OMNI3D: A Large Benchmark and Model for 3D Object Detection in the Wild

Garrick Brazil[1]   Abhinav Kumar[2]   Julian Straub[1]   Nikhila Ravi[1]

Justin Johnson[1]   Georgia Gkioxari[3]

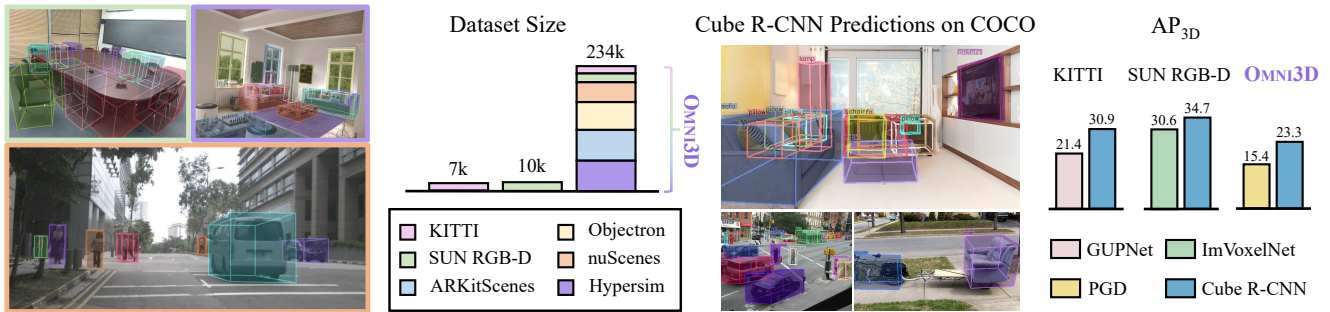[1]Meta AI   [2]Michigan State University   [3]Caltech

Figure 1. **Left**: We introduce OMNI3D, a benchmark for 3D object detection which is larger and more diverse than popular 3D benchmarks. **Right**: We propose Cube R-CNN, which generalizes to unseen datasets (*e.g.* COCO [45]) and outperforms prior works on existing datasets.

Abstract

Recognizing scenes and objects in 3D from a single image is a longstanding goal of computer vision with applications in robotics and AR/VR. For 2D recognition, large datasets and scalable solutions have led to unprecedented advances. In 3D, existing benchmarks are small in size and approaches specialize in few object categories and specific domains, e.g. urban driving scenes. Motivated by the success of 2D recognition, we revisit the task of 3D object detection by introducing a large benchmark, called OMNI3D. OMNI3D re-purposes and combines existing datasets resulting in 234k images annotated with more than 3 million instances and 98 categories. 3D detection at such scale is challenging due to variations in camera intrinsics and the rich diversity of scene and object types. We propose a model, called Cube R-CNN, designed to generalize across camera and scene types with a unified approach. We show that Cube R-CNN outperforms prior works on the larger OMNI3D and existing benchmarks. Finally, we prove that OMNI3D is a powerful dataset for 3D object recognition and show that it improves single-dataset performance and can accelerate learning on new smaller datasets via pre-training.[1]

---

[1]We release the OMNI3D benchmark and Cube R-CNN models at https://github.com/facebookresearch/omni3d.

1. Introduction

Understanding objects and their properties from single images is a longstanding problem in computer vision with applications in robotics and AR/VR. In the last decade, 2D object recognition [26,40,63,64,72] has made tremendous advances toward predicting objects on the image plane with the help of large datasets [24,45]. However, the world and its objects are three dimensional laid out in 3D space. Perceiving objects in 3D from 2D visual inputs poses new challenges framed by the task of 3D object detection. Here, the goal is to estimate a 3D location and 3D extent of each object in an image in the form of a tight oriented 3D bounding box.

Today 3D object detection is studied under two different lenses: for urban domains in the context of autonomous vehicles [7,13,50,52,57] or indoor scenes [31,36,58,73]. Despite the problem formulation being shared, methods share little insights between domains. Often approaches are tailored to work only for the domain in question. For instance, urban methods make assumptions about objects resting on a ground plane and model *only* yaw angles for 3D rotation. Indoor techniques may use a confined depth range (*e.g.* up to 6m in [58]). These assumptions are generally not true in the real world. Moreover, the most popular benchmarks for image-based 3D object detection are small. Indoor SUN RGB-D [70] has 10k images, urban KITTI [21] has 7k images; 2D benchmarks like COCO [45] are 20× larger.

We address the absence of a general large-scale dataset for 3D object detection by introducing a large and diverse 3D benchmark called OMNI3D. OMNI3D is curated from publicly released datasets, SUN RBG-D [70], ARKitScenes [6], Hypersim [65], Objectron [2], KITTI [21] and nuScenes [9], and comprises 234k images with 3 million objects annotated with 3D boxes across 98 categories including *chair, sofa, laptop, table, cup, shoes, pillow, books, car, person, etc.* Sec. 3 describes the curation process which involves re-purposing the raw data and annotations from the aforementioned datasets, which originally target different applications. As shown in Fig. 1, OMNI3D is 20× larger than existing popular benchmarks used for 3D detection, SUN RGB-D and KITTI. For efficient evaluation on the large OMNI3D, we introduce a new algorithm for intersection-over-union of 3D boxes which is 450× faster than previous solutions [2]. We empirically prove the impact of OMNI3D as a large-scale dataset and show that it improves single-dataset performance by up to 5.3% AP on urban and 3.8% on indoor benchmarks.

On the large and diverse OMNI3D, we design a general and simple 3D object detector, called Cube R-CNN, inspired by advances in 2D and 3D recognition of recent years [22, 52, 64, 68]. Cube R-CNN detects all objects and their 3D location, size and rotation end-to-end from a single image of any domain and for many object categories. Attributed to OMNI3D's diversity, our model shows strong generalization and outperforms prior works for indoor and urban domains with one unified model, as shown in Fig. 1. Learning from such diverse data comes with challenges as OMNI3D contains images of highly varying focal lengths which exaggerate scale-depth ambiguity (Fig. 4). We remedy this by operating on *virtual depth* which transforms object depth with the same virtual camera intrinsics across the dataset. An added benefit of virtual depth is that it allows the use of data augmentations (*e.g.* image rescaling) during training, which is a critical feature for 2D detection [12, 80], and as we show, also for 3D. Our approach with one unified design outperforms prior best approaches in AP$_{3D}$, ImVoxelNet [66] by 4.1% on indoor SUN RGB-D, GUPNet [52] by 9.5% on urban KITTI, and PGD [77] by 7.9% on OMNI3D.

We summarize our contributions:

- We introduce OMNI3D, a benchmark for image-based 3D object detection sourced from existing 3D datasets, which is 20× larger than existing 3D benchmarks.

- We implement a new algorithm for IoU of 3D boxes, which is 450× faster than prior solutions.

- We design a general-purpose baseline method, Cube R-CNN, which tackles 3D object detection for many categories and across domains with a unified approach. We propose *virtual depth* to eliminate the ambiguity from varying camera focal lengths in OMNI3D.

2. Related Work

Cube R-CNN and OMNI3D draw from key research advances in 2D and 3D object detection.

2D Object Detection. Here, methods include two-stage approaches [26, 64] which predict object regions with a region proposal network (RPN) and then refine them via an MLP. Single-stage detectors [44, 47, 63, 72, 85] omit the RPN and predict regions directly from the backbone.

3D Object Detection. Monocular 3D object detectors predict 3D cuboids from single input images. There is extensive work in the urban self-driving domain where the *car* class is at the epicenter [13, 19, 23, 25, 30, 46, 49, 54, 57, 62, 68, 69, 76, 78, 84]. CenterNet [85] predicts 3D depth and size from fully-convolutional center features, and is extended by [14, 37, 42, 48, 50–52, 55, 83, 84, 87]. M3D-RPN [7] trains an RPN with 3D anchors, enhanced further by [8, 18, 38, 75, 88]. FCOS3D [78] extends the anchorless FCOS [72] detector to predict 3D cuboids. Its successor PGD [77] furthers the approach with probabilistic depth uncertainty. Others use pseudo depth [4, 15, 53, 59, 79, 81] and explore depth and point-based LiDAR techniques [35, 60]. Similar to ours, [11, 68, 69] add a 3D head, specialized for urban scenes and objects, on two-stage Faster R-CNN. [25, 69] augment their training by synthetically generating depth and box-fitted views, coined as virtual views or depth. In our work, virtual depth aims at addressing *varying focal lengths*.

For indoor scenes, a vast line of work tackles room layout estimation [17, 28, 41, 56]. Huang *et al* [31] predict 3D oriented bounding boxes for indoor objects. Factored3D [73] and 3D-RelNet [36] jointly predict object voxel shapes. Total3D [58] predicts 3D boxes and meshes by additionally training on datasets with annotated 3D shapes. ImVoxelNet [66] proposes domain-specific methods which share an underlying framework for processing volumes of 3D voxels. In contrast, we explore 3D object detection in its general form by tackling outdoor and indoor domains jointly in a single model and with a vocabulary of 5× more categories.

3D Datasets. KITTI [21] and SUN RGB-D [70] are popular datasets for 3D object detection on urban and indoor scenes respectively. Since 2019, 3D datasets have emerged, both for indoor [2, 3, 6, 16, 65] and outdoor [9, 10, 20, 29, 32, 33, 71]. In isolation, these datasets target different tasks and applications and have unique properties and biases, *e.g.* object and scene types, focal length, coordinate systems, *etc.* In this work, we unify existing representative datasets [2, 6, 9, 21, 65, 70]. We process the raw visual data, re-purpose their annotations, and carefully curate the union of their semantic labels in order to build a coherent large-scale benchmark, called OMNI3D. OMNI3D is 20× larger than widely-used benchmarks and notably more diverse. As such, new challenges arise stemming from the increased variance in visual domain, object rotation, size, layouts, and camera intrinsics.
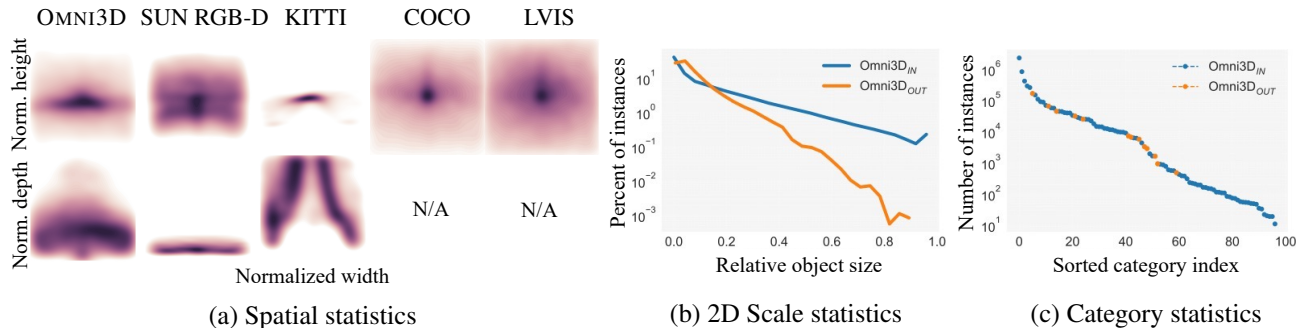
Figure 2. OMNI3D analysis. (a) distribution of object centers on (top) normalized image, XY-plane, and (bottom) normalized depth, XZ-plane, (b) relative 2D object scales, and (c) category frequency.

3. The OMNI3D Benchmark

The primary benchmarks for 3D object detection are small, focus on a few categories and are of a single domain. For instance, the popular KITTI [21] contains only urban scenes, has 7k images and 3 categories, with a focus on car. SUN RGB-D [70] has 10k images. The small size and lack of variance in 3D datasets is a stark difference to 2D counterparts, such as COCO [45] and LVIS [24], which have pioneered progress in 2D recognition.

We aim to bridge the gap to 2D by introducing OMNI3D, a large-scale and diverse benchmark for image-based 3D object detection consisting of 234k images, 3 million labeled 3D bounding boxes, and 98 object categories. We source from recently released 3D datasets of urban (nuScenes [9] and KITTI [21]), indoor (SUN RGB-D [70], ARKitScenes [6] and Hypersim [65]), and general (Objectron [2]) scenes. Each of these datasets target different applications (*e.g.* point-cloud recognition or reconstruction in [6], inverse rendering in [65]), provide visual data in different forms (*e.g.* videos in [2, 6], rig captures in [9]) and annotate different object types. To build a coherent benchmark, we process the varying raw visual data, re-purpose their annotations to extract 3D cuboids in a *unified 3D camera coordinate system*, and carefully curate the final vocabulary. More details about the benchmark creation in the Appendix.

We analyze OMNI3D and show its rich spatial and semantic properties proving it is visually diverse, similar to 2D data, and highly challenging for 3D as depicted in Fig. 2. We show the value of OMNI3D for the task of 3D object detection with extensive quantitative analysis in Sec. 5.

3.1. Dataset Analysis

Splits. We split the dataset into 175k/19k/39k images for train/val/test respectively, consistent with original splits when available, and otherwise free of overlapping video sequences in splits. We denote indoor and outdoor subsets as OMNI3D$_{IN}$ (SUN RGB-D, Hypersim, ARKit), and OMNI3D$_{OUT}$ (KITTI, nuScenes). Objectron, with primarily close-up objects, is used only in the *full* OMNI3D setting.

Layout statistics. Fig. 2(a) shows the distribution of object centers onto the image plane by projecting centroids on the XY-plane (top row), and the distribution of object depths by projecting centroids onto the XZ-plane (bottom row). We find that OMNI3D's spatial distribution has a center bias, similar to 2D datasets COCO and LVIS. Fig. 2(b) depicts the relative object size distribution, defined as the square root of object area divided by image area. Objects are more likely to be small in size similar to LVIS (Fig. 6c in [24]) suggesting that OMNI3D is also challenging for 2D detection, while objects in OMNI3D$_{OUT}$ are noticeably smaller.

The bottom row of Fig. 2(a) normalizes object depth in a $[0, 20m]$ range, chosen for visualization and satisfies 88% of object instances; OMNI3D depth ranges as far as 300m. We observe that the OMNI3D depth distribution is far more diverse than SUN RGB-D and KITTI, which are biased toward near or road-side objects respectively. See Appendix for each data source distribution plot. Fig. 2(a) demonstrates OMNI3D's rich diversity in spatial distribution and depth which suffers significantly less bias than existing 3D benchmarks and is comparable in complexity to 2D datasets.

2D and 3D correlation. A common assumption in urban scenes is that objects rest on a ground plane and appear smaller with depth. To verify if that is true generally, we compute correlations. We find that 2D y and 3D z are indeed fairly correlated in OMNI3D$_{OUT}$ at 0.524, but *significantly* less in OMNI3D$_{IN}$ at 0.006. Similarly, relative 2D object size ($\sqrt{h \cdot w}$) and z correlation is 0.543 and 0.102 respectively. This confirms our claim that common assumptions in urban scenes are not generally true, making the task challenging.

Category statistics. Fig. 2(c) plots the distribution of instances across the 98 categories of OMNI3D. The long-tail suggests that low-shot recognition in both 2D and 3D will be critical for performance. In this work, we want to focus on large-scale and diverse 3D recognition, which is comparably unexplored. We therefore filter and focus on categories with at least 1000 instances. This leaves 50 diverse categories including *chair, sofa, laptop, table, books, car, truck, pedestrian* and more, 19 of which have more than 10k instances. We provide more per-category details in the Appendix.
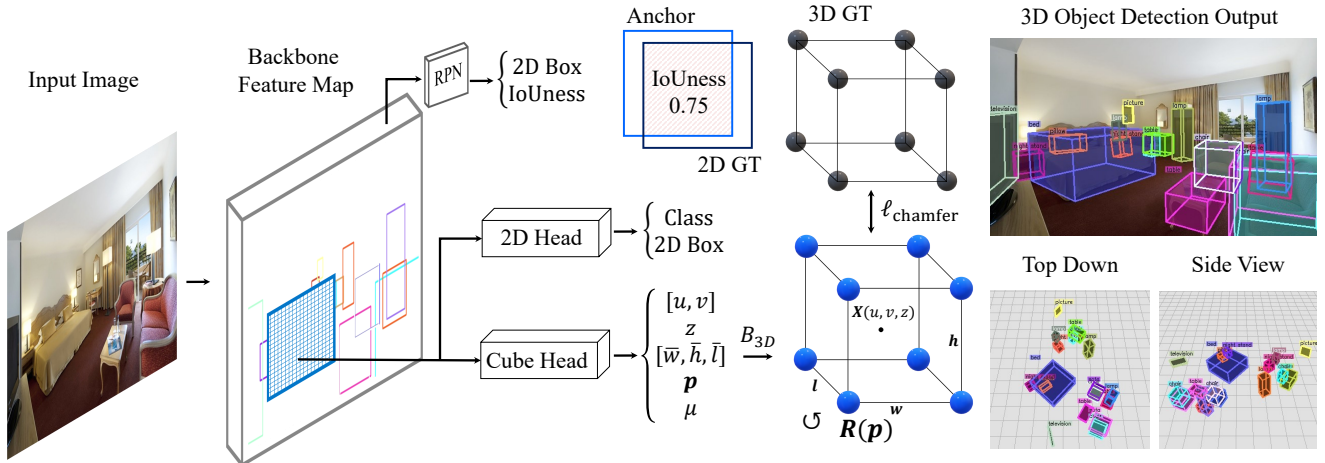
Figure 3. Overview. Cube R-CNN takes as input an RGB image, detects all objects in 2D and predicts their 3D cuboids B_{3D}. During training, the 3D corners of the cuboids are compared against 3D ground truth with the point-cloud chamfer distance.

4. Method

Our goal is to design a simple and effective model for *general* 3D object detection. Hence, our approach is free of domain or object specific strategies. We design our 3D object detection framework by extending Faster R-CNN [64] with a 3D object head which predicts a cuboid per each detected 2D object. We refer to our method as *Cube R-CNN*. Figure 3 shows an overview of our approach.

4.1. Cube R-CNN

Our model builds on Faster R-CNN [64], an end-to-end region-based object detection framework. Faster-RCNN consists of a backbone network, commonly a CNN, which embeds the input image into a higher-dimensional feature space. A region proposal network (RPN) predicts regions of interest (RoIs) representing object candidates in the image. A 2D box head inputs the backbone feature map and processes each RoI to predict a category and a more accurate 2D box. Faster R-CNN can be easily extended to tackle more tasks by adding task-specific heads, *e.g.* Mask R-CNN [26] adds a mask head to additionally output object silhouettes.

For the task of 3D object detection, we extend Faster R-CNN by introducing a 3D detection head which predicts a 3D cuboid for each detected 2D object. Cube R-CNN extends Faster R-CNN in three ways: (a) we replace the binary classifier in RPN which predicts region *objectness* with a regressor that predicts *IoUness*, (b) we introduce a *cube head* which estimates the parameters to define a 3D cuboid for each detected object (similar in concept to [68]), and (c) we define a new training objective which incorporates a *virtual depth* for the task of 3D object detection.

IoUness. The role of RPN is two-fold: (a) it proposes RoIs by regressing 2D box coordinates from pre-computed anchors and (b) it classifies regions as *object or not* (object-

ness). This is sensible in exhaustively labeled datasets where it can be reliably assessed if a region contains an object. However, OMNI3D combines many data sources with no guarantee that all instances of all classes are labeled. We overcome this by replacing objectness with *IoUness*, applied only to foreground. Similar to [34], a regressor predicts IoU between a RoI and a ground truth. Let o be the predicted IoU for a RoI and $\hat{o}$ be the 2D IoU between the region and its ground truth; we apply a binary cross-entropy (CE) loss $\mathcal{L}_{\text{IoUness}} = \ell_{\text{CE}}(o, \hat{o})$. We train on regions whose IoU exceeds 0.05 with a ground truth in order to learn IoUness from a wide range of region overlaps. Thus, the RPN training objective becomes $\mathcal{L}_{\text{RPN}} = \hat{o} \cdot (\mathcal{L}_{\text{IoUness}} + \mathcal{L}_{\text{reg}})$, where $\mathcal{L}_{\text{reg}}$ is the 2D box regression loss from [64]. The loss is weighted by $\hat{o}$ to prioritize candidates close to true objects.

Cube Head. We extend Faster R-CNN with a new head, called cube head, to predict a 3D cuboid for each detected 2D object. The cube head inputs 7×7 feature maps pooled from the backbone for each predicted region and feeds them to 2 fully-connected (FC) layers with 1024 hidden dimensions. All 3D estimations in the cube head are category-specific. The cube head represents a 3D cuboid with 13 parameters each predicted by a final FC layer:

- $[u, v]$ represent the projected 3D center on the image plane relative to the 2D RoI
- $z \in \mathbb{R}_+$ is the object's center depth in meters transformed from virtual depth z_v (explained below)
- $[\bar{w}, \bar{h}, \bar{l}] \in \mathbb{R}_+^3$ are the log-normalized physical box dimensions in meters
- $\mathbf{p} \in \mathbb{R}^6$ is the continuous 6D [86] allocentric rotation
- $\mu \in \mathbb{R}_+$ is the predicted 3D uncertainty

The above parameters form the final 3D box in camera view coordinates for each detected 2D object. The object's 3D center $\mathbf{X}$ is estimated from the predicted 2D projected

center $[u, v]$ and depth z via

$$\mathbf{X}(u, v, z) = \left(\tfrac{z}{f_x}(r_x + ur_w - p_x), \, \tfrac{z}{f_y}(r_y + vr_h - p_y), \, z \right) \tag{1}$$

where $[r_x, r_y, r_w, r_h]$ is the object's 2D box, (f_x, f_y) are the camera's known focal lengths and (p_x, p_y) the principal point. The 3D box dimensions $\mathbf{d}$ are derived from $[\bar{w}, \bar{h}, \bar{l}]$ which are log-normalized with category-specific pre-computed means (w_0, h_0, l_0) for width, height and length respectively, and are are arranged into a diagonal matrix via

$$\mathbf{d}(\bar{w}, \bar{h}, \bar{l}) = \mathrm{diag}\left(\exp(\bar{w})w_0, \, \exp(\bar{h})h_0, \, \exp(\bar{l})l_0 \right) \tag{2}$$

Finally, we derive the object's pose $\mathbf{R}(\mathbf{p})$ as a 3×3 rotation matrix based on a 6D parameterization (2 directional vectors) of $\mathbf{p}$ following [86] which is converted from allocentric to egocentric rotation similar to [39], defined formally in Appendix. The final 3D cuboid, defined by 8 corners, is

$$B_{\mathrm{3D}}(u, v, z, \bar{w}, \bar{h}, \bar{l}, \mathbf{p}) = \mathbf{R}(\mathbf{p}) \, \mathbf{d}(\bar{w}, \bar{h}, \bar{l}) \, B_{\mathrm{unit}} + \mathbf{X}(u, v, z) \tag{3}$$

where B_{unit} are the 8 corners of an axis-aligned unit cube centered at $(0, 0, 0)$. Lastly, μ denotes a learned 3D uncertainty, which is mapped to a confidence at inference then joined with the classification score s from the 2D box head to form the final score for the prediction, as $\sqrt{s \cdot \exp(-\mu)}$.

Training objective. Our training objective consists of 2D losses from the RPN and 2D box head and 3D losses from the cube head. The 3D objective compares each predicted 3D cuboid with its matched ground truth via a chamfer loss, treating the 8 box corners of the 3D boxes as point clouds, namely $\mathcal{L}_{\mathrm{3D}}^{\mathrm{all}} = \ell_{\mathrm{chamfer}}(B_{\mathrm{3D}}, B_{\mathrm{3D}}^{\mathrm{gt}})$. Note that $\mathcal{L}_{\mathrm{3D}}^{\mathrm{all}}$ *entangles* all 3D variables via the box predictor B_{3D} (Eq. 3), such that that errors in variables may be ambiguous from one another. Thus, we isolate each variable group with separate disentangled losses, following [68]. The disentangled loss for each variable group substitutes all but its variables with the ground truth from Eq. 3 to create a pseudo box prediction. For example, the disentangled loss for the projected center $[u, v]$ produces a 3D box with all but (u, v) replaced with the *true* values and then compares to the ground truth box,

$$\mathcal{L}_{\mathrm{3D}}^{(u,v)} = \| B_{\mathrm{3D}}(u, v, z^{\mathrm{gt}}, \bar{w}^{\mathrm{gt}}, \bar{h}^{\mathrm{gt}}, \bar{l}^{\mathrm{gt}}, \mathbf{p}^{\mathrm{gt}}) - B_{\mathrm{3D}}^{\mathrm{gt}} \|_1 \tag{4}$$

We use an L_1 loss for $\mathcal{L}_{\mathrm{3D}}^{(u,v)}$, $\mathcal{L}_{\mathrm{3D}}^{(z)}$ and $\mathcal{L}_{\mathrm{3D}}^{(\bar{w}, \bar{h}, \bar{l})}$ and chamfer for $\mathcal{L}_{\mathrm{3D}}^{\mathbf{P}}$ to account for cuboid symmetry such that rotation matrices of Euler angles modulo π produce the same non-canonical 3D cuboid. Losses in box coordinates have natural advantages over losses which directly compare variables to ground truths in that their gradients are appropriately weighted by the error as shown in [68]. The 3D objective is defined as, $\mathcal{L}_{\mathrm{3D}} = \mathcal{L}_{\mathrm{3D}}^{(u,v)} + \mathcal{L}_{\mathrm{3D}}^{(z)} + \mathcal{L}_{\mathrm{3D}}^{(\bar{w}, \bar{h}, \bar{l})} + \mathcal{L}_{\mathrm{3D}}^{\mathbf{P}} + \mathcal{L}_{\mathrm{3D}}^{\mathrm{all}}$.

The collective training objective of Cube R-CNN is,

$$\mathcal{L} = \mathcal{L}_{\mathrm{RPN}} + \mathcal{L}_{\mathrm{2D}} + \sqrt{2} \cdot \exp(-\mu) \cdot \mathcal{L}_{\mathrm{3D}} + \mu \tag{5}$$
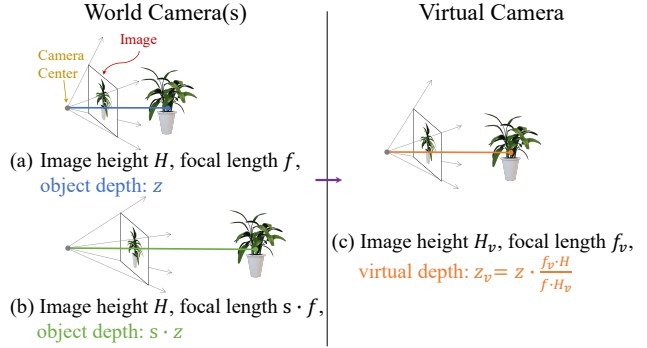

Figure 4. Varying camera intrinsics exaggerate scale-depth ambiguity – the same object at two different depths can project to the same image, as in (a) and (b). We address this by introducing a *virtual camera* which is invariant to intrinsics and transforms the object's depth to a *virtual depth* z_v such that the effective image size H_v and focal length f_v are consistent across images.

$\mathcal{L}_{\mathrm{RPN}}$ is the RPN loss, described above, and $\mathcal{L}_{\mathrm{2D}}$ is the 2D box head loss from [64]. The 3D loss is weighted by the predicted 3D uncertainty (inspired by [52]), such that the model may trade a penalty to reduce the 3D loss when uncertain. In practice, μ is helpful for both improving the 3D ratings at inference and reducing the loss of hard samples in training.

4.2. Virtual Depth

A critical part of 3D object detection is predicting an object's depth in metric units. Estimating depth from visual cues requires an implicit mapping of 2D pixels to 3D distances, which is more ambiguous if camera intrinsics vary. Prior works are able to ignore this as they primarily train on images from a single sensor. OMNI3D contains images from many sensors and thus demonstrates large variations in camera intrinsics. We design Cube R-CNN to be robust to intrinsics by predicting the object's *virtual depth* z_v, which projects the metric depth z to an invariant camera space.

Virtual depth scales the depth using the (known) camera intrinsics such that the effective image size and focal length are *consistent across the dataset*, illustrated in Fig. 4. The *effective* image properties are referred to as virtual image height H_v and virtual focal length f_v, and are both hyperparameters. If z is the true metric depth of an object from an image with height H and focal length f, its virtual depth is defined as $z_v = z \cdot \frac{f_v}{f} \frac{H}{H_v}$. The derivation is in the Appendix.

Invariance to camera intrinsics via virtual depth *also* enables scale augmentations during training, since H can vary without harming the consistency of z_v. Data augmentations from image resizing tend to be critical for 2D models but are not often used in 3D methods [7, 50, 68] since if unaccounted for they increase scale-depth ambiguity. Virtual depth lessens such ambiguities and therefore enables powerful data augmentations in training. We empirically prove the two-fold effects of *virtual depth* in Sec. 5
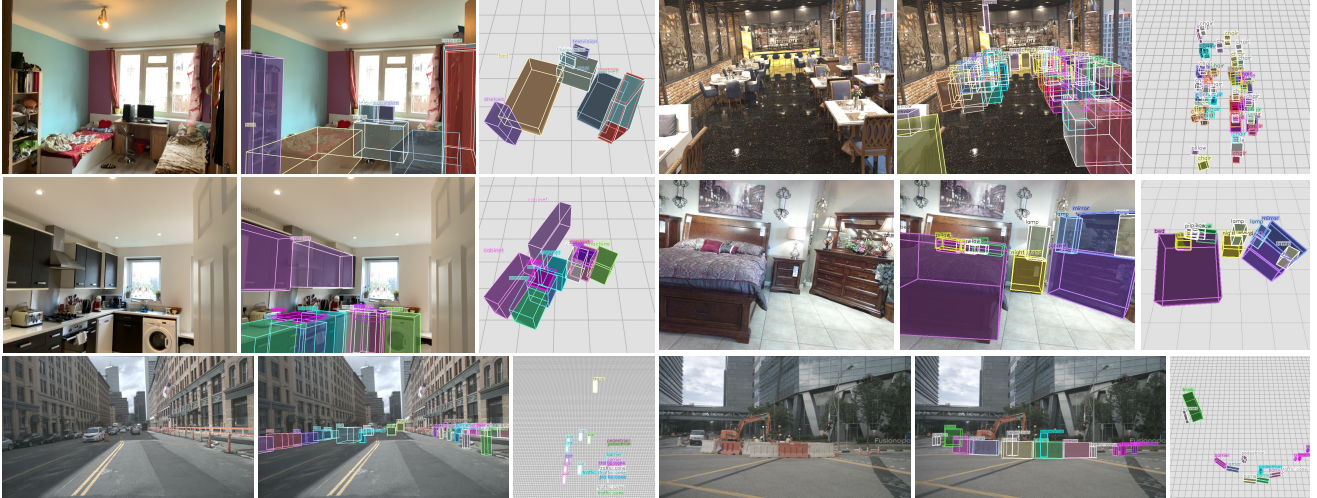
Figure 5. Cube R-CNN predictions on OMNI3D test. For each example, we show the input image, the 3D predictions overlaid on the image and a top view with a base composed of 1m×1m tiles. See more results from images and videos in the Appendix.

5. Experiments

We tackle image-based 3D object detection on OMNI3D and compare to prior best methods. We evaluate Cube R-CNN on existing 3D benchmarks with a single unified model. Finally, we prove the effectiveness of OMNI3D as a large-scale 3D object detection dataset by showing comprehensive cross-dataset generalization and its impact for pre-training.

Implementation details. We implement Cube R-CNN using Detectron2 [80] and PyTorch3D [61]. Our backbone is DLA34 [82]-FPN [43] pretrained on ImageNet [67]. We train for 128 epochs with a batch size of 192 images across 48 V100s. We use SGD with a learning rate of 0.12 which decays by a factor of 10 after 60% and 80% of training. During training, we use random data augmentation of horizontal flipping and scales $\in [0.50, 1.25]$, enabled by virtual depth. Virtual camera parameters are set to $f_v = H_v = 512$. Source code and models are publicly available.

Metric. Following popular benchmarks for 2D and 3D recognition, we use average-precision (AP) as our 3D metric. Predictions are matched to ground truth by measuring their overlap using IoU_{3D} which computes the intersection-over-union of 3D cuboids. We compute a *mean* AP_{3D} across all 50 categories in OMNI3D and over a range of IoU_{3D} thresholds $\tau \in [0.05, 0.10, \dots, 0.50]$. The range of τ is more relaxed than in 2D to account for the new dimension – see the Appendix for a comparison between 3D and 2D ranges for IoU. General 3D objects may be occluded by other objects or truncated by the image border, and can have arbitrarily small projections. Following [21], we ignore objects with high occlusion ($> 66\%$) or truncation ($> 33\%$), and tiny projected objects ($< 6.25\%$ image height). Moreover, we also report AP_{3D} at varying levels of depth d as near: $0 < d \le 10m$, medium: $10m < d \le 35m$, far: $35m < d \le \infty$.

Fast IoU3D. IoU_{3D} compares two 3D cuboids by computing their intersection-over-union. Images usually have many objects and produce several predictions so IoU_{3D} computations need to be fast. Prior implementations [21] approximate IoU_{3D} by projecting 3D boxes on a ground plane and multiply the top-view 2D intersection with the box heights to compute a 3D volume. Such approximations become notably inaccurate when objects are not on a planar ground or have an arbitrary orientation (*e.g.* with nonzero pitch or roll). Objectron [2] provides an exact solution, but relies on external libraries [5, 74] implemented in C++ and is not batched. We implement a new, fast and exact algorithm which computes the intersecting shape by representing cuboids as meshes and finds face intersections – described in Appendix. Our algorithm is batched with C++ and CUDA support. Our implementation is $90\times$ faster than Objectron in C++, and $450\times$ faster in CUDA. As a result, evaluation on the large OMNI3D takes only seconds instead of several hours.

5.1. Model Performance

First, we ablate the design choices of Cube R-CNN. Then, we compare to state-of-the-art methods on existing benchmarks and OMNI3D and show that it performs superior or on par with prior works which are specialized for their respective domains. Our Cube R-CNN uses a single unified design to tackle general 3D object detection across domains. Fig. 5 shows qualitative predictions on the OMNI3D test set.

Ablations. Table 1 ablates the features of Cube R-CNN on OMNI3D. We report AP_{3D}, performance at single IoU thresholds (0.25 and 0.50), and for near, medium and far objects. We further train and evaluate on domain-specific subsets, OMNI3D$_{IN}$ (AP_{3D}^{IN}) and OMNI3D$_{OUT}$ (AP_{3D}^{OUT}) to show how ablation trends hold for single domain scenarios. From Table 1 we draw a few standout observations:

Cube R-CNN	AP_{3D}	AP_{3D}^{25}	AP_{3D}^{50}	AP_{3D}^{near}	AP_{3D}^{med}	AP_{3D}^{far}	AP_{3D}^{IN}	AP_{3D}^{OUT}
w/o disentangled	22.6	24.3	9.2	26.8	11.8	8.2	14.5	30.9
w/o IoUness	22.2	23.6	8.8	26.4	11.1	8.3	14.3	31.0
w/o $\mathcal{L}_{3D}^{all}$	20.2	21.8	6.4	26.4	**12.1**	7.4	13.8	29.1
w/o scale aug.	20.2	21.5	8.0	23.5	9.8	6.8	12.3	28.1
w/o scale | virtual	19.8	21.2	7.5	23.4	8.6	5.7	12.2	26.0
w/o virtual depth	17.3	18.4	4.4	22.7	7.9	7.1	13.2	30.3
w/o uncertainty μ	17.2	18.3	5.4	20.5	10.8	7.0	9.1	25.4
ours	**23.3**	**24.9**	**9.5**	**27.9**	**12.1**	**8.5**	**15.0**	**31.9**

Table 1. Cube R-CNN ablations. We report AP_{3D} on OMNI3D, then at IoU thresholds 0.25 and 0.50, and for near, medium and far objects. We further report AP_{3D} when both training and evaluating on OMNI3D$_{IN}$ and OMNI3D$_{OUT}$ subsets.

Method	f	OMNI3D$_{OUT}$ AP_{3D}^{KIT}	AP_{3D}^{NU}	AP_{3D}^{OUT}	OMNI3D AP_{3D}
M3D-RPN [7]	✗	10.4	17.9	13.7	-
SMOKE [50]	✗	25.4	20.4	19.5	9.6
FCOS3D [78]	✗	14.6	20.9	17.6	9.8
PGD [77]	✗	21.4	26.3	22.9	11.2
GUPNet [52]	✓	24.5	20.5	19.9	-
M3D-RPN +*vc*	✓	16.2$_{+5.8}$	20.4$_{+2.9}$	17.0$_{+3.3}$	-
ImVoxelNet [66]	✓	23.5	23.4	21.5	9.4
SMOKE +*vc*	✓	25.9$_{+0.5}$	20.4$_{+0.0}$	20.0$_{+0.5}$	10.4$_{+0.8}$
FCOS3D +*vc*	✓	17.8$_{+3.2}$	25.1$_{+4.2}$	21.3$_{+3.7}$	10.6$_{+0.8}$
PGD +*vc*	✓	22.8$_{+1.4}$	31.2$_{+4.9}$	26.8$_{+3.9}$	15.4$_{+4.2}$
Cube R-CNN	✓	**36.0**	**32.7**	**31.9**	**23.3**

Table 2. OMNI3D comparison with GUPNet [52], ImVoxelNet [66], M3D-RPN [7], FCOS3D [78], PGD [77], SMOKE [50]. We apply virtual camera to the last four (+*vc*) and show the gains in purple. f denotes if a method handles variance in focal length.

Method	D	AP_{3D}^{70} Easy	Med	Hard	AP_{BEV}^{70} Easy	Med	Hard
SMOKE [50]	✗	14.03	9.76	7.84	20.83	14.49	12.75
ImVoxelNet [66]	✗	17.15	10.97	9.15	25.19	16.37	13.58
PGD [77]	✗	19.05	11.76	9.39	26.89	16.51	13.49
GUPNet [52]	✗	22.26	**15.02**	**13.12**	30.29	21.19	18.20
Cube R-CNN	✗	**23.59**	15.01	12.56	**31.70**	**21.20**	**18.43**
MonoDTR [30]	✓	21.99	15.39	12.73	28.59	20.38	17.14
DD3D [59]	✓	23.19	16.87	14.36	32.35	23.41	20.42

Table 3. KITTI leaderboard results on *car* for image-based approaches. D denotes methods trained with extra depth supervision.

Virtual depth is effective and improves AP_{3D} by +6%, most noticable in the full OMNI3D which has the largest variances in camera intrinsics. Enabled by virtual depth, scale augmentations during training increase AP_{3D} by +3.1% on OMNI3D, +2.7% on OMNI3D$_{IN}$ and +3.8% on OMNI3D$_{OUT}$. We find scale augmentation controlled *without* virtual depth to be harmful by −2.5% (rows 6 - 7).

3D uncertainty boosts performance by about +6% on all OMNI3D subsets. Intuitively, it serves as a measure of confidence for the model's 3D predictions; removing it means that the model relies only on its 2D classification to score cuboids. If uncertainty is used only to scale samples in Eq. 5, but not at inference then AP_{3D} still improves but by +1.6%.

Table 1 also shows that the entangled $\mathcal{L}_{3D}^{all}$ loss from Eq. 5 boosts AP_{3D} by +3.1% while disentangled losses contribute less (+0.7%). Replacing *objectness* with *IoUness* in the RPN head improves AP_{3D} on OMNI3D by +1.1%.

Comparison to other methods. We compare Cube R-CNN to prior best approaches. We choose representative state-of-the-art methods designed for the urban and indoor domains, and evaluate on our proposed OMNI3D benchmark and single-dataset benchmarks, KITTI and SUN RGB-D.

Comparisons on OMNI3D. Table 2 compares Cube R-CNN to M3D-RPN [7], a single-shot 3D anchor approach, FCOS3D [78] and its follow-up PGD [77], which use a fully convolutional one-stage model, GUPNet [52], which uses

2D-3D geometry and camera focal length to derive depth, SMOKE [50] which predicts 3D object center and offsets densely and ImVoxelNet [66], which uses intrinsics to un-project 2D image features to a 3D volume followed by 3D convolutions. These methods originally experiment on urban domains, so we compare on OMNI3D$_{OUT}$ and the full OMNI3D. We train each method using public code and report the best of multiple runs after tuning their hyper-parameters to ensure best performance. We extend all but GUPNet and ImVoxelNet with a *virtual camera* to handle varying intrinsics, denoted by +*vc*. GUPNet and ImVoxelNet dynamically use per-image intrinsics, which can naturally account for camera variation. M3D-RPN and GUPNet do not support multi-node training, which makes scaling to OMNI3D difficult and are omitted. For OMNI3D$_{OUT}$, we report AP_{3D}^{OUT} on its test set and its subparts, AP_{3D}^{KIT} and AP_{3D}^{NU}.

From Table 2 we observe that our Cube R-CNN outperforms competitive methods on both OMNI3D$_{OUT}$ and OMNI3D. The *virtual camera* extensions result in performance gains of varying degree for all approaches, *e.g.* +4.2% for PGD on OMNI3D, proving its impact as a general purpose feature for mixed dataset training for the task of 3D object detection. FCOS3D, SMOKE, and ImVoxelNet [66] struggle the most on OMNI3D, which is not surprising as they are typically tailored for dataset-specific model choices.

Comparisons on KITTI. Table 3 shows results on KITTI's test set using their server [1]. We train Cube R-CNN on KITTI only. Cube R-CNN is a general purpose 3D object detector; its design is not tuned for the KITTI benchmark. Yet, it performs better or on par with recent best methods which are heavily tailored for KITTI and is only slightly worse to models trained with extra depth supervision.

Method	Trained on	IoU_{3D}	AP_{3D}
Total3D [58]	SUN RGB-D	23.3	-
ImVoxelNet [66]	SUN RGB-D	-	30.6
Cube R-CNN	SUN RGB-D	36.2	34.7
Cube R-CNN	$\text{OMNI3D}_{\text{IN}}$	**37.8**	**35.4**

Table 4. We compare to indoor models with common categories. For Total3D, we use oracle 2D detections to report IoU_{3D} (3$^{\text{rd}}$ col). For ImVoxelNet, we report the detection metric of AP_{3D} (4$^{\text{th}}$ col).

KITTI's AP_{3D} is fundamentally similar to ours but sets a single IoU_{3D} threshold at 0.70, which behaves similar to a nearly perfect 2D threshold of 0.94 (see the Appendix). To remedy this, our AP_{3D} is a mean over many thresholds, inspired by COCO [45]. In contrast, nuScenes proposes a mean AP based on 3D center distance but omits size and rotation, with settings generally tailored to the urban domain (*e.g.* size and rotation variations are less extreme for cars). We provide more analysis with this metric in the Appendix.

Comparisons on SUN RGB-D. Table 4 compares Cube R-CNN to Total3D [58] and ImVoxelNet [66], two state-of-the-art methods on SUN RGB-D on 10 common categories. Total3D's public model *requires* 2D object boxes and categories as input. Hence, for a fair comparison, we use ground truth 2D detections as input to both our and their method, each trained using a ResNet34 [27] backbone and report mean IoU_{3D} (3$^{\text{rd}}$ col). We compare to ImVoxelNet in the full 3D object detection setting and report AP_{3D} (4$^{\text{th}}$ col). We compare Cube R-CNN from two respects, first trained on SUN RGB-D identical to the baselines, then trained on $\text{OMNI3D}_{\text{IN}}$ which subsumes SUN RGB-D. Our model outperforms Total3D by $+12.9\%$ and ImVoxelNet by $+4.1\%$ when trained on the same training set and increases the performance gap when trained on the larger $\text{OMNI3D}_{\text{IN}}$.

Zero-shot Performance. One commonly sought after property of detectors is zero-shot generalization to other datasets. Cube R-CNN outperforms all other methods at zero-shot. We find our model trained on KITTI achieves 12.7% AP_{3D} on nuScenes; the second best is M3D-RPN+*vc* with 10.7%. Conversely, when training on nuScenes, our model achieves 20.2% on KITTI; the second best is GUPNet with 17.3%.

5.2. The Impact of the OMNI3D Benchmark

Sec. 5.1 analyzes the performance of Cube R-CNN for the task of 3D object detection. Now, we turn to OMNI3D and its impact as a large-scale benchmark. We show two use cases of OMNI3D: (a) a universal 3D dataset which integrates smaller ones, and (b) a pre-training dataset.

OMNI3D as a universal dataset. We treat OMNI3D as a dataset which integrates smaller single ones and show its impact on each one. We train Cube R-CNN on OMNI3D and compare to single-dataset training in Table 5, for the indoor (left) and urban (right) domain. AP_{3D} is reported on the category intersection (10 for indoor, 3 for outdoor) to ensure comparability. Training on OMNI3D and its domain-specific

Trained on	$\text{AP}^{\text{HYP}}_{3D}$	$\text{AP}^{\text{SUN}}_{3D}$	$\text{AP}^{\text{AR}}_{3D}$	Trained on	$\text{AP}^{\text{KIT}}_{3D}$	$\text{AP}^{\text{NU}}_{3D}$
Hypersim	15.2	9.5	7.5	KITTI	37.1	12.7
SUN	5.8	34.7	13.1	nuScenes	20.2	38.6
ARKit	5.9	14.2	38.6	OMNI3D	37.8	35.8
OMNI3D	**19.0**	32.6	38.2	$\text{OMNI3D}_{\text{OUT}}$	**42.4**	**39.0**
$\text{OMNI3D}_{\text{IN}}$	17.8	**35.4**	**41.2**			

Table 5. Cross-dataset performance on *intersecting* categories (10 for indoor, 3 for outdoor) for comparable cross-evaluations.

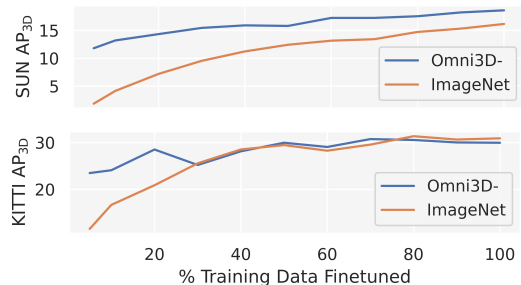Figure 6. Pre-training on OMNI3D- vs. ImageNet.

subsets, $\text{OMNI3D}_{\text{IN}}$ and $\text{OMNI3D}_{\text{OUT}}$, results in higher performance compared to single-dataset training, signifying that our large OMNI3D generalizes better and should be preferred over single dataset training. ARKit sees a $+2.6\%$ boost and KITTI $+5.3\%$. Except for Hypersim, the domain-specific subsets tend to perform better on their domain, which is not surprising given their distinct properties (Fig. 2).

OMNI3D as a pre-training dataset. Next, we demonstrate the utility of OMNI3D for pre-training. In this setting, an unseen dataset is used for finetuning from a model pre-trained on OMNI3D. The motivation is to determine how a large-scale 3D dataset could accelerate low-shot learning with minimum need for costly 3D annotations on a new dataset. We choose SUN RGB-D and KITTI as our *unseen* datasets given their popularity and small size. We pre-train Cube R-CNN on OMNI3D-, which removes them from OMNI3D, then finetune the models using a % of their training data. The curves in Fig. 6 show the model quickly gains its upper-bound performance at a small fraction of the training data, when pre-trained on OMNI3D- vs. ImageNet, without any few-shot training tricks. A model finetuned on only 5% of its target can achieve >70% of the upper-bound performance.

6. Conclusion

We propose a large and diverse 3D object detection benchmark, OMNI3D, and a general purpose 3D object detector, Cube R-CNN. Models and data are publicly released. Our extensive analysis (Table 1, Fig. 5, Appendix) show the strength of general 3D object detection as well as the limitations of our method, *e.g.* localizing far away objects, and uncommon object types or contexts. Moreover, for accurate real-world 3D predictions, our method is limited by the assumption of known camera intrinsics, which may be minimized in future work using self-calibration techniques.

References

[1] KITTI server. https://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d. Accessed: 2022-11-01. 7

[2] Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. *CVPR*, 2021. 2, 3, 6

[3] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2CAD: Learning CAD model alignment in RGB-D scans. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 2614–2623, 2019. 2

[4] Wentao Bao, Bin Xu, and Zhenzhong Chen. MonoFENet: Monocular 3D object detection with feature enhancement networks. *TIP*, 2019. 2

[5] Bradford Barber, David Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 1996. 6

[6] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitScenes - a diverse real-world dataset for 3D indoor scene understanding using mobile RGB-D data. In *NeurIPS Datasets and Benchmarks Track*, 2021. 2, 3

[7] Garrick Brazil and Xiaoming Liu. M3D-RPN: Monocular 3D region proposal network for object detection. In *ICCV*, 2019. 1, 2, 5, 7

[8] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele. Kinematic 3D object detection in monocular video. In *ECCV*, 2020. 2

[9] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 2, 3

[10] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3D tracking and forecasting with rich maps. In *CVPR*, 2019. 2

[11] Hansheng Chen, Yuyao Huang, Wei Tian, Zhong Gao, and Lu Xiong. MonoRUn: Monocular 3D object detection by reconstruction and uncertainty propagation. In *CVPR*, 2021. 2

[12] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 2

[13] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3D object detection for autonomous driving. In *CVPR*, 2016. 1, 2

[14] Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li. MonoPair: Monocular 3D object detection using pairwise spatial relationships. In *CVPR*, 2020. 2

[15] Yi-Nan Chen, Hang Dai, and Yong Ding. Pseudo-stereo for monocular 3D object detection in autonomous driving. In *CVPR*, 2022. 2

[16] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, 2017. 2

[17] Saumitro Dasgupta, Kuan Fang, Kevin Chen, and Silvio Savarese. Delay: Robust spatial layout estimation for cluttered indoor scenes. In *CVPR*, 2016. 2

[18] Mingyu Ding, Yuqi Huo, Hongwei Yi, Zhe Wang, Jianping Shi, Zhiwu Lu, and Ping Luo. Learning depth-guided convolutions for monocular 3D object detection. In *CVPRW*, 2020. 2

[19] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep Ordinal Regression Network for Monocular Depth Estimation. In *CVPR*, 2018. 2

[20] Nils Gählert, Nicolas Jourdan, Marius Cordts, Uwe Franke, and Joachim Denzler. Cityscapes 3D: Dataset and benchmark for 9 DOF vehicle detection. In *CVPRW*, 2020. 2

[21] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012. 1, 2, 3, 6

[22] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh R-CNN. In *ICCV*, 2019. 2

[23] Jiaqi Gu, Bojian Wu, Lubin Fan, Jianqiang Huang, Shen Cao, Zhiyu Xiang, and Xian-Sheng Hua. Homography loss for monocular 3D object detection. In *CVPR*, 2022. 2

[24] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 1, 3

[25] Chenhang He, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Aug3D-RPN: Improving monocular 3D object detection by synthetic images with virtual depth. *arXiv preprint arXiv:2107.13269*, 2021. 2

[26] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 1, 2, 4

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 8

[28] Varsha Hedau, Derek Hoiem, and David Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, 2009. 2

[29] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Long Chen, Ashesh Jain, Sammy Omari, Vladimir Iglovikov, and Peter Ondruska. One thousand and one hours: Self-driving motion prediction dataset. *arXiv preprint arXiv:2006.14480*, 2020. 2

[30] Kuan-Chih Huang, Tsung-Han Wu, Hung-Ting Su, and Winston Hsu. MonoDTR: Monocular 3D object detection with depth-aware transformer. In *CVPR*, 2022. 2, 7

[31] Siyuan Huang, Siyuan Qi, Yinxue Xiao, Yixin Zhu, Ying Nian Wu, and Song-Chun Zhu. Cooperative holistic scene understanding: Unifying 3D object, layout, and camera pose estimation. In *NeurIPS*, 2018. 1, 2

[32] Xinyu Huang, Peng Wang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *TPAMI*, 2019. 2

[33] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet. Level 5 perception dataset 2020. https://level-5.global/level5/data/, 2019. 2

[34] Dahun Kim, Tsung-Yi Lin, Anelia Angelova, In Kweon, and Weicheng Kuo. Learning open-world object proposals without learning to classify. *RAL*, 2022. 4

[35] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven Waslander. Joint 3D proposal generation and object detection from view aggregation. In *IROS*, 2018. 2

[36] Nilesh Kulkarni, Ishan Misra, Shubham Tulsiani, and Abhinav Gupta. 3D-RelNet: Joint object and relational network for 3D prediction. In *ICCV*, 2019. 1, 2

[37] Abhinav Kumar, Garrick Brazil, Enrique Corona, Armin Parchami, and Xiaoming Liu. DEVIANT: Depth EquiVarIAnt NeTwork for monocular 3D object detection. In *ECCV*, 2022. 2

[38] Abhinav Kumar, Garrick Brazil, and Xiaoming Liu. GrooMeD-NMS: Grouped mathematically differentiable NMS for monocular 3D object detection. In *CVPR*, 2021. 2

[39] Abhijit Kundu, Yin Li, and James Rehg. 3D-RCNN: Instance-level 3D object reconstruction via render-and-compare. In *CVPR*, 2018. 5

[40] Hei Law and Jia Deng. CornerNet: Detecting objects as paired keypoints. In *ECCV*, 2018. 1

[41] David C Lee, Martial Hebert, and Takeo Kanade. Geometric reasoning for single image structure recovery. In *CVPR*, 2009. 2

[42] Peixuan Li, Huaici Zhao, Pengfei Liu, and Feidao Cao. RTM3D: Real-time monocular 3D detection from object keypoints for autonomous driving. In *ECCV*, 2020. 2

[43] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 6

[44] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 2

[45] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1, 3, 8

[46] Ce Liu, Shuhang Gu, Luc Van Gool, and Radu Timofte. Deep line encoding for monocular 3D object detection and depth prediction. In *BMVC*, 2021. 2

[47] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander Berg. SSD: Single shot multibox detector. In *ECCV*, 2016. 2

[48] Xianpeng Liu, Nan Xue, and Tianfu Wu. Learning auxiliary monocular contexts helps monocular 3D object detection. In *AAAI*, 2022. 2

[49] Yuxuan Liu, Yuan Yixuan, and Ming Liu. Ground-aware monocular 3D object detection for autonomous driving. *RAL*, 2021. 2

[50] Zechen Liu, Zizhang Wu, and Roland Tóth. SMOKE: Single-stage monocular 3D object detection via keypoint estimation. In *CVPRW*, 2020. 1, 2, 5, 7

[51] Zongdai Liu, Dingfu Zhou, Feixiang Lu, Jin Fang, and Liangjun Zhang. Autoshape: Real-time shape-aware monocular 3D object detection. In *ICCV*, 2021. 2

[52] Yan Lu, Xinzhu Ma, Lei Yang, Tianzhu Zhang, Yating Liu, Qi Chu, Junjie Yan, and Wanli Ouyang. Geometry uncertainty projection network for monocular 3D object detection. In *ICCV*, 2021. 1, 2, 5, 7

[53] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking pseudo-lidar representation. In *ECCV*, 2020. 2

[54] Xinzhu Ma, Zhihui Wang, Haojie Li, Pengbo Zhang, Wanli Ouyang, and Xin Fan. Accurate monocular 3D object detection via color-embedded 3D reconstruction for autonomous driving. In *ICCV*, 2019. 2

[55] Xinzhu Ma, Yinmin Zhang, Dan Xu, Dongzhan Zhou, Shuai Yi, Haojie Li, and Wanli Ouyang. Delving into localization errors for monocular 3D object detection. In *CVPR*, 2021. 2

[56] Arun Mallya and Svetlana Lazebnik. Learning informative edge maps for indoor scene layout prediction. In *ICCV*, 2015. 2

[57] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3D bounding box estimation using deep learning and geometry. In *CVPR*, 2017. 1, 2

[58] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Zhang. Total3DUnderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *CVPR*, 2020. 1, 2, 8

[59] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3D object detection? In *ICCV*, 2021. 2, 7

[60] Charles Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas Guibas. Frustum pointnets for 3D object detection from RGB-D data. In *CVPR*, 2018. 2

[61] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3D deep learning with PyTorch3D. *arXiv:2007.08501*, 2020. 6

[62] Cody Reading, Ali Harakeh, Julia Chae, and Steven Waslander. Categorical depth distribution network for monocular 3D object detection. In *CVPR*, 2021. 2

[63] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 1, 2

[64] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1, 2, 4, 5

[65] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Bautista, Nathan Paczan, Russ Webb, and Joshua Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *ICCV*, 2021. 2, 3

[66] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. ImVoxelNet: Image to voxels projection for monocular and multi-view general-purpose 3D object detection. In *WACV*, 2022. 2, 7, 8

[67] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander Berg, and Li

Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 6

[68] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3D object detection. In *ICCV*, 2019. 2, 4, 5

[69] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Elisa Ricci, and Peter Kontschieder. Towards generalization across depth for monocular 3D object detection. In *ECCV*, 2020. 2

[70] Shuran Song, Samuel Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *CVPR*, 2015. 1, 2, 3

[71] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 2

[72] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *ICCV*, 2019. 1, 2

[73] Shubham Tulsiani, Saurabh Gupta, David Fouhey, Alexei Efros, and Jitendra Malik. Factoring shape, pose, and layout from the 2D image of a 3D scene. In *CVPR*, 2018. 1, 2

[74] Pauli Virtanen, Ralf Gommers, Travis Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 2020. 6

[75] Li Wang, Liang Du, Xiaoqing Ye, Yanwei Fu, Guodong Guo, Xiangyang Xue, Jianfeng Feng, and Li Zhang. Depth-conditioned dynamic message propagation for monocular 3D object detection. In *CVPR*, 2021. 2

[76] Li Wang, Li Zhang, Yi Zhu, Zhi Zhang, Tong He, Mu Li, and Xiangyang Xue. Progressive coordinate transforms for monocular 3D object detection. In *NeurIPS*, 2021. 2

[77] Tai Wang, Zhu Xinge, Jiangmiao Pang, and Dahua Lin. Probabilistic and geometric depth: Detecting objects in perspective. In *CoRL*, 2022. 2, 7

[78] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. FCOS3D: Fully convolutional one-stage monocular 3D object detection. In *ICCVW*, 2021. 2, 7

[79] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving. In *CVPR*, 2019. 2

[80] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 2, 6

[81] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Weinberger. Pseudo-lidar++: Accurate depth for 3D object detection in autonomous driving. In *ICLR*, 2020. 2

[82] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *CVPR*, 2018. 6

[83] Yunpeng Zhang, Jiwen Lu, and Jie Zhou. Objects are different: Flexible monocular 3D object detection. In *CVPR*, 2021. 2

[84] Dingfu Zhou, Xibin Song, Yuchao Dai, Junbo Yin, Feixiang Lu, Miao Liao, Jin Fang, and Liangjun Zhang. IAFA: Instance-aware feature aggregation for 3D object detection from a single image. In *ACCV*, 2020. 2

[85] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019. 2

[86] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019. 4, 5

[87] Yunsong Zhou, Yuan He, Hongzi Zhu, Cheng Wang, Hongyang Li, and Qinhong Jiang. Monocular 3D object detection: An extrinsic parameter free approach. In *CVPR*, 2021. 2

[88] Zhikang Zou, Xiaoqing Ye, Liang Du, Xianhui Cheng, Xiao Tan, Li Zhang, Jianfeng Feng, Xiangyang Xue, and Errui Ding. The devil is in the task: Exploiting reciprocal appearance-localization features for monocular 3D object detection. In *ICCV*, 2021. 2