latents to unit scale, and $\mathrm{Conv}_{1\times1}$ is a 1×1 convolution projecting from depth dimension to visual dimension. The convolution is *zero-initialized*, so at training start $\mathbf{V}' = \mathbf{V}$ (identity), and

the depth contribution is gradually learned without disrupting pretrained visual features. Notably, only the depth branch passes through trainable layers; the visual features are added as-is, preserving the pretrained feature distribution.

2.2 Promptable detector

The promptable detector conditions metric-depth-aware visual features on user-supplied prompts to produce per-object predictions. It accepts four complementary prompt types:

- **Text prompt.** A category name (*e.g.*, "car"), selecting all instances of that category.
- **Point prompt.** One or more 2D pixel coordinates (u, v), each labeled as positive (on the object) or negative (background), selecting the single object at that location [9, 13, 61].
- **Box prompt.** A 2D bounding box (x_1, y_1, x_2, y_2), selecting the single object within the specified region.
- **Exemplar prompt.** A 2D bounding box used as a visual exemplar, detecting all visually similar objects in the scene.

During training, all four prompt types are sampled jointly to ensure balanced learning across modalities.

Prompt encoding. We adopt SAM3's prompt encoding [8] as described below. *Text prompts* are tokenized with a CLIP-style [38] BPE tokenizer and encoded by a 24-layer causal text Transformer (width 1024, 16 heads), then linearly projected to $d = 256$. *Box and point prompts* are encoded by a geometry encoder that sums three complementary representations: (1) a direct linear projection of the coordinates, (2) ROI-aligned features pooled from the image backbone (for boxes) or grid-sampled features (for points), and (3) sinusoidal positional encoding. A learnable positive/negative label embedding is added, and the result is refined by a 3-layer Transformer with cross-attention to image features. *Exemplar prompts* reuse the same box encoding pipeline but are differentiated by a special text token ("visual") and a multi-target matching strategy that assigns all instances of the same category as ground truth. The encoded text and geometry tokens are concatenated into a single prompt sequence, which serves as cross-attention memory in both the encoder and decoder stages.

Per-prompt batching. Rather than constructing the training batch at the per-image level, we batch at the *per-prompt* level. For example, every unique text category yields a separate batch entry that aggregates all images containing that category. This strategy enables fine-grained multi-instance supervision and allows the model to handle an arbitrary number of categories per image without padding or truncation.

2.3 Deeply-supervised 3D detection head

The 3D detection head (red block in Figure 3) lifts the 2D query features produced by the promptable detector into 3D bounding-box predictions. It comprises L Transformer decoder layers, each of which outputs its own set of 3D predictions; the training losses, which will be detailed in Section 2.4, are applied at every layer with equal weights. This *deep-supervision* strategy encourages even the earliest layers to develop 3D localization capability, yielding faster convergence and more robust intermediate representations. The remainder of this section details the individual components of the head.

Multi-source information aggregation. For each decoder layer $l \in \{1, \dots, L\}$, the hidden states $\mathbf{H}^l \in \mathbb{R}^{S \times d}$ (where $d = 256$) are sequentially enriched with camera geometry and depth information through two dedicated cross-attention modules. First, a *camera prompt* branch incorporates spatial ray features. Given camera intrinsics $\mathbf{K}$, we generate per-pixel ray directions $\mathbf{r}_{i,j} = \mathbf{K}^{-1}[u, v, 1]^\top$ and encode them using 8th-order real spherical harmonics:

$$\phi(\mathbf{r}) = \mathrm{RSH}_8\left(\frac{\mathbf{r}}{\|\mathbf{r}\|}\right) \in \mathbb{R}^{81}, \tag{2}$$

where RSH_8 denotes the 8th-order spherical harmonic basis functions. The ray features are then fused via cross-attention:

$$\tilde{\mathbf{H}}^l = \mathrm{FFN}\Big(\mathrm{CrossAttn}\big(\mathrm{SelfAttn}(\mathbf{H}^l),\ f_r(\phi(\mathbf{r}))\big)\Big), \tag{3}$$

where $f_r : \mathbb{R}^{81} \to \mathbb{R}^d$ projects the spherical harmonic ray features. Then, a *depth prompt* branch fuses depth latents:

$$\hat{\mathbf{H}}^l = \mathrm{FFN}\Big(\mathrm{CrossAttn}\big(\mathrm{SelfAttn}(\tilde{\mathbf{H}}^l),\ f_d(\mathbf{Z}_d)\big)\Big), \tag{4}$$

where $f_d : \mathbb{R}^{C_d} \to \mathbb{R}^d$ is a learned projection that maps depth latents to the query embedding space. Both cross-attention modules use a single attention head, and each decoder layer has its own independent set of projection parameters.

3D box parameterization. The fused query features are passed through a two-layer MLP to predict a 12-dimensional 3D box encoding:

$$\mathbf{p}_{3\mathrm{d}} = \big(\underbrace{\Delta c_x, \Delta c_y}_{\text{center offset}}, \underbrace{\hat{d}}_{\text{log depth}}, \underbrace{\hat{w}, \hat{h}, \hat{l}}_{\text{log dims}}, \underbrace{r_1, \ldots, r_6}_{\text{rotation}} \big). \tag{5}$$

The components are defined as:

- **Center offset** $(\Delta c_x, \Delta c_y)$: the displacement between the 2D projection of the 3D center and the 2D box center, normalized by a scale factor $s_c = 10$.
- **Log-depth** $\hat{d} = s_d \cdot \log(d)$: the logarithm of the metric depth, scaled by $s_d = 2.0$.
- **Log-dimensions** $(\hat{w}, \hat{h}, \hat{l}) = s_{\mathrm{dim}} \cdot \log(w, h, l)$: the logarithm of physical dimensions in meters, scaled by $s_{\mathrm{dim}} = 2.0$.
- **6D rotation** $(r_1, \ldots, r_6)$: the first two rows of the 3×3 rotation matrix, following the continuous 6D representation [66], from which the full rotation matrix is recovered via Gram–Schmidt orthogonalization.

Unambiguous rotation normalization. Oriented 3D bounding boxes have an inherent rotation ambiguity: a box with dimensions (w, h, l) rotated by yaw θ is geometrically identical to one with swapped dimensions (l, h, w) rotated by $\theta + 90^\circ$, and similarly a 180° yaw flip yields the same box for symmetric objects. Without normalization, the model must learn multiple equivalent representations for the same 3D box, which makes training harder.

We resolve this with a two-step unambiguous rotation normalization applied to the ground-truth rotation and dimensions before loss computation: (1) *Dimension ordering*: if $w > l$, swap (w, l) and rotate by $R_y(90^\circ)$ so that $w \le l$ always holds. (2) *Yaw folding*: fold the yaw angle into $[0, \pi)$ by applying $R_y(180^\circ)$ when yaw < 0 or yaw $\ge \pi$. Together, these two steps reduce a 4-fold rotation–dimension ambiguity to a unique unambiguous form, yielding a one-to-one mapping between box geometry and the regression target. The same normalization is applied to predictions at inference time before evaluation.

At inference, the 3D center is recovered by adding the predicted offset to the 2D box center, then back-projecting through $\mathbf{K}^{-1}$ at the predicted depth $d = \exp(\hat{d}/s_d)$.

3D confidence prediction. In addition to the box regression branch, we introduce a parallel confidence branch—a two-layer MLP—that predicts a scalar 3D detection quality score $s_{3\mathrm{D}} \in [0, 1]$. During training, the soft target is defined as:

$$q^* = \beta \cdot q_{\mathrm{depth}} + (1 - \beta) \cdot \mathrm{IoU}_{3\mathrm{D}}, \tag{6}$$

where $q_{\mathrm{depth}} = \exp(-|\log \hat{d} - \log d^*|)$ measures depth prediction quality as a symmetric ratio bounded in $[0, 1]$, $\mathrm{IoU}_{3\mathrm{D}}$ is the 3D box IoU with the matched ground truth, and $\beta = 0.7$ to emphasize depth accuracy, which is the primary bottleneck in monocular 3D detection. At inference, the final detection score combines the 2D objectness score $s_{2\mathrm{D}}$ (from the IoU-aware classification head) and the 3D confidence $s_{3\mathrm{D}}$:

$$s = s_{2\mathrm{D}} + \alpha \cdot s_{3\mathrm{D}}, \tag{7}$$

with $\alpha = 0.5$. This additive formulation allows the 3D confidence to re-rank detections that have similar 2D scores but differ in geometric quality, while keeping $s_{2\mathrm{D}}$ as the dominant term so that high-confidence 2D detections are not suppressed by uncertain 3D estimates.

2.4 Multi-task learning

During training, each category in an image produces two branches of queries: (1) a *multi-target* query—sampled as 50% text-only and 50% exemplar box (with optional category label)—that is supervised against all instances of that category, and (2) a *single-target* geometric query (box or point prompt with optional category label) that is directly assigned to one selected instance. This dual-branch design ensures all prompt modalities receive supervision simultaneously.

Independently of the query branch, we employ one-to-many (O2M) matching [8]: each ground-truth object is paired with its top-k scoring predictions (k=4), providing denser supervision that accelerates convergence.

The overall training loss aggregates 3D regression, 3D confidence losses, geometry estimation loss, and 2D detection loss:

$$\mathcal{L} = \underbrace{\mathcal{L}_{\text{3D}} + \mathcal{L}_{\text{conf}}}_{\text{3D detection losses}} + \underbrace{\mathcal{L}_{\text{geom}} + \mathcal{L}_{\text{2D}}}_{\text{auxiliary losses}}. \tag{8}$$

3D regression loss $\mathcal{L}_{\text{3D}}$. For each matched prediction–target pair, we compute an L1 loss on the encoded 3D parameters (Eq. 5):

$$\mathcal{L}_{\text{3D}} = \frac{1}{N_{\text{pos}}} \sum_{i \in \mathcal{M}} \sum_{k} w_k \left| p_k^{(i)} - p_k^{*(i)} \right|, \tag{9}$$

where $\mathcal{M}$ is the set of matched indices, p_k and p_k^* are the k-th components of the predicted and target encodings, and w_k are per-component validity weights (set to zero when depth or dimensions are unavailable in the annotation).

3D confidence loss $\mathcal{L}_{\text{conf}}$. The confidence branch is trained with an IoU-aware focal BCE loss. For each matched prediction with raw logit c_i, we construct an adaptive soft target:

$$t_i = \sigma(c_i)^{\alpha} \cdot {q_i^*}^{1-\alpha}, \tag{10}$$

where $\sigma(\cdot)$ is the sigmoid function, q_i^* is the regression quality (Eq. 6), and $\alpha = 0.25$. The total confidence loss combines a positive term over matched queries and a focal-weighted negative term over unmatched queries:

$$\mathcal{L}_{\text{conf}} = \frac{1}{N_{\text{pos}}} \sum_{i \in \mathcal{M}} w_+ \cdot \text{BCE}(c_i, t_i) + \frac{1}{N_{\text{neg}}} \sum_{j \notin \mathcal{M}} \sigma(c_j)^{\gamma} \cdot \text{BCE}(c_j, 0), \tag{11}$$

where $w_+ = 5$ is a positive sample weight and $\gamma = 2$ is the focal exponent that down-weights easy negatives.

Auxiliary geometry loss $\mathcal{L}_{\text{geom}}$. The geometry backend loss aggregates below losses on the predicted depth map and camera intrinsics:

- **L1 metric depth** loss between predicted and ground-truth depth at valid pixels;
- **scale-invariant logarithmic depth** loss [15] between predicted and ground-truth depth at valid pixels;
- **confidence mask binary cross-entropy** loss that supervises a per-pixel depth validity prediction;
- **affine-invariant point-map** losses (global alignment, multi-scale local alignment, and edge-aware losses) [51] computed on back-projected 3D point maps for geometric consistency;
- **camera ray directions L2** loss that supervises the predicted intrinsics against ground-truth camera parameters.

For completeness, the detailed formulations are provided in Appendix.

Auxiliary 2D detection loss $\mathcal{L}_{\text{2D}}$. The 2D detection loss aggregates below losses:

- **IoU-aware binary cross-entropy** loss [8] for box classification, where the soft target is the 2D IoU between the predicted and ground-truth boxes;
- **box regression** combines L1 loss on the center-size representation and generalized IoU loss [41] for bounding boxes in pixel-space.
- **per-category presence** loss supervises whether a queried category exists in the image.
- **one-to-many matching** loss: each ground-truth object is paired with its top-k (k=4) scoring predictions (see above), providing denser gradient signals for both 2D and 3D heads.

For completeness, the detailed formulations are provided in Appendix.

Ignore-region suppression. A fundamental challenge in monocular 3D detection is *non-exhaustive annotation: not every visible object has a valid 3D ground truth.* For example, in both Omni3D [6] and WildDet3D-Data, objects with invalid 3D measurements, heavy truncation, severe occlusion, or placement behind the camera are annotated as `IGNORE`—they retain their 2D bounding boxes but are excluded from the set of positive 3D targets. We address this challenge in a consistent way for both evaluation and training. During **evaluation**, a prediction that matches an ignored ground-truth box is treated as *neutral*: it counts as neither a true

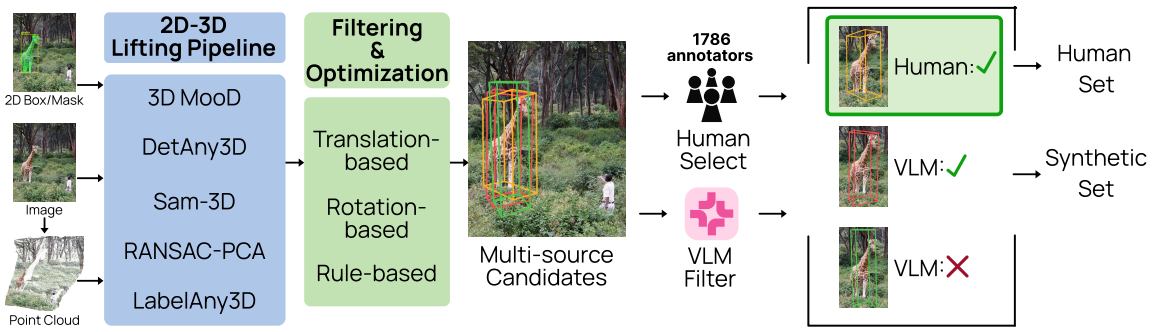


Figure 4 Overview of the WildDet3D-Data pipeline. Given an image with 2D boxes or masks and a depth-derived point cloud, five complementary models generate candidate 3D bounding boxes, shown in different colors. These candidates are refined through translation and rotation optimization, followed by rule-based filtering. The filtered candidates then enter two parallel selection branches: a VLM filtering branch, which scores each candidate on six perceptual criteria and retains those whose scores exceed a threshold, and a human annotation branch, in which annotators select the best candidate and assess its quality.

positive nor a false positive, and ignored ground truths do not contribute to the false-negative count. During **training**, we adopt the *ignore-region suppression* strategy widely used in 2D methods [29, 17, 11]. Concretely, we suppress the negative classification loss for any prediction whose 2D IoU with an ignore-annotated box exceeds 0.5 (2D IoU is used because ignore-annotated objects lack valid 3D ground truth). This ensures the training objective is consistent with the evaluation protocol, allowing the model to confidently detect objects regardless of whether their 3D annotations are available.

3 WildDet3D-Data

Existing 3D detection datasets such as Omni3D [6] provide high-quality annotations but suffer from two key limitations: (1) **limited scale**, typically covering fewer than 100 categories, and (2) **narrow domain coverage**, focusing on settings such as autonomous driving or indoor scenes. Scaling 3D detection to the open world is fundamentally more challenging than its 2D counterpart: unlike 2D bounding boxes, 3D annotations require metric depth and calibrated camera intrinsics, both of which are costly to obtain at scale.

We introduce WildDet3D-Data, a large-scale dataset for open-vocabulary 3D detection in the wild. Our dataset covers over **1M images** across **22 scene categories** (shown in Figure 6), with **3.7M valid 3D annotations**, and **13.5K object categories**—a **138×** increase in category coverage over Omni3D. To collect this dataset, we develop a three-stage pipeline: (1) multiple complementary models generate candidate 3D boxes for each existing 2D annotation (Section 3.1), (2) rule-based geometric and semantic filters remove implausible candidates (Section 3.2), and (3) human annotators or VLM-based selectors choose the best candidate and rate its quality (Section 3.3). An overview of the pipeline is shown in Figure 4. Selected dataset samples are visualized in Figure 5. More examples are available in Appendix E.

3.1 Candidates generation

Each image undergoes a multi-step processing pipeline to produce candidate 3D bounding boxes for existing 2D annotation.

Data sources. We draw 2D bounding box annotations from four large-scale detection datasets: **COCO** [29] (118K train, 5K val images), **LVIS** [18] (using COCO images with long-tail annotations over 1,200+ categories), **Objects365** [44] (609K train, 30K val images, 365 categories), and **V3Det** [50] (183K train, 30K val images, 13K+ fine-grained categories). Together these provide dense, high-quality 2D annotations with broad category and scene diversity, forming the foundation for 3D lifting.

Figure 5 Qualitative examples from WildDet3D-Data. Each pair shows 3D bounding box annotations overlaid on the input image with category labels (left) and the corresponding 3D bounding boxes rendered in the reconstructed point cloud (right). The dataset covers diverse settings including indoor scenes (desks, kitchens), outdoor environments (floating markets, streets), and animals in the wild, spanning a wide range of object categories, scales, and spatial layouts.

Monocular depth and camera estimation. We first apply 4× image super-resolution [62] to increase spatial detail for downstream point cloud generation. MoGe-2 [51] then produces a metric depth map at 1024-long-edge resolution, while PerspectiveFields [21] and WildCamera [67] estimate camera pose (roll/pitch) and intrinsics (f_x, f_y, c_x, c_y), respectively. The depth map is reprojected into a 3D point cloud using the estimated camera parameters.

Multi-model candidate generation. Five complementary methods generate candidate 3D boxes per 2D annotation, each capturing different geometric cues:

- **3D-MOOD** [58]: Runs open-vocabulary text-based detection and matches predictions to ground-truth 2D boxes via IoU.
- **DetAny3D** [63]: Directly regresses a 3D box from each 2D box using dense feature extraction.
- **SAM-3D** [49]: Reconstructs a 3D mesh from the object mask and depth-derived point map, then extracts an oriented bounding box from the mesh vertices.
- **RANSAC-PCA**: A purely geometric method that extracts object points via masks, applies statistical outlier removal and HDBSCAN clustering, then fits an oriented box using RANSAC rectangle fitting with PCA-based gravity alignment.
- **LabelAny3D** [60]: Single-image 3D reconstruction that lifts 2D crops to 3D meshes and aligns them to the scene depth.

3D box optimization. After initial prediction, each candidate undergoes two refinement steps (Figure 4, green): (1) *translation optimization*, which aligns the predicted depth to the estimated depth map using percentile-based scaling or anchor-based optimization; and (2) *rotation optimization*, which corrects orientation using PCA-based gravity alignment and 2D projection constraints. The candidates from all five models are then merged into a unified 10D format (center, dimensions, quaternion), yielding up to five candidates per 2D annotation.

3.2 Rule-based filtering

All candidates and annotations undergo multi-stage filtering. Failed annotations are never deleted but flagged as `ignore3D=1`, preserving the full 2D annotation set for recall evaluation.

Rule-based filtering. Before selection, candidates are filtered by three geometric criteria: edge contact ratio $\geq 3\%$ (box at image boundary), occlusion ratio $> 15\%$ (for the RANSAC-PCA method), or 3D-to-2D projection size ratio outside $[0.5, 1.5]$. Candidates failing any criterion are discarded before entering the annotation or

VLM selection stage.

Depicted object filter. A VLM-based classifier based on Qwen3.5-9B [2] identifies and discards annotations of *depicted* objects such as pictures, posters, reflections, or screen displays that portray objects rather than real 3D instances.

Composite image filter. Images composed of multiple sub-images are detected using Qwen3.5-9B and removed from the dataset, since they often result in inaccurate depth maps that could be confusing for the model.

LLM-estimated size and geometry filter. We use GPT-4.1-mini [35] to estimate expected physical dimensions for each object category (shortest/middle/longest axis ranges, depth-to-width ratio bounds, and a fixed/variable size classification). Annotations are then filtered in three passes:

- **Absolute size**: Each axis must fall within the LLM-estimated range. Fixed-size categories (*e.g.*, person, car) use a tight tolerance of 1.5×, while variable-size categories (*e.g.*, toy, sculpture) use 3.0× to accommodate natural size variation; both are relaxed to 2.5×/5.0× for fine-grained datasets. Flat and elongated objects skip certain axes.
- **Depth-to-width ratio**: The Z/X extent ratio must not exceed a per-category maximum, catching depth estimation stretching artifacts.
- **Axis proportion**: For non-flat, non-elongated objects, the shortest-to-middle axis ratio must be plausible.

Small object upgrade. Objects initially filtered as "small" (2D area $< 0.5\%$ of image) are re-evaluated using the same VLM criteria as synthetic selection. Qualifying small objects are upgraded to valid annotations, recovering additional long-tail supervision.

3.3 Candidate selection

We obtain final 3D annotations from generated candidates through two complementary paths: human annotation for a carefully sampled subset and VLM-based automatic selection for the remainder.

Human selection. For a balanced subset of images, crowdsourced annotators on Prolific [37] evaluates up to five candidates per object. Each candidate is visualized from four viewpoints: a perspective overlay on the original image and three orthographic point cloud views. Annotators *select the best candidate* and *rate its quality* as `good_fit`, `acceptable`, or `unacceptable`. Each batch contains 50 regular tasks plus 5 gold (quality-control) tasks with known-bad annotations; batches where annotators fail to identify $\geq 2/5$ gold tasks are discarded and reassigned to new qualified annotators until the target data quality is reached. Overall pass rates range from 84% to 98% across dataset splits.

VLM selection. For images without human annotation, we automatically select the best candidate a Molmo2 [10] checkpoint fine-tuned for this task with synthetically generated positive and negative candidate pairs from Omni3D. Each candidate is scored on six perceptual criteria: *category correctness*, *scale accuracy*, *translation accuracy*, *shape fidelity*, *rotation correctness*, and *vertical tilt alignment*. Scores range from 0 to 2 for all criteria except category correctness, which ranges from 0 to 1, giving a maximum total score of 11. Given a cropped image overlaid with the projected 3D box wireframe, the fine-tuned Molmo2 predicts structured scores for each criterion. We keep the highest-scoring candidate when its total score is greater than 10.

3.4 Statistics

Table 1 summarizes the dataset. The human-annotated portion covers ~103K images with quality ratings: across all splits, 35–48% of annotations are rated `good_fit`, 33–50% `acceptable`, and 24–39% `unacceptable` (the latter flagged as ignored). The VLM-filtered portion adds ~896K images with automatically verified annotations.

Val/test sampling strategy. For the validation and test sets, we use a three-phase balanced sampling algorithm: (1) greedy set cover for 100% category coverage, (2) multi-objective balanced fill optimizing category rarity, scene diversity, depth distribution, and source balance, and (3) targeted patching to ensure ≥ 3 samples per category.

Category coverage. Combining all sources, WildDet3D-Data spans **13,499 unique categories**. Of the 881 val

Table 1 WildDet3D-Data statistics. Human annotations are rated by crowd-source workers, while synthetic annotations are auto-selected by VLM scoring. Combined, the dataset spans 13.5K categories—a 138× increase over Omni3D's 98 categories.

Split	Source	Images	Ann.	Categories	Type	Scene	Max depth
Existing datasets							
Omni3D [6]	KITTI, nuSc., SUNRGBD, etc.	234K	3M+	98	Human	Driving, Furniture	67 m
COCO-3D [60]	COCO	18K	92K	80	Synthetic	In-the-wild	35 m
CA-1M [24]	ARKitScenes	3,500 (videos)	400K	Class-agnostic	Human	Indoor	5 m
WildDet3D-Data							
Train (Human)	COCO, LVIS, Obj365, V3Det	102,979	229,934	12,064	Human	In-the-wild	
Train (Synthetic)	COCO, LVIS, Obj365, V3Det	896,004	3,483,292	11,896	VLM filter	In-the-wild	
Val	COCO, LVIS, Obj365	2,470	9,256	785	Human	In-the-wild	
Test	COCO, LVIS, Obj365	2,433	5,596	633	Human	In-the-wild	
WildDet3D-Data (total)		**1,003,886**	**3,728,078**	**13,499**	Human + VLM	In-the-wild	**81 m**

categories, 826 (99.9%) have ≥ 1 training annotation, and ~820 have ≥ 3.

Candidate model distribution. Among valid synthetic annotations, SAM-3D contributes ~55% of selected boxes, RANSAC-PCA ~28%, and LabelAny3D ~17%, reflecting the complementary strengths of mesh-based reconstruction, geometric fitting, and single-image 3D reconstruction.

Filtering impact. The multi-stage filtering pipeline removes 15–20% of annotations via size/geometry checks, with the absolute size filter contributing the largest share. The depiction filter catches ~2% of annotations across human and synthetic splits.

3.5 Pipeline validation

To validate the annotation pipeline, we analyze the human-annotated train split (230K accepted annotations) along two axes: how candidate model quality varies, and how well VLM scoring predicts human judgment.

Candidate model quality. Table 2 (top) reports the selection share and rejection rate of each candidate model as judged by human annotators. SAM-3D accounts for the largest share of accepted annotations (40.4%), followed by RANSAC-PCA (28.2%), DetAny3D (14.5%), LabelAny3D (13.0%), and 3D-MOOD (3.8%). Rejection rates vary by more than 3× across models: RANSAC-PCA achieves the lowest rate (12.5%) while DetAny3D is rejected most frequently (42.9%). This disparity confirms that candidate quality differs substantially across models and can only be reliably distinguished through human evaluation.

VLM-human correlation. VLM scores exhibit a perfect monotonic correlation with human rejection rates (Spearman $\rho = -1.0$), as shown in Table 2 (bottom): rejection decreases steadily from 71.9% at score < 7 to 9.2% at score 11 (AUC = 0.66, point-biserial $r = 0.30$, $p < 10^{-100}$, $n = 481$K). Among the six VLM dimensions, *scale* (AUC = 0.60) and *shape* (AUC = 0.56) are the strongest predictors of human acceptance, indicating that size and geometry fidelity are the primary bottlenecks in candidate quality. Furthermore, the VLM's top-2 ranked candidates cover **73.4%** of human selections, demonstrating that VLM scoring effectively narrows the candidate space.

Limits of automatic scoring. Despite strong correlation, VLM scoring alone cannot substitute for human judgment. Even at score 10—which accounts for 64.5% of all candidates (310K)—the human rejection rate remains 16.7%. This gap motivates our two-stage design: VLM scoring as an efficient pre-filter, followed by human verification for quality-critical subsets.

4 Experiments

We first evaluate on WildDet3D-Bench, our proposed open-vocabulary in-the-wild benchmark with 700+ categories (Section 4.2), then on the standard Omni3D benchmark (Section 4.3) and zero-shot transfer to Argoverse 2 and ScanNet (Section 4.4). We further test with real depth on Stereo4D (Section 4.5) and present ablation studies (Section 4.6). All evaluations use both text-prompt and oracle (box-prompt) modes.

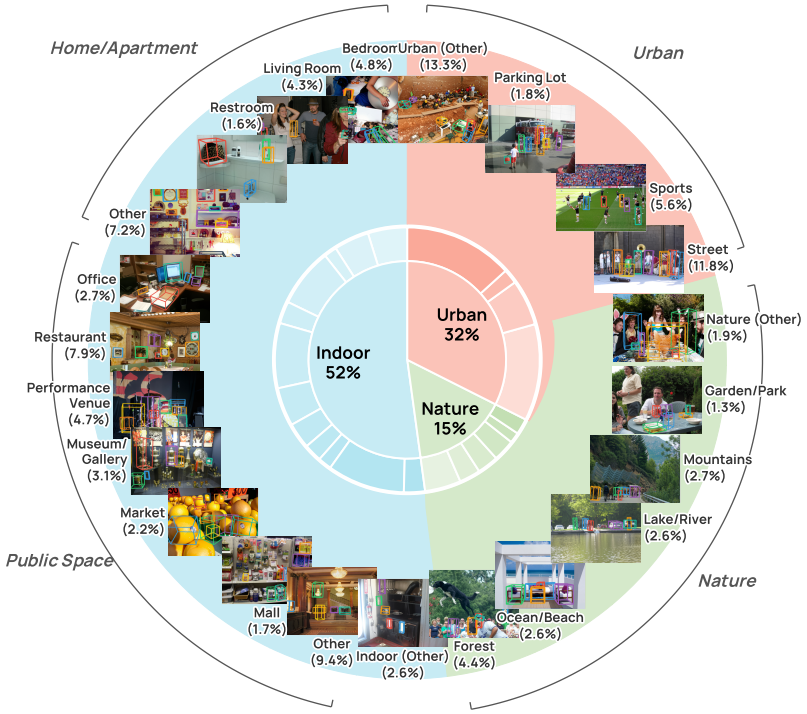


Figure 6 Scene category distribution of WildDet3D-Data. Images span three macro-categories: Indoor (52%), Urban (32%), and Nature (15%), with fine-grained sub-categories illustrated by representative examples.

Model	Sel. Share	Rej. Rate
SAM-3D	40.4%	17.3%
RANSAC-PCA	28.2%	12.5%
DetAny3D	14.5%	42.9%
LabelAny3D	13.0%	21.3%
3D-MOOD	3.8%	25.7%
Overall	—	22.0%

VLM Score	Rej. Rate	n
< 7	71.9%	1,992
7	67.4%	13,670
8	45.3%	18,665
9	36.1%	83,882
10	16.7%	310,329
11	9.2%	52,684

VLM Top-2 Coverage: **73.4%**

Table 2 Pipeline validation on the human-annotated train set. *Top:* Candidate model selection share and human rejection rate. Rejection rates vary by $> 3\times$ across models. *Bottom:* VLM composite score vs. human rejection rate. Scores correlate perfectly with human judgment.

4.1 Experimental setup

Datasets. We evaluate across four benchmarks covering diverse detection settings.

- **In-the-wild detection on WildDet3D-Bench**, which is our proposed in-the-wild benchmark as detailed in Section 4.2. WildDet3D-Bench covers 700+ open-vocabulary categories with human-verified 3D annotations.
- **Evaluation on Omni3D** [6] unifies six datasets spanning indoor and outdoor scenes: KITTI [17] (7.5K outdoor driving images), nuScenes [7] (28K multi-view driving images), SUNRGBD [45] (10K indoor RGB-D images), Hypersim [42] (100K+ synthetic indoor images), ARKitScenes [4] (55K mobile AR indoor images), and Objectron [1] (15K+ object-centric images), with a unified label space of 98 categories and standardized 3D box annotations.
- **Zero-shot detection evaluation.** Following 3D-MOOD [58], we evaluate cross-dataset generalization on Argoverse 2 [54] (outdoor driving, 26 classes) and ScanNet [12] (indoor, 18 classes), both unseen during training.
- **Real stereo depth detection on Stereo4D** [22], which provides 383 in-the-wild images with real depth maps across 78 categories, used to evaluate depth generalization zero-shot.

Evaluation metrics. We evaluate in two modes: (1) **text prompt**, where category names are used as open-vocabulary text queries for end-to-end detection, and (2) **box prompt** (oracle), where ground-truth 2D boxes serve as geometric prompts to isolate 3D regression quality. For **Omni3D**, we follow the standard protocol and report Average Precision (AP_{3D}) at 3D IoU thresholds $[0.05 : 0.50 : 0.05]$ (10 thresholds). Objects annotated as *ignore* (*e.g.*, invalid 3D annotation, heavy truncation) are excluded from both the positive ground-truth set and the false-positive count: a prediction matching an ignored object is treated as neutral (see Section 2.4). For **zero-shot transfer** on Argoverse 2 and ScanNet, we follow the 3D-MOOD protocol [58] and report the Open Detection Score (ODS), which combines AP with three error metrics into a unified score: $\text{ODS} = (3 \cdot \text{AP} + (1 - \text{mATE}) + (1 - \text{mAOE}) + (1 - \text{mASE}))/6$, where mATE, mAOE, and mASE denote mean

translation, orientation, and scale errors, respectively. We use the canonical rotation convention for mAOE, which resolves the 180° ambiguity for symmetric objects.

In-the-wild evaluation protocol. From the validation set, we construct WildDet3D-Bench, an in-the-wild evaluation benchmark spanning 700+ open-vocabulary categories from COCO, LVIS, and Objects365. Categories are split by annotation frequency into three groups: **rare** (< 5 samples, 464 categories), **common** (5–20 samples, 283 categories), and **frequent** (> 20 samples, 63 categories). For WildDet3D-Bench and Stereo4D, where open-vocabulary categories span a wide range of object sizes, we report AP using center-distance matching [58] (AP_{3D}): a prediction is matched to a ground truth if the 3D center distance is within a fraction of the object radius, with thresholds $[0.50 : 1.00 : 0.05]$. We report AP_{3D} separately for each frequency group (AP_{rare}, AP_{common}, $AP_{frequent}$) and overall. Since the annotations are not exhaustive—not every object in each image has a valid 3D bounding box—we follow the federated evaluation protocol of LVIS [18]: for text-prompt evaluation, a prediction that overlaps with a 2D-annotated object lacking a valid 3D box is treated as neutral rather than a false positive.

Implementation details. We train WildDet3D in three stages using AdamW [32] with a base learning rate of 10^{-4} and weight decay of 10^{-4}, on 4 nodes (32 GPUs) with per-GPU batch size 4 (effective batch size 128).

- **Stage 1** trains on Omni3D for 12 epochs.
- **Stage 2** fine-tunes on a mixture of Omni3D and WildDet3D-Data (human and synthetic), plus several supplementary 3D datasets collectively referred to as "Others": CA-1M [24] (indoor), Waymo [47] (driving), 3EED [27] (detection and referring expression), and FoundationPose [53] (object-level pose). These are limited to closed-set categories and specific domains but provide complementary depth ranges and scene layouts that strengthen the model's geometric estimation; open-vocabulary diversity is primarily driven by WildDet3D-Data. Stage 2 trains for 12 epochs.
- **Stage 3** further fine-tunes on Omni3D and WildDet3D-Data (human) with mask-guided point/box training for 3 epochs. The learning rate follows a multi-step decay schedule; full data mixing ratios and schedule details are in the appendix.

The SAM3 backbone is partially frozen (first 28 transformer blocks fixed). The LingBot-Depth geometry backend encoder is also partially frozen (first 21 of 24 blocks fixed, last 3 trainable). The 3D detection head is trained from scratch. Input images are resized to 1008×1008 pixels. Data augmentation includes random resizing (scale $[0.75, 1.25]$) and random horizontal flip. At test time, we apply per-category NMS with an IoU threshold of 0.6.

4.2 In-the-wild evaluation: Results on WildDet3D-Bench

We evaluate on WildDet3D-Bench (Section 4.2), which covers 700+ open-vocabulary categories with human-verified 3D annotations.

Results. Table 3 presents comprehensive in-the-wild results across prompt modes and depth settings. When trained on Omni3D only with text prompts, our method already achieves 6.8 AP, outperforming 3D-MOOD (2.3 AP) by **3.0×**. With additional training data (+ Others + WildDet3D-Data), our text-prompt result reaches **22.6 AP**—a **9.8×** improvement over 3D-MOOD.

Effect of GT depth. Providing ground-truth depth at test time dramatically improves performance. For the Omni3D-only model, text-prompt AP jumps from 6.8 to 20.7 (**+13.9**); for the full model, it increases from 22.6 to **41.6** (**+19.0**), demonstrating that our architecture effectively leverages depth signals when available.

Text vs. box prompt. Box prompts consistently outperform text prompts when no depth is provided (8.4 *vs.* 6.8 for Omni3D; 24.8 *vs.* 22.6 for +Others+WildDet3D-Data), confirming that 2D detection is a bottleneck. Interestingly, with GT depth the text-prompt setting (41.6 AP) is competitive with oracle (47.2 AP) for the full model, and both dramatically outperform all baselines.

Frequency splits. The improvements are consistent across all frequency groups, with the largest gains on rare categories ($AP_{rare} = 47.4$ *vs.* 2.4 for 3D-MOOD), demonstrating strong generalization to novel categories.

Table 3 WildDet3D-Bench evaluation. We observe that (1) WildDet3D outperforms baseline models when trained on the same data (*i.e.*, Omni3D), (2) our newly introduced WildDet3D-Data further improves performance by a significant margin (6.8→22.6, 8.4→24.8), and (3) incorporating depth input at test time nearly doubles performance (22.6→41.6, 24.8→47.2).

Method	Training data	AP_{rare}	AP_{common}	$AP_{frequent}$	AP_{3D}
Text Prompt					
3D-MOOD [58]	Omni3D	2.4	2.1	2.6	2.3
WildDet3D	Omni3D	9.0	6.5	5.2	6.8
WildDet3D w/ depth	Omni3D	23.0	21.5	16.1	20.7
WildDet3D	Omni3D, Others, WildDet3D-Data	28.3	21.6	18.7	22.6
WildDet3D w/ depth	Omni3D, Others, WildDet3D-Data	**47.4**	**40.7**	**37.2**	**41.6**
Box Prompt					
OVMono3D-LIFT [59]	Omni3D	7.4	8.8	5.1	7.7
DetAny3D [63]	Omni3D, Others	9.9	7.4	6.3	7.8
WildDet3D	Omni3D	12.0	7.9	5.3	8.4
WildDet3D w/ depth	Omni3D	26.4	24.4	19.6	23.9
WildDet3D	Omni3D, Others, WildDet3D-Data	30.0	24.2	20.3	24.8
WildDet3D w/ depth	Omni3D, Others, WildDet3D-Data	**53.7**	**46.1**	**42.5**	**47.2**

4.3 Results on Omni3D

We compare our method against several baselines: Cube R-CNN [6], a strong monocular 3D detector; UniMODE [28], a unified monocular 3D detection model; 3D-MOOD [58] with Swin-T and Swin-B backbones; and DetAny3D [63], a recent 3D detection foundation model.

Results. Table 4 reports results on Omni3D in both text prompt and oracle (box prompt) settings. With text prompts, our method achieves 34.2 AP, surpassing 3D-MOOD (28.4 AP) by **+5.8 AP** with 10× fewer training epochs (12 *vs*. 120). In the oracle setting, our method reaches 36.4 AP, outperforming DetAny3D (34.4 AP) by **+2.0 AP** despite training for only 12 epochs *vs*. 80. The gains are especially pronounced on indoor datasets: ARKitScenes and Objectron, demonstrating stronger geometry estimation in cluttered environments.

Effect of sparse depth. When sparse depth is provided at test time, performance further improves across both settings: oracle with depth reaches **45.8 AP** (**+11.4** over DetAny3D), with dramatic gains on indoor datasets equipped with depth sensors (SUNRGBD, Hypersim, ARKitScenes).

Training efficiency. A key advantage of our approach is training efficiency: we achieve superior results with 12 epochs compared to 80–120 epochs for baselines. This is enabled by the strong pre-trained representations from SAM3 and LingBot-Depth, which provide a high-quality initialization for both detection and depth estimation.

4.4 Zero-shot evaluation

To evaluate cross-dataset generalization, we train on Omni3D and test zero-shot on Argoverse 2 [54] (outdoor driving, 26 classes) and ScanNet [12] (indoor, 18 classes including novel categories unseen in Omni3D).

Results. Table 5 shows the zero-shot results. Our model achieves **40.3 ODS** on AV2 and **48.9 ODS** on ScanNet, outperforming 3D-MOOD Swin-B by **+16.5** and **+17.4** ODS respectively. The detection AP is substantially higher than all baselines: 43.4 *vs*. 14.8 on AV2 (+28.6) and 56.5 *vs*. 28.8 on ScanNet (+27.7), demonstrating strong cross-dataset generalization. Our model also achieves the best orientation estimation (mAOE): 0.526 on AV2 and 0.437 on ScanNet, significantly better than 3D-MOOD Swin-B (0.580 and 0.655). On AV2, our model also achieves the best translation accuracy (mATE = 0.714 *vs*. 0.755 for Swin-B), showing that the large AP gain does not come at the cost of localization precision.

Effect of GT depth on cross-dataset transfer. Providing ground-truth depth yields a clear improvement on

Table 4 Omni3D evaluation. Our model outperforms previous state-of-the-art methods in both text and box prompt settings. Incorporating depth input at test time further improves performance significantly.

Method	KITTI [17]	nuScenes [7]	SUNRGBD [45]	Hypersim [42]	ARKitScenes [4]	Objectron [1]	AP_{3D}
Text Prompt							
Cube R-CNN [6]	32.6	30.1	15.3	7.5	41.7	50.8	23.3
Uni-MODE* [28]	29.2	**36.0**	23.0	8.1	48.0	66.1	28.2
3D-MOOD Swin-T [58]	32.8	31.5	21.9	10.5	51.0	64.3	28.4
3D-MOOD Swin-B [58]	31.4	35.8	23.8	9.1	53.9	67.9	30.0
WildDet3D	**37.0**	31.7	38.9	16.5	64.6	60.5	34.2
WildDet3D w/ depth	36.1	32.0	**51.1**	**26.6**	**73.3**	**68.3**	**41.6**
Box Prompt							
OVMono3D-LIFT [59]	31.4	32.5	23.2	11.9	54.2	63.5	29.6
DetAny3D [63]	38.7	**37.6**	46.1	16.0	50.6	56.8	34.4
WildDet3D	**44.3**	35.3	43.1	17.3	66.6	60.8	36.4
WildDet3D w/ depth	42.8	35.9	**58.7**	**30.4**	**76.6**	**68.5**	**45.8**

Table 5 Zero-shot evaluation. ODS is the Open Detection Score [58]; higher is better. mATE, mASE, mAOE denote mean translation, scale, and orientation errors (lower is better).

	Argoverse 2 [54]					ScanNet [12]				
Method	AP↑	mATE↓	mASE↓	mAOE↓	ODS↑	AP↑	mATE↓	mASE↓	mAOE↓	ODS↑
Cube R-CNN [6]	8.6	0.903	0.867	0.953	8.9	20.0	0.733	0.774	0.921	19.5
3D-MOOD Swin-T [58]	14.8	0.782	0.697	0.612	22.5	27.3	0.630	0.726	0.650	30.2
3D-MOOD Swin-B [58]	14.7	0.755	0.680	0.580	23.8	28.8	0.612	**0.706**	0.655	31.5
WildDet3D	43.4	0.714	0.645	0.526	40.3	56.5	0.601	0.720	0.437	48.9
WildDet3D w/ depth	**43.4**	**0.701**	**0.645**	**0.526**	**40.4**	**57.6**	**0.589**	0.707	**0.422**	**50.2**

ScanNet (48.9→**50.2** ODS, +1.3), where indoor scenes benefit from accurate metric depth for resolving scale ambiguity. On AV2 the gain is marginal (40.3→40.4), suggesting that the model's monocular depth estimation is already well-calibrated for outdoor driving scenes at the scale and depth ranges present in Argoverse 2.

4.5 In-the-wild evaluation with real depth

To further validate generalization with real depth, we evaluate on Stereo4D [22], a video dataset with real stereo depth maps (383 images, 78 categories after filtering), with 2D annotations collected using the SVG2 pipeline [16]. Categories are split into rare (<5), common (5–10), and frequent (≥10) groups. AP is computed using center-distance matching.

Results. Table 6 shows zero-shot results on Stereo4D. Without depth, our monocular model (7.5 AP) is competitive with DetAny3D (7.1 AP), while OVMono3D-LIFT achieves the highest monocular AP (9.9) due to stronger monocular depth estimation on this low-resolution stereo domain. When real depth is provided, performance dramatically improves to **27.7 AP**, a 2.8× improvement over OVMono3D-LIFT (9.9 AP), demonstrating that our architecture effectively leverages real depth signals for accurate 3D localization.

4.6 Ablation

We conduct ablation studies to analyze the contribution of individual components. All ablations use the oracle (box prompt) evaluation setting on Omni3D, training on Omni3D only.

Joint 2D+3D detection. The most critical architectural choice is joint 2D and 3D prediction through a shared detection head. Removing the 2D head and predicting 3D boxes directly causes AP to collapse from 30.2 to

Table 6 Stereo4D evaluation. AP is computed using center-distance matching. All models are evaluated in a zero-shot manner, *i.e.*, not trained on Stereo4D.

Method	AP_{rare}	AP_{common}	$AP_{frequent}$	AP_{3D}
Box Prompt				
OVMono3D-LIFT [59]	<u>12.3</u>	7.1	<u>11.4</u>	<u>9.9</u>
DetAny3D [63]	8.3	<u>8.2</u>	4.9	7.1
WildDet3D	8.1	6.3	8.5	7.5
WildDet3D w/ depth	**26.2**	**31.1**	**24.6**	**27.7**

Table 7 Detection head architecture. Joint 2D+3D prediction is critical; the 3D confidence head provides complementary geometry-aware scoring. Evaluated in the oracle setting on Omni3D.

Configuration	KITTI [17]	nuScenes [7]	SUNRGBD [45]	Hypersim [42]	ARKitScenes [4]	Objectron [1]	AP_{3D}
Full model	27.9	28.2	33.9	13.2	59.4	56.8	30.2
w/o 3D confidence head	28.0	27.9	32.1	13.0	58.2	56.9	29.4 (−0.8)
w/o 2D head (3D only)	18.3	15.6	5.1	9.7	28.5	10.9	11.1 (−19.1)

11.1 (−19.1), with indoor datasets hit hardest (SUNRGBD: 33.9→5.1, Objectron: 56.8→10.9). This confirms that 2D detection provides essential spatial priors— accurate object localization in the image plane—that anchor the subsequent 3D regression. Without these priors, the model struggles to jointly localize and estimate 3D geometry from scratch.

3D confidence head. The 3D confidence head (Eq. 6) re-ranks detections using a geometry-aware score that complements 2D objectness. Removing it drops AP by 0.8 (30.2→29.4), since 2D objectness alone cannot distinguish well-localized 3D predictions from spatially inaccurate ones.

One-to-many matching. Among training objectives, one-to-many (O2M) auxiliary matching contributes the largest gain (−2.5 AP without it). The drop is most pronounced on driving datasets (KITTI: 27.9→23.2, nuScenes: 28.2→23.9), where dense, similarly-sized objects benefit most from the richer supervision signal that O2M provides during training.

Geometry loss. Explicit geometric supervision through the depth and camera-ray losses contributes −1.7 AP. The effect concentrates on indoor scenes (SUNRGBD: 33.9→28.6, Hypersim: 13.2→11.1), where accurate metric depth estimation is essential for correct 3D box placement.

Deep supervision and ignore-aware suppression. Deep supervision on intermediate decoder layers provides a modest −0.3 AP improvement. Ignore-aware suppression—which prevents the model from being penalized for detecting objects marked as ignore during evaluation—contributes −0.2 AP on Omni3D. The small magnitude is expected since ignore annotations are sparse in this benchmark; we expect a larger effect on WildDet3D-Bench where partial 3D annotations are common.

Sparse depth at test time. When sparse depth measurements from depth sensors are available at inference, they can be incorporated through the geometry backend. As shown in Table 4, this provides substantial gains, particularly on indoor datasets equipped with RGB-D sensors. The improvement of +9.4 AP (oracle: 36.4→45.8) and +7.4 AP (text: 34.2→41.6) demonstrates that our architecture gracefully accommodates additional depth signals without architectural changes.

4.7 Qualitative results

Figure 7 compares WildDet3D against OVMono3D [59] and DetAny3D [63] on four in-the-wild scenes using box prompts. On the outdoor animal scene (first scene), WildDet3D correctly localizes multiple zebras with tight 3D boxes, while DetAny3D and OVMono3D both detect unrealistic bounding box shapes. On the cluttered indoor desk (second scene), WildDet3D detects fine-grained objects such as monitors and keyboards with correct scale and orientation, whereas other methods struggle with overlapping objects and inaccurate

Table 8 Training objectives. One-to-many matching and geometry loss are the most impactful training signals; deep supervision and ignore-aware suppression provide smaller but consistent gains. Evaluated in the oracle setting on Omni3D.

Configuration	KITTI [17]	nuScenes [7]	SUNRGBD [45]	Hypersim [42]	ARKitScenes [4]	Objectron [1]	AP_{3D}
Full model	27.9	28.2	33.9	13.2	59.4	56.8	30.2
w/o O2M matching	23.2	23.9	30.8	12.2	56.8	53.5	27.7 (−2.5)
w/o geometry loss	28.3	27.7	28.6	11.1	57.0	56.4	28.5 (−1.7)
w/o deep supervision	28.1	28.3	32.4	12.6	58.7	56.6	29.9 (−0.3)
w/o ignore-aware suppression	28.2	29.4	33.2	13.0	59.2	56.4	30.0 (−0.2)

Figure 7 Qualitative comparison on in-the-wild images (box prompts). Each block shows the same scene detected by three models, all prompted using 2D bounding boxes. From top to bottom: 2D box prompt visualizations (only box prompts are used, the text labels are for reference), ground truth 3D boxes, WildDet3D predictions, OVMono3D predictions, and DetAny3D predictions, with 2D overlays and corresponding 3D bounding boxes. WildDet3D produces more accurate 3D localization and tighter boxes across diverse scenarios, including animals, vehicles, indoor electronics, and small food items.

bounding boxes. For the street scene (third scene), WildDet3D accurately captures objects at varying depths; competing methods either hallucinate large boxes (DetAny3D) or fail to estimate correct orientation (OVMono3D). On the food scene (fourth scene), WildDet3D produces well-fitting 3D boxes for individual dishes at close range, with accurate inter-object bounding box relationships (the noodles and the meatballs bounding box fit entirely inside the bowl bounding box), while competing models predict unrealistically placed objects or incorrect dimensions.

Figure 8 compares WildDet3D against 3D-MOOD [58] on four in-the-wild scenes using text prompts. WildDet3D consistently detects more object categories and more realistic object placement orientation, relative position, and shape.

These results highlight WildDet3D's advantages in multi-object scenes with diverse scales, cluttered layouts, and open-vocabulary categories. For more qualitative results see Appendix F.

5 Applications

Beyond benchmark evaluation, we demonstrate WildDet3D across a range of real-world deployment scenarios, spanning on-device mobile inference, AR headsets, vision-language model integration, and robotic manipulation.

WildDet3D in your pocket. We deploy WildDet3D on iPhone via a client-server architecture, where the iPhone captures RGB frames and LiDAR depth via ARKit and streams them to a cloud-hosted inference endpoint (Figure 9a). The APP supports multiple interaction modes: open-vocabulary text queries, 2D bounding box prompts for geometric detection, and real-time camera-based inference. Detected 3D boxes are rendered as AR overlays anchored to the physical scene using ARKit's world tracking, enabling interactive 3D perception

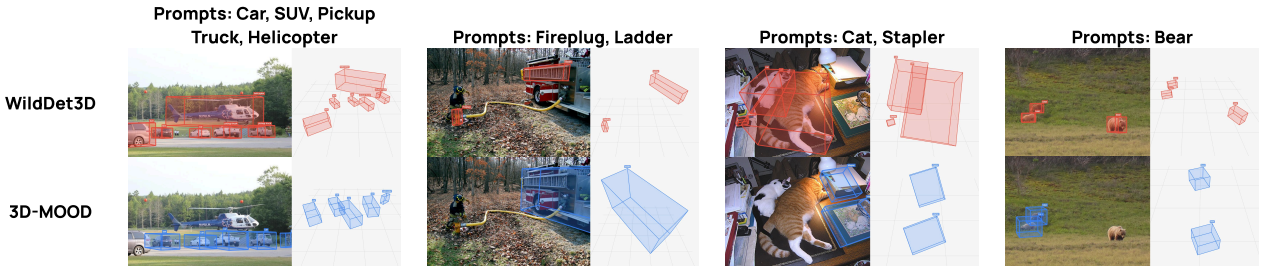
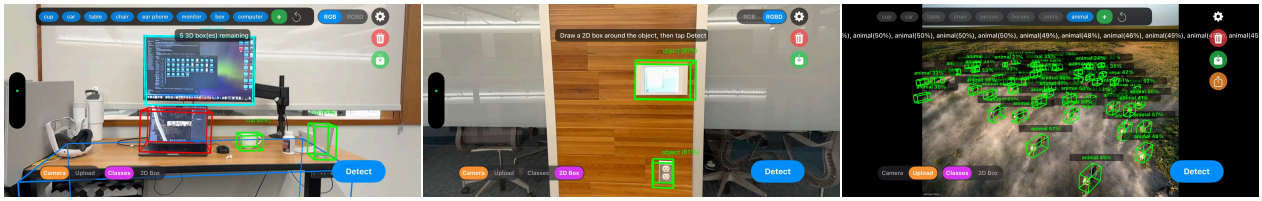
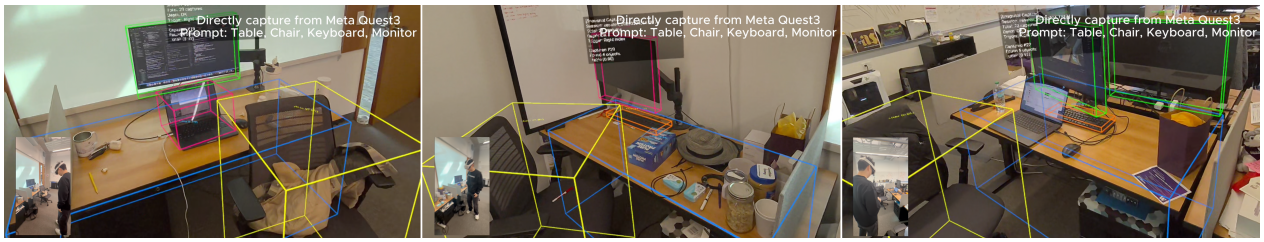


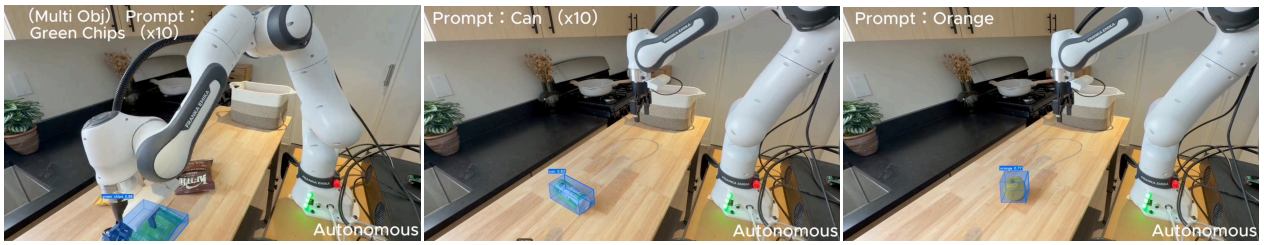
Figure 8 Qualitative comparison on in-the-wild images (text prompts). Each block shows the same scene detected by WildDet3D (top) and 3D-MOOD (bottom), prompted with text categories only.



(a) Mobile APP (iPhone) : text prompt detection in an office (left), 2D box prompt for geometric detection (middle), and open-vocabulary animal detection outdoors (right).



(b) Augmented Reality (Meta Quest 3): passthrough AR with 3D bounding boxes rendered in real time across three different desk scenes.



(c) Robotics (manipulation): Franka Emika Panda autonomously grasping objects specified by open-vocabulary text prompts ("Green Chips", "Can", "Orange").

Figure 9 Real-world deployment demos. Each row shows three frames from a different deployment platform, demonstrating WildDet3D across diverse interaction modes and environments.

on consumer hardware. The app is publicly available on the App Store.

WildDet3D for Augmented Reality (AR). We integrate WildDet3D with Meta Quest 3 to enable 3D object detection in augmented reality (Figure 9b). The Unity client captures passthrough camera frames with calibrated intrinsics and 6-DoF pose from the Quest's tracking system, sends them to the WildDet3D API, and renders detected 3D bounding boxes as overlays in the passthrough view. This enables spatial understanding for AR applications, where users can query objects in their environment by category and see metric 3D boxes anchored in physical space.

WildDet3D for Robotics. We apply WildDet3D to robotic manipulation with a Franka Emika Panda arm (Figure 9c), where accurate 3D localization is essential for grasping and interaction planning. A third-view camera captures the scene, and WildDet3D produces open-vocabulary 3D bounding boxes that are transformed into the robot's coordinate frame. The predicted box centers and dimensions are directly consumed for grasp pose generation, which will input to an IK-based interpolation planner for robot to execute, providing a zero-shot alternative to task-specific 3D perception modules that require per-object training or CAD models.

WildDet3D-agent: referring expression localization. Existing vision-language models can ground objects in 2D, but many real-world spatial questions—"what can I reach from here?" or "which object is blocking the door?"—demand 3D understanding that 2D boxes alone cannot provide. We address this by pairing WildDet3D with off-the-shelf grounded VLMs in a two-stage pipeline (Figure 10). Given a free-form query, the VLM performs open-ended reasoning and returns a 2D bounding box around the relevant object; WildDet3D then accepts this box as a geometric prompt and lifts it to a full 3D bounding box with metric depth, dimensions, and orientation. For example, when asked to "locate the most expensive object in this scene," the VLM correctly reasons and recognizes that the computer is probably most expensive and grounds it with a 2D box, after which WildDet3D produces the corresponding 3D cuboid. When VLM models such as VST or Qwen3-VL are asked to directly produce the 3D box, they both provided (rather inaccurate) boxes for the monitor instead of the computer. WildDet3D-agent can fully leverage language models' visual reasoning

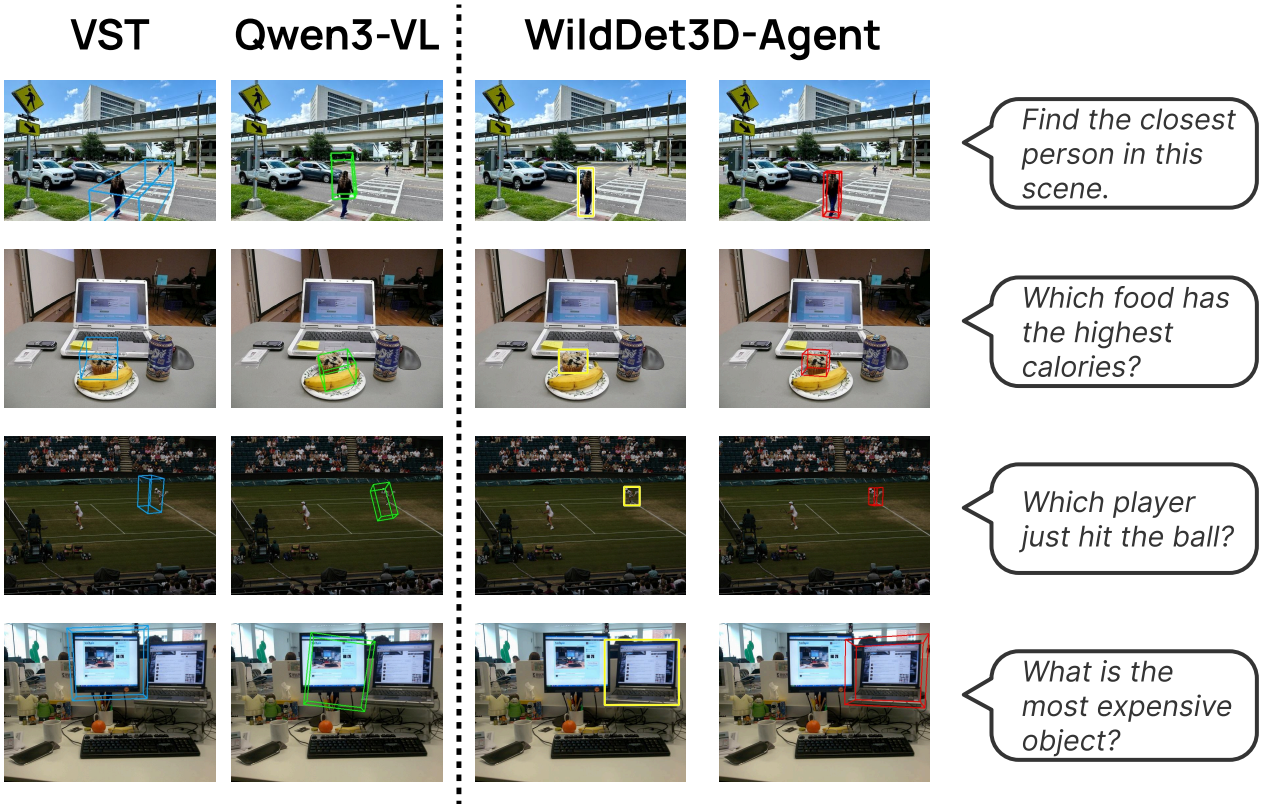


Figure 10 WildDet3D-agent: referring expression localization. Results of 3D box outputs by WildDet3D compared to VST [56] and Qwen3-VL [55]. WildDet3D-Agent more reliably localizes the queried object.

capabilities combined with WildDet3D's 3D detection ability. Because WildDet3D's box-prompt interface is model-agnostic, any VLM with grounding capability—Molmo 2 [10], Qwen3 [55], or future models—can be used as the reasoning front-end in a plug-and-play fashion, turning WildDet3D into a universal 3D lifting module that bridges high-level language reasoning with precise spatial localization.

6 Related work

Monocular 3D object detection. Monocular 3D object detection aims to recover 3D bounding boxes from a single RGB image, a fundamentally ill-posed problem due to scale ambiguity, occlusion, and missing geometric cues. Early work in this area largely focused on closed-set settings and domain-specific benchmarks such as autonomous driving [5, 31, 46, 65, 52, 40] and indoor scene understanding [43, 25, 45, 12]. Omni3D [6] took an important step toward unification by introducing a cross-dataset benchmark and model spanning multiple indoor and outdoor domains, while follow-up efforts such as UniMODE [28] further improved unified monocular 3D detection across diverse scenarios. More recent methods have improved geometric reasoning and cross-domain transfer, but most still operate in restricted label spaces or assume a fixed interaction mode. A recent line of work begins to extend monocular 3D detection toward open-set or open-vocabulary scenarios. Open Vocabulary Monocular 3D Object Detection [59], 3D-MOOD [58], and OVM3D-Det [19] explore lifting open-vocabulary 2D detections into 3D and show promising generalization beyond fixed category vocabularies. Other methods, such as DetAny3D [63], emphasize promptable 3D box prediction from localized 2D regions. These approaches establish strong baselines, but they typically specialize to a single prompt interface or supervision pipeline: text-query methods are well suited for category-level retrieval, while box-conditioned methods assume oracle or externally provided 2D localization. In contrast, we target a more general setting in which a user may specify an object by text, a point click, or a 2D box, and where additional geometric signals such as sparse depth may be available at test time.

Open-vocabulary and promptable visual perception. Our work is also related to the rapid progress in open-vocabulary 2D perception driven by vision-language pretraining. Early grounded and open-vocabulary detectors such as GLIP [26], OWL-ViT [33, 34], and Grounding DINO [30] showed that large-scale language supervision can support detection beyond fixed taxonomies. In parallel, promptable segmentation systems such as SEEM [68] and SAM 3 [8] moved visual perception toward a more interactive interface, supporting textual and geometric prompts within a unified framework. Recent multimodal LLMs have also pushed visual grounding toward more flexible language-conditioned interaction, including systems for reasoning-based segmentation and pointing such as LISA family [23, 57, 3] and the Molmo family [13, 10, 9]. We build on this trend, but move from 2D grounding and segmentation to 3D detection: rather than only predicting 2D regions, our model must infer metric center, extent, and orientation in 3D. More broadly, our work connects to interactive perception systems in which users communicate targets through flexible prompts. Existing 2D systems already support text, clicks, masks, or boxes, but comparable flexibility is rare in monocular 3D detection. Prior 3D systems usually expose either category queries or externally provided 2D boxes, which makes them less suitable for real-world human-in-the-loop applications such as robotics, AR/VR, and grounded visual question answering. Our goal is to bring the prompt flexibility of modern 2D foundation models into 3D, while preserving open-vocabulary recognition and enabling graceful improvement when depth is available.

3D annotation pipelines and open-world 3D data. A major bottleneck for generalized 3D detection is data. Compared with 2D detection, large-scale 3D box annotation is substantially more expensive because it requires metric structure, camera parameters, and careful geometric verification. Omni3D [6] provides a valuable benchmark across multiple datasets, but it still covers a limited category vocabulary and does not fully reflect the diversity of open-world imagery. More recent efforts therefore explore scalable ways to construct broader supervision, including synthetic data composition for 2D detection and grounding [20]. In the 3D setting, LabelAny3D [60] introduces an analysis-by-synthesis pipeline for producing 3D box annotations in the wild and builds COCO3D, a benchmark for open-vocabulary monocular 3D detection. Related efforts such as 3D-MOOD [58] and SAM-3D [49] further demonstrate that lifting 2D cues into 3D, or combining reconstruction with model- and human-in-the-loop annotation, can provide useful supervision at scale. However, automatic annotations remain noisy, especially for object scale, rotation, and extent. Our dataset construction pipeline builds on this insight: instead of relying on a single lifting method, we combine multiple complementary

candidate generators and then apply VLM-based scoring, human selection, and geometry-aware filtering to obtain a large-scale in-the-wild dataset for open-vocabulary 3D detection.

7 Limitations

While WildDet3D achieves strong results across diverse settings, several limitations remain.

Camera intrinsics accuracy. Our geometry backend can predict camera intrinsics when they are not provided, enabling fully uncalibrated inference. However, predicted intrinsics are less accurate than ground-truth calibration, leading to degraded 3D localization, particularly for absolute depth and physical dimensions. Closing this gap remains an open challenge for in-the-wild deployment where camera metadata is unavailable.

Single-image depth ambiguity. Monocular 3D detection is inherently ill-posed: a single image cannot fully resolve metric depth without additional cues. Our geometry backend mitigates this through learned depth priors, but performance on distant or heavily occluded objects remains limited. The substantial gains from sparse depth input (Table 4) highlight this fundamental bottleneck.

Rotation estimation. Despite unambiguous rotation normalization, rotation prediction remains the weakest component of our 3D box estimation. Objects with near-symmetric geometry (*e.g.*, round tables, square boxes) or limited visible surface area pose particular challenges, as the visual signal for orientation is inherently ambiguous.

Computational cost. The dual-backbone design (vision encoder + geometry backbone running in parallel) increases memory and compute requirements compared to 2D-only detectors. While acceptable for server-side deployment, the full model is too large for real-time on-device inference without distillation or quantization.

Long-tail categories. Performance on rare categories in open-world evaluation lags behind frequent ones. The long-tailed distribution of WildDet3D-Data partially addresses this, but categories with very few training examples still exhibit high variance in 3D prediction quality.

Intended use and deployment boundaries. The applications demonstrated in Section 5 (iPhone, AR, robotics, VLM integration) are intended as research prototypes illustrating the versatility of open-vocabulary 3D detection, not as production-ready systems. Predictions may contain incorrect depth, dimensions, or missed detections, and the model provides no guaranteed error bounds. WildDet3D is *not intended for safety-critical applications* such as autonomous navigation, surgical planning, or structural assessment.

8 Conclusion

We presented WildDet3D, an open-vocabulary monocular 3D object detector that unifies text, point, and box prompts within a single geometry-aware architecture, and WildDet3D-Data, a large-scale in-the-wild dataset spanning 1M images and 13.5K categories with human-verified 3D annotations.

On the model side, WildDet3D introduces dual vision encoders with a depth fusion module that gracefully incorporates optional depth input, an integrator that accommodates diverse prompt modalities, and a 3D detection head that aggregates depth, spatial, and semantic features. On the data side, WildDet3D-Data expands category coverage by 138× over Omni3D through a multi-model candidate generation pipeline followed by two-stage human and VLM selection, providing broad open-world supervision previously unavailable for 3D detection.

Experiments demonstrate that WildDet3D achieves state-of-the-art results on Omni3D (34.2 AP_{3D} text, 36.4 AP_{3D} oracle) with 6–10× fewer training epochs than prior methods, generalizes zero-shot to Argoverse 2 and ScanNet (40.3 and 48.9 ODS), and shows strong open-world transfer across 700+ in-the-wild categories. We further demonstrate practical deployment on iPhone, Meta Quest 3, robotic manipulation, and VLM-based spatial reasoning, showing that WildDet3D serves as a general-purpose 3D perception module across diverse platforms and applications.

Acknowledgements

This work would not be possible without the support of our colleagues at Ai2. We thank David Albright, Kristin Cha, Stephen Kelman, Yiqin Dai, Byron Bischoff, Cailin Brashear, Caleb Ouellette, David Everhart, Emily Mullen, Jon Borchardt, Crystal Nam, Patricia Balik, Tina Weiss, Kyle Wiggers, Will Smith, Peter Clark, and Noah Smith for their important work for the WildDet3D public release.

References

[1] A. Ahmadyan, L. Zhang, A. Ablavatski, J. Wei, and M. Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. *CVPR*, 2021.

[2] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, W. Ge, Z. Guo, Q. Huang, J. Huang, F. Huang, B. Hui, S. Jiang, Z. Li, M. Li, M. Li, K. Li, Z. Lin, J. Lin, X. Liu, J. Liu, C. Liu, Y. Liu, D. Liu, S. Liu, D. Lu, R. Luo, C. Lv, R. Men, L. Meng, X. Ren, X. Ren, S. Song, Y. Sun, J. Tang, J. Tu, J. Wan, P. Wang, P. Wang, Q. Wang, Y. Wang, T. Xie, Y. Xu, H. Xu, J. Xu, Z. Yang, M. Yang, J. Yang, A. Yang, B. Yu, F. Zhang, H. Zhang, X. Zhang, B. Zheng, H. Zhong, J. Zhou, F. Zhou, J. Zhou, Y. Zhu, and K. Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[3] Z. Bai, T. He, H. Mei, P. Wang, Z. Gao, J. Chen, L. Liu, Z. Zhang, and M. Z. Shou. One token to seg them all: Language instructed reasoning segmentation in videos. In *NeurIPS*, 2024.

[4] G. Baruch, Z. Chen, A. Dehghan, T. Dimry, Y. Feigin, P. Fu, T. Gebauer, B. Joffe, D. Kurz, A. Schwartz, and E. Shulman. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data, 2022. URL https://arxiv.org/abs/2111.08897.

[5] G. Brazil and X. Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *ICCV*, 2019.

[6] G. Brazil, A. Kumar, J. Straub, N. Ravi, J. Johnson, and G. Gkioxari. Omni3d: A large benchmark and model for 3d object detection in the wild, 2023. URL https://arxiv.org/abs/2207.10660.

[7] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving, 2020. URL https://arxiv.org/abs/1903.11027.

[8] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang, J. Lei, T. Ma, B. Guo, A. Kalla, M. Marks, J. Greer, M. Wang, P. Sun, R. Rädle, T. Afouras, E. Mavroudi, K. Xu, T.-H. Wu, Y. Zhou, L. Momeni, R. Hazra, S. Ding, S. Vaze, F. Porcher, F. Li, S. Li, A. Kamath, H. K. Cheng, P. Dollár, N. Ravi, K. Saenko, P. Zhang, and C. Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

[9] C. Clark, Y. Yang, J. S. Park, Z. Ma, J. Zhang, R. Tripathi, M. Salehi, S. Lee, T. Anderson, W. Han, et al. Molmopoint: Better pointing for vlms with grounding tokens. *arXiv preprint arXiv:2603.28069*, 2026.

[10] C. Clark, J. Zhang, Z. Ma, J. S. Park, M. Salehi, R. Tripathi, S. Lee, Z. Ren, C. D. Kim, Y. Yang, V. Shao, Y. Yang, W. Huang, Z. Gao, T. Anderson, J. Zhang, J. Jain, G. Stoica, W. Han, A. Farhadi, and R. Krishna. Molmo2: Open weights and data for vision-language models with video understanding and grounding, 2026. URL https://arxiv.org/abs/2601.10611.

[11] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[12] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017.

[13] M. Deitke, C. Clark, S. Lee, R. Tripathi, Y. Yang, J. S. Park, M. Salehi, N. Muennighoff, K. Lo, L. Soldaini, J. Lu, T. Anderson, E. Bransom, K. Ehsani, H. Ngo, Y. Chen, A. Patel, M. Yatskar, C. Callison-Burch, A. Head, R. Hendrix, F. Bastani, E. VanderBilt, N. Lambert, Y. Chou, A. Chheda, J. Sparks, S. Skjonsberg, M. Schmitz, A. Sarnat, B. Bischoff, P. Walsh, C. Newell, P. Wolters, T. Gupta, K.-H. Zeng, J. Borchardt, D. Groeneveld, C. Nam, S. Lebrecht, C. Wittlif, C. Schoenick, O. Michel, R. Krishna, L. Weihs, N. A. Smith, H. Hajishirzi, R. Girshick, A. Farhadi, and A. Kembhavi. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *CVPR*, 2025.

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

[15] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network, 2014. URL https://arxiv.org/abs/1406.2283.

[16] Z. Gao, J. Zhang, W. O. Ikezogwo, J. S. Park, T. G. You, D. Ogbu, C. Zheng, W. Huang, Y. Yang, W. Han, Q. Kong, R. Saini, and R. Krishna. Synthetic visual genome 2: Extracting large-scale spatio-temporal scene graphs from videos, 2026. URL https://arxiv.org/abs/2602.23543.

[17] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[18] A. Gupta, P. Dollar, and R. Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019.

[19] R. Huang, H. Zheng, Y. Wang, Z. Xia, M. Pavone, and G. Huang. Training an open-vocabulary monocular 3d detection model without 3d data. In *NeurIPS*, 2024.

[20] W. Huang, J. Zhang, T. Jia, C. Zheng, Z. Gao, J. S. Park, W. Han, and R. Krishna. Synthetic object compositions for scalable and accurate learning in detection, segmentation, and grounding, 2026. URL https://arxiv.org/abs/2510.09110.

[21] L. Jin, J. Zhang, Y. Hold-Geoffroy, O. Wang, K. Matzen, M. Sticha, and D. F. Fouhey. Perspective fields for single image camera calibration, 2023. URL https://arxiv.org/abs/2212.03239.

[22] L. Jin, R. Tucker, Z. Li, D. Fouhey, N. Snavely, and A. Holynski. Stereo4D: Learning How Things Move in 3D from Internet Stereo Videos. In *CVPR*, 2025.

[23] X. Lai, Z. Tian, Y. Chen, Y. Li, Y. Yuan, S. Liu, and J. Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023.

[24] J. Lazarow, D. Griffiths, G. Kohavi, F. Crespo, and A. Dehghan. Cubify anything: Scaling indoor 3d object detection, 2024. URL https://arxiv.org/abs/2412.04458.

[25] J. Lazarow, D. Griffiths, G. Kohavi, F. Crespo, and A. Dehghan. Cubify anything: Scaling indoor 3d object detection. In *CVPR*, 2025.

[26] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, et al. Grounded language-image pre-training. In *CVPR*, 2022.

[27] R. Li, Y. Dong, T. Hu, A. Liang, Y. Liu, D. Lu, L. Pan, L. Kong, J. Liang, and Z. Liu. 3eed: Ground everything everywhere in 3d, 2025. URL https://arxiv.org/abs/2511.01755.

[28] Z. Li, X. Xu, S. Lim, and H. Zhao. Unimode: Unified monocular 3d object detection. In *CVPR*, 2024.

[29] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft coco: Common objects in context, 2015. URL https://arxiv.org/abs/1405.0312.

[30] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[31] Z. Liu, Z. Wu, and R. Tóth. Smoke: Single-stage monocular 3d object detection via keypoint estimation. In *CVPR Workshop*, 2020.

[32] I. Loshchilov and F. Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

[33] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, et al. Simple open-vocabulary object detection. In *ECCV*, 2022.

[34] M. Minderer, A. Gritsenko, and N. Houlsby. Scaling open-vocabulary object detection. *NeurIPS*, 2023.

[35] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H. W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S. P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S. S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, Łukasz Kaiser, A. Kamali, I. Kanitscheider, N. S. Keskar, T. Khan, L. Kilpatrick, J. W. Kim, C. Kim, Y. Kim, J. H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, Łukasz Kondraciuk, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C. M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin,

K. Mayer, A. Mayne, B. McGrew, S. M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O'Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. de Avila Belbute Peres, M. Petrov, H. P. de Oliveira Pinto, Michael, Pokorny, M. Pokrass, V. H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F. P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M. B. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J. F. C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J. J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, and B. Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

[36] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. URL https://arxiv.org/abs/2304.07193.

[37] Prolific. Prolific academic online research, 2025. URL https://www.prolific.com. Accessed: March 20, 2026.

[38] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

[39] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer. Sam 2: Segment anything in images and videos. In *ICLR*, 2025.

[40] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander. Categorical depth distribution network for monocular 3d object detection. In *CVPR*, 2021.

[41] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. Generalized intersection over union: A metric and a loss for bounding box regression, 2019. URL https://arxiv.org/abs/1902.09630.

[42] M. Roberts, J. Ramapuram, A. Ranjan, A. Kumar, M. A. Bautista, N. Paczan, R. Webb, and J. M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *ICCV*, 2021.

[43] D. Rukhovich, A. Vorontsova, and A. Konushin. Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection. In *WACV*, 2022.

[44] S. Shao, Z. Li, T. Zhang, C. Peng, G. Yu, X. Zhang, J. Li, and J. Sun. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, 2019.

[45] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, 2015.

[46] Z. Song, L. Liu, F. Jia, Y. Luo, C. Jia, G. Zhang, L. Yang, and L. Wang. Robustness-aware 3d object detection in autonomous driving: A review and outlook. *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[47] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[48] B. Tan, C. Sun, X. Qin, H. Adai, Z. Fu, T. Zhou, H. Zhang, Y. Xu, X. Zhu, Y. Shen, and N. Xue. Masked depth modeling for spatial perception. *arXiv preprint arXiv:2601.17895*, 2026.

[49] S. D. Team, X. Chen, F.-J. Chu, P. Gleize, K. J. Liang, A. Sax, H. Tang, W. Wang, M. Guo, T. Hardin, X. Li, A. Lin, J. Liu, Z. Ma, A. Sagar, B. Song, X. Wang, J. Yang, B. Zhang, P. Dollár, G. Gkioxari, M. Feiszli, and J. Malik. Sam 3d: 3dfy anything in images, 2025. URL https://arxiv.org/abs/2511.16624.

[50] J. Wang, P. Zhang, T. Chu, Y. Cao, Y. Zhou, T. Wu, B. Wang, C. He, and D. Lin. V3det: Vast vocabulary visual detection dataset, 2023. URL https://arxiv.org/abs/2304.03752.

[51] R. Wang, S. Xu, Y. Dong, Y. Deng, J. Xiang, Z. Lv, G. Sun, X. Tong, and J. Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details, 2025. URL https://arxiv.org/abs/2507.02546.

[52] T. Wang, X. Zhu, J. Pang, and D. Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *CVPR*, 2021.

[53] B. Wen, W. Yang, J. Kautz, and S. Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects, 2024. URL https://arxiv.org/abs/2312.08344.

[54] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, D. Ramanan, P. Carr, and J. Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting, 2023. URL https://arxiv.org/abs/2301.00493.

[55] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Zhou, J. Lin, K. Dang, K. Bao, K. Yang, L. Yu, L. Deng, M. Li, M. Xue, M. Li, P. Zhang, P. Wang, Q. Zhu, R. Men, R. Gao, S. Liu, S. Luo, T. Li, T. Tang, W. Yin, X. Ren, X. Wang, X. Zhang, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Zhang, Y. Wan, Y. Liu, Z. Wang, Z. Cui, Z. Zhang, Z. Zhou, and Z. Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[56] R. Yang, Z. Zhu, Y. Li, J. Huang, S. Yan, S. Zhou, Z. Liu, X. Li, S. Li, W. Wang, Y. Lin, and H. Zhao. Visual spatial tuning, 2025. URL https://arxiv.org/abs/2511.05491.

[57] S. Yang, T. Qu, X. Lai, Z. Tian, B. Peng, S. Liu, and J. Jia. Lisa++: An improved baseline for reasoning segmentation with large language model. *arXiv preprint arXiv:2312.17240*, 2023.

[58] Y.-H. Yang, L. Piccinelli, M. Segu, S. Li, R. Huang, Y. Fu, M. Pollefeys, H. Blum, and Z. Bauer. 3d-mood: Lifting 2d to 3d for monocular open-set object detection, 2025. URL https://arxiv.org/abs/2507.23567.

[59] J. Yao, H. Gu, X. Chen, J. Wang, and Z. Cheng. Open vocabulary monocular 3d object detection, 2025. URL https://arxiv.org/abs/2411.16833.

[60] J. Yao, R. M. Redoy, S. Elbaum, M. B. Dwyer, and Z. Cheng. Labelany3d: Label any object 3d in the wild, 2026. URL https://arxiv.org/abs/2601.01676.

[61] W. Yuan, J. Duan, V. Blukis, W. Pumacay, R. Krishna, A. Murali, A. Mousavian, and D. Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. In *CoRL*, 2024.

[62] Z. Yue, K. Liao, and C. C. Loy. Arbitrary-steps image super-resolution via diffusion inversion, 2025. URL https://arxiv.org/abs/2412.09013.

[63] H. Zhang, H. Jiang, Q. Yao, Y. Sun, R. Zhang, H. Zhao, H. Li, H. Zhu, and Z. Yang. Detect anything 3d in the wild, 2025. URL https://arxiv.org/abs/2504.07958.

[64] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models, 2023. URL https://arxiv.org/abs/2302.05543.

[65] R. Zhang, H. Qiu, T. Wang, Z. Guo, Z. Cui, Y. Qiao, H. Li, and P. Gao. Monodetr: Depth-guided transformer for monocular 3d object detection. In *ICCV*, 2023.

[66] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the continuity of rotation representations in neural networks, 2020. URL https://arxiv.org/abs/1812.07035.

[67] S. Zhu, A. Kumar, M. Hu, and X. Liu. Tame a wild camera: In-the-wild monocular camera calibration. In *NeurIPS*, 2023.

[68] X. Zou, J. Yang, H. Zhang, F. Li, L. Li, J. Wang, L. Wang, J. Gao, and Y. J. Lee. Segment everything everywhere all at once. In *NeurIPS*, 2023.

Appendix

The appendix includes the following sections:

A Model and loss details

All geometry backend losses are scaled by a global factor $\lambda_{\text{geom}} = 5.0$. Each component is clipped to a maximum of 10 before scaling to prevent gradient explosion from outlier pixels.

A.1 Auxiliary geometry loss $\mathcal{L}_{\text{geom}}$

The geometry backend produces depth estimates, 3D point maps, a confidence mask, and camera intrinsics. The auxiliary geometry loss comprises eight terms:

Metric depth L1. A standard L1 loss between predicted and ground-truth depth at valid pixels (where $d^* > 0$ and depth ratio $\hat{d}/d^* \in [1/3, 3]$):

$$\mathcal{L}_{\text{depth-L1}} = \frac{1}{|\mathcal{V}|} \sum_{p \in \mathcal{V}} |\hat{d}_p - d_p^*|, \tag{12}$$

where $\mathcal{V}$ is the set of valid pixels. Weight: $w = 1.0$.

Scale-invariant logarithmic depth (SILog). Following Eigen *et al.* [15]:

$$\mathcal{L}_{\text{SILog}} = \sqrt{\text{Var}(g) + 0.15 \cdot \text{Mean}(g)^2}, \quad g_p = \log \hat{d}_p - \log d_p^*. \tag{13}$$

Weight: $w = 0.5$.

Affine-invariant point-map losses. We back-project predicted and ground-truth depth maps to 3D point clouds and compute three MoGe2-based [51] losses:

- *Global alignment* ($w = 10.0$): aligns the predicted point cloud to the GT via optimal affine transform at resolution 48^2;
- *Local alignment* at two scales ($w = 10.0$ each): level-4 (24^2 patches, 16 samples) and level-16 (12^2 patches, 256 samples), capturing fine-grained local geometry;
- *Edge loss* ($w = 10.0$): penalizes depth discontinuity mismatches at object boundaries.

Confidence mask BCE. A per-pixel binary cross-entropy loss that supervises the depth validity confidence prediction against a three-state ground-truth mask (finite depth, infinite/invalid depth, and unknown). For sparse depth inputs (coverage < 70%), only annotated pixels contribute to the loss. Weight: $w = 0.1$.

Camera ray MSE. An L2 loss between predicted and ground-truth camera ray directions, derived from the respective intrinsics matrices:

$$\mathcal{L}_{\text{ray}} = \text{MSE}(\mathbf{r}(\hat{\mathbf{K}}),\ \mathbf{r}(\mathbf{K}^*)), \tag{14}$$

where $\mathbf{r}(\mathbf{K})$ denotes the ray direction field generated from intrinsics $\mathbf{K}$. Weight: $w = 1.0$.

A.2 Auxiliary 2D detection loss $\mathcal{L}_{\text{2D}}$

The 2D detection losses follow the SAM 3 [8] design with minor modifications.

IoU-aware classification (BCE). For each matched prediction–target pair, the classification target is an IoU-aware soft label:

$$t = \sigma(z)^{\alpha} \cdot \mathrm{IoU}_{\mathrm{2D}}^{1-\alpha}, \quad \alpha = 0.25, \tag{15}$$

where $\sigma(z)$ is the predicted probability and $\mathrm{IoU}_{\mathrm{2D}}$ is the 2D box IoU with the matched GT. Positive predictions are weighted by $w_+ = 5$; unmatched predictions receive a focal-weighted negative loss with $\gamma = 2$. Loss weight: $w = 20$.

Box regression. Combines an L1 loss on normalized center-size coordinates ($w = 5$) and a generalized IoU loss [41] on pixel-space boxes ($w = 2$):

$$\mathcal{L}_{\mathrm{box}} = 5 \cdot \mathrm{L1}(\hat{b}_{\mathrm{cxcywh}}, b^*_{\mathrm{cxcywh}}) + 2 \cdot (1 - \mathrm{GIoU}(\hat{b}, b^*)). \tag{16}$$

Per-category presence. A sigmoid BCE loss ($w = 20$) that predicts whether each queried category has any instance in the image. Uses $\alpha = 0.5$ and $\gamma = 0$ (plain BCE without focal weighting).

One-to-many (O2M) matching. Each ground-truth box is matched to its top-k ($k = 4$) scoring predictions using a binary matcher with IoU threshold 0.4. The same classification, box, and 3D losses are computed for all matched pairs, scaled by $w_{\mathrm{o2m}} = 2.0$ and clipped at 150 to prevent gradient explosion.

B Training details

B.1 Three-stage training pipeline.

Table 9 summarizes the three training stages. All stages use AdamW with base learning rate 10^{-4}, weight decay 10^{-4}, 4 nodes (32 GPUs), and per-GPU batch size 4 (total 128). The learning rate follows a multi-step decay: for stages with N epochs, it decays by 0.1× at epochs $\lfloor N \cdot s_1 \rfloor$ and $\lfloor N \cdot s_2 \rfloor$.

Table 9 Training stage summary.

Stage	Data	Epochs	LR decay (s_1/s_2)	Init
1	Omni3D	12	2/3 / 5/6	Scratch
2	Omni3D + Others + WildDet3D-Data (H+S)	12	2/3 / 5/6	Stage 1
3	Omni3D + WildDet3D-Data (H)	3	1/3 / 2/3	Stage 2

Stage 2 data mixing ratios. Stage 2 combines seven datasets with the following sampling proportions: Omni3D 40%, CA-1M 10%, Waymo 5%, 3EED-det 2.5%, 3EED-ref 2.5%, FoundationPose 20%, and WildDet3D-Data (human + synthetic) 20%.

Stage 3 mask-guided training. Stage 3 uses Omni3D (90%) and WildDet3D-Data human annotations (10%). To leverage 2D segmentation masks from SAM 2 [39], we apply mask-guided point/box training: for images with available masks, box prompts and point prompts sampled inside the mask region are used as geometric inputs, encouraging the model to learn tighter 3D localization from precise 2D evidence.

Freeze configuration. The SAM3 ViT backbone has its first 28 transformer blocks frozen across all stages. The LingBot-Depth geometry backend uses a ViT-L encoder (24 blocks); the first 21 blocks are frozen and the last 3 remain trainable to allow the depth encoder to adapt to new data distributions. The 3D detection head is trained from scratch in all stages.

C Evaluation details

We describe the evaluation metrics and protocols used across all benchmarks.

3D IoU matching (bbox mode). For Omni3D evaluation, we use 3D bounding box IoU as the matching criterion. Predictions are matched to ground truths per category using 3D IoU, computed via oriented bounding

box overlap from the 8 corner points in camera coordinates [6]. AP is averaged over 10 IoU thresholds $\tau \in \{0.05, 0.10, \ldots, 0.50\}$, with per-threshold AP values denoted AP15, AP25, AP50 for $\tau = 0.15, 0.25, 0.50$. We also report results stratified by object depth: near (<10 m), medium (10–35 m), and far (>35 m). Maximum detections per image is capped at 100.

Center-distance matching (dist mode). For WildDet3D-Bench and Stereo4D, we use center-distance matching. A prediction is matched to a ground truth if the Euclidean distance between their 3D centers is below a threshold proportional to the object's spatial extent:

$$\|\hat{\mathbf{c}} - \mathbf{c}^*\| < \tau \cdot r, \quad r = \frac{\|\mathbf{d}^*\|_2}{2}, \tag{17}$$

where $\mathbf{d}^* = (w, h, l)$ are the GT dimensions and r is the object half-diagonal (radius). AP is averaged over 11 distance thresholds $\tau \in \{0.50, 0.55, \ldots, 1.00\}$. Depth stratification follows the same near/medium/far splits.

Open Detection Score (ODS). For zero-shot evaluation on Argoverse 2 and ScanNet, we report the Open Detection Score [58], a composite metric:

$$\text{ODS} = \frac{3 \cdot \text{AP} + (1 - \text{mATE}) + (1 - \text{mAOE}) + (1 - \text{mASE})}{6}, \tag{18}$$

where mATE is the mean translation error (center distance normalized by the matching distance threshold), mAOE is the mean absolute orientation error (normalized by π), and mASE is the mean scale error ($1 - \text{IoU}_{\text{scale}}$, where $\text{IoU}_{\text{scale}}$ is the volumetric IoU computed from axis-aligned dimension overlap). AP contributes 50% of the score (weight 3/6), encouraging both detection quality and geometric accuracy.

Frequency-split AP. On WildDet3D-Bench, we partition the 700+ evaluation categories into three groups based on the number of images containing each category: rare (<5), common (5–20), and frequent (>20), and report per-group AP (AP_r, AP_c, AP_f). This follows the LVIS [18] evaluation protocol for long-tail category distributions.

Federated evaluation. Since WildDet3D-Bench annotations are not exhaustive—not every object in each image has a valid 3D bounding box—we follow the federated evaluation protocol of LVIS [18]: a prediction that overlaps with a 2D-annotated object lacking a valid 3D box is treated as neutral rather than a false positive.

Post-processing. At test time, we apply per-category NMS with a 2D IoU threshold of 0.6. Predictions with a 2D objectness score below 0.05 are discarded before evaluation.

D Dataset details

Annotator screening. Before participating in the main annotation study, workers on Prolific complete a 10-task screening batch (~5 minutes, $1.50 reward). The screening tasks are drawn from a curated gold set and assess two independent skills: (1) **Unacceptable detection** (Filter 1): annotators must correctly flag $\geq 2/3$ known-bad annotations as `unacceptable` without mislabeling any as `good_fit`; (2) **Candidate selection accuracy** (Filter 2): annotators must select the correct best candidate on $\geq 5/7$ candidate tasks. Both filters must be passed to qualify. A total of 500 screening batches were issued; qualified workers are then invited to the main annotation batches.

3D box translation optimization. Each candidate box is aligned to the scene depth via a two-stage optimization. First, a coarse grid search evaluates $5^3 = 125$ candidate translations within a window proportional to the box dimensions (scale factor 1.0), using a combined loss:

$$\mathcal{L} = \mathcal{L}_{3D} + \lambda_{2D}\, \mathcal{L}_{2D}, \quad \lambda_{2D} = 0.5 \tag{19}$$

where $\mathcal{L}_{3D} = \lambda_{\text{in}}\, \mathcal{L}_{\text{in}} + \lambda_{\text{tight}}\, \mathcal{L}_{\text{tight}}$ (λ_{in}=1.0, λ_{tight}=0.5) combines an *inclusion loss* (anchor points should lie inside the box, buffer 0.02 m) and a *tightness loss* (box faces should be close to point cloud, buffer 0.1 m); $\mathcal{L}_{2D} = 1 - \text{GIoU}(\hat{b}_{2D}, b_{2D})$ penalizes 2D projection mismatch. The best grid point initializes an L-BFGS-B

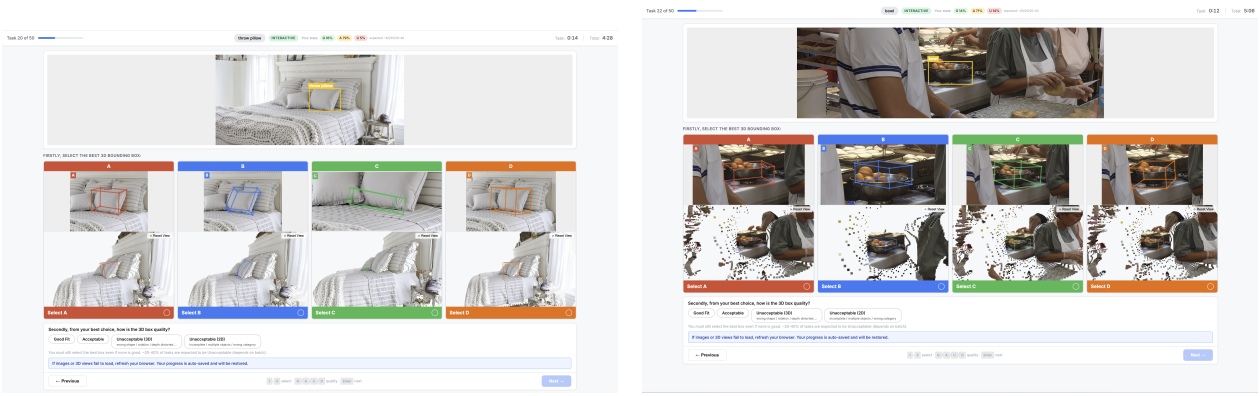


Figure 11 Annotation interface. Two example tasks from the WildDet3D-Data annotation interface. The top panel shows the reference image with the target 2D box highlighted in yellow and all candidate 3D boxes projected in color. Each candidate column (A–D, color-coded) shows a cropped 2D projection and an orthographic point cloud view. Annotators select the best candidate and rate its quality as `good_fit`, `acceptable`, `unacceptable (3D)`, or `unacceptable (2D)`.

local optimizer (max 100 iterations, f-tolerance 10^{-6}). Box dimensions and rotation are kept fixed throughout; only the 3D center is optimized. An adaptive IoU switch (threshold 0.4) selects between this optimized result and a simpler height-based scaling method (0.7×height +0.3×width scale). Anchor points (256 per box) are sampled from an eroded mask point cloud with Mahalanobis-distance weighting (α=0.5) to downweight outliers.

Val/test sampling. The three-phase balanced sampling targets the following per-split distributions. **Depth** (object-level): near (<10 m) 50%, mid (10–35 m) 25%, far (35–100 m) 20%, super-far (>100 m) 5%. **Source**: COCO 20%, LVIS 40%, Objects365 40%. Categories with fewer than 3 sampled images are marked as `rare_category` and excluded from evaluation.

Small object upgrade. Objects initially filtered as small (2D area < 0.5% of image) are re-evaluated in a separate pipeline. A candidate qualifies if: VLM score ≥ 10, category sub-score = 1, and model ∈ {LabelAny3D, SAM-3D, RANSAC-PCA}. Candidates with VLM score = 10 additionally require 3D-to-2D projected IoU ≥ 0.5 (score 11 is exempt). Per-category selection is capped at 1,500 annotations, grouping images together to avoid splitting same-image annotations.

GPT-4.1-mini category size estimation. For each object category, GPT-4.1-mini (temperature 0) is prompted to estimate the physical 3D bounding box dimensions in metres, returning six range fields (shortest/middle/longest axis min/max) plus Boolean flags `is_flat` and `is_elongated`. The system prompt instructs the model to be generous with ranges (catching clearly wrong sizes, not borderline cases), and to return a JSON object only. Flat categories (flags, plates, posters) skip the shortest-axis check; elongated categories (poles, pens, bats) skip the shortest and longest axes.

Annotator demographics. A total of 1,786 unique annotators participated through Prolific. The pool is gender-balanced (50.7% male, 49.0% female), with ages ranging from 18 to 90 (mean 42.1). Annotators are predominantly from English-speaking countries: United States (55.9%), United Kingdom (31.8%), and Canada (7.0%), consistent with the English language requirement of the task. Ethnicity is distributed as White (78.0%), Black (9.4%), Asian (5.7%), and Mixed (4.9%).

Ethics. Annotation tasks were conducted on the Prolific platform. Participants voluntarily accepted tasks and were compensated at $3.50 per batch of 55 annotations (average completion time 12–15 minutes, corresponding to an effective hourly rate of $14–17.50), meeting or exceeding the minimum wage standards of all participating countries. Demographic statistics are aggregate data provided by the platform. All studies were reviewed and approved by Prolific's platform guidelines to ensure fair treatment and ethical standards for participants. The annotator pool skews toward English-speaking Western countries (86% US/UK/CA, 78% White), which may influence judgments on what constitutes a plausible 3D box for culture-specific object categories or

unfamiliar scene layouts. Similarly, the LLM-based size filters and VLM scoring heuristics used in the pipeline inherit biases from their training data, potentially affecting which annotations are retained or rejected for underrepresented object types or scenes. We report these demographics and pipeline choices transparently so that downstream users can account for potential biases.

E More WildDet3D-Data examples

See Figure 12 for additional examples.

Figure 12 Qualitative examples from WildDet3D-Data. Each pair shows 3D bounding box annotations overlaid on the input image with category labels (left) and the corresponding 3D bounding boxes rendered in the reconstructed point cloud (right).

F Additional qualitative results

See Figure 13 and Figure 14 for additional examples.

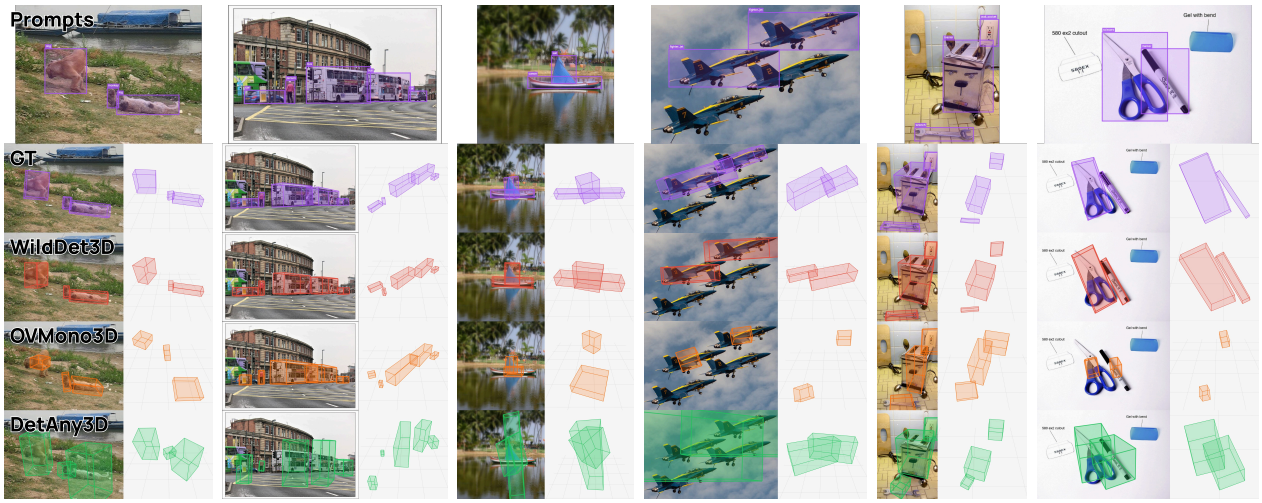


Figure 13 Box-prompted comparison. Each block shows the same scene detected by three models, all prompted using 2D bounding boxes. From top to bottom: 2D box prompt visualizations (only box prompts are used, the text labels are for reference), ground truth 3D boxes, WildDet3D predictions, OVMono3D predictions, and DetAny3D predictions, with 2D overlays and corresponding 3D bounding boxes.

Figure 14 Text-prompted comparison. Each block shows the same scene detected by WildDet3D (top) and 3D-MOOD (bottom), prompted with text categories only.