Boxer: Robust Lifting of Open-World 2D Bounding Boxes to 3D

Daniel DeTone, Tianwei Shen, Fan Zhang, Lingni Ma, Julian Straub, Richard Newcombe, and Jakob Engel

Meta Reality Labs Research, USA
ddetone@meta.com

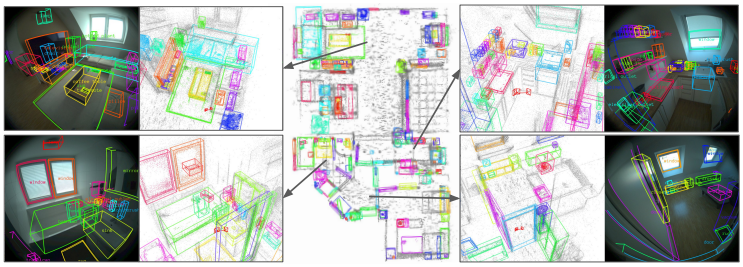

Fig. 1: Boxer takes posed images with optional depth and off-the-shelf open-world 2DBB detections as input, estimating static, global 3D bounding boxes. We run Boxer on a sequence and highlight various scenes to show the accuracy and open-world coverage of objects such as a spice jar, hairdryer, sink drain, and TV remote.

Abstract. Detecting and localizing objects in space is a fundamental computer vision problem. While much progress has been made to solve 2D object detection, 3D object localization is much less explored and far from solved, especially for open-world categories. To address this research challenge, we propose Boxer, an algorithm to estimate static 3D bounding boxes (3DBBs) from 2D open-vocabulary object detections, posed images, and optional depth represented either as a sparse point cloud or dense depth. At its core is BoxerNet, a transformer-based network that lifts 2D bounding box (2DBB) proposals into 3D, followed by multi-view fusion and geometric filtering to produce globally consistent de-duplicated 3DBBs in metric world space. Boxer leverages the power of existing 2DBB detection algorithms (*e.g.,* DETIC [52], OWLv2 [29], SAM3 [8]) to localize objects in 2D. This allows the main BoxerNet model to focus on lifting to 3D rather than detecting, ultimately reducing the demand for costly annotated 3DBB training data. Extending the CuTR [20] formulation, we incorporate an aleatoric uncertainty for robust regression, a mean depth patch encoding to support sparse depth inputs, and large-scale training with over 1.2 million unique 3DBBs. BoxerNet outperforms state-of-the-art baselines in open-world 3DBB lifting, including CuTR in egocentric settings without dense depth (0.532 vs. 0.010 mAP) and on CA-1M with dense depth available (0.412 vs. 0.250 mAP). Project page and code are available at https://facebookresearch.github.io/boxer.

1 Introduction

The transition from passive observation to active interaction in contextual AI and embodiment requires engaging the metric 3D world. Estimating 3D bounding boxes (3DBBs) for arbitrary objects is central to this process, providing the semantic and geometric foundation for spatial understanding. An agent must reason about object identity, scale, orientation, and proximity to support goal-directed autonomous navigation and dexterous manipulation. The same capability is critical for immersive augmented reality (AR) and digital twins, where AI must align semantic knowledge with physical space for seamless human-machine collaboration.

Despite being a cornerstone of contextual AI and embodiment, 3DBB estimation remains fraught with unresolved challenges. The primary obstacle is a profound data-annotation disparity. While 2D detection scales through intuitive image-space labeling [18, 23], metric 3DBB estimation from a monocular view is inherently ill-posed. Collecting and annotating 3DBBs therefore requires specialized setups, typically involving LiDAR, depth sensors, or calibrated multi-view systems for 3D grounding. This creates a bottleneck, leaving even the largest 3D datasets orders of magnitude smaller than their 2D counterparts [6, 20]. As a result, 3D models are often starved of the "web-scale" diversity that has enabled 2D vision-language models to learn the long tail of open-world objects. Furthermore, the current 3DBB data landscape is fragmented across sensors and modalities. Available datasets form a patchwork: some provide dense depth maps from RGB-D cameras, while others provide sparse point clouds from simultaneous localization and mapping (SLAM) [13] or structure-from-motion (SfM) [40]. Data are also captured with diverse camera models, ranging from pinhole to fisheye and panoramic lenses.

The data and annotation landscape leaves the field under-explored and technologically rigid. Some methods remain trapped in a "closed-set" taxonomy, with no clear path to the open-world long tail [48, 50], as they rely heavily on closed-set 3D supervision [6]. Other approaches rely heavily on 3D supervision for both detection and localization, placing high demands on training data and failing to leverage mature 2D detection as a bootstrap [20, 42]. Moreover, current 3D estimation paradigms struggle to handle this heterogeneity because camera models and sensor modalities are often baked into the architecture, requiring substantial re-engineering or retraining to transfer across settings. In summary, there is still no universal interface for 3D perception that bridges powerful 2D semantics and metric 3D geometry. Such a system should adapt its predictions to available geometric cues, whether from dense geometry or sparse points, while remaining agnostic to the underlying camera model.

Motivated by this gap, we bridge 2D and 3D detection by decomposing the problem into two stages: 2D localization followed by metric 3D lifting. By using existing open-world 2D detectors trained on internet-scale data, we inherit broad semantic coverage and strong localization in the image plane. We then focus model capacity on the geometric lifting step, learning to predict metric 3D bounding boxes from 2D proposals and depth cues. As illustrated in Fig. 1,

this decomposition allows us to benefit from the scale of 2D supervision while addressing the geometric requirements of 3D detection. Additionally, we use flexible input conditioning to support training on pinhole and fisheye camera models and on datasets with and without sparse point clouds or dense depth maps. The contributions of this work are summarized as follows.

- We propose **Boxer**, a complete algorithm for estimating open-world, global, de-duplicated 3D bounding boxes for objects using posed, calibrated video with optional sparse point clouds or dense depth.
- We introduce **BoxerNet**, a transformer-based model that lifts 2D bounding box detections from off-the-shelf open-world detectors into metric 3D bounding boxes. It improves upon the CuTR [20] architecture by incorporating an **aleatoric uncertainty** head and a **mean depth patch encoding** which enables **large-scale training** on over 1.22 million unique 3DBBs across four different device types.
- BoxerNet **outperforms state-of-the-art baselines** in the task of open-world 2D to 3D bounding box lifting across different camera types, including outperforming CuTR in egocentric settings without dense depth (0.532 vs. 0.010 mAP) and on CA-1M with dense depth available (0.412 vs. 0.250 mAP).
- We release code and model to enable inference on a variety of input data.

2 Related Work

We review closed- and open-set 3D object detection methods, as well as open-set 2D bounding-box detection, since Boxer builds on these.

Closed-world 3D detection. Many approaches for 3DBB detection focus on a fixed set of classes (e.g., chair, table, lamp), such as Cube-RCNN [6], which uses a Faster-R-CNN-style 2D detection approach trained on a large annotated dataset called Omni3D to detect objects from monocular images. Other works such as H3DNet [51], FCAF [37], TR3D [39], VoteNet [34] and 3DETR [30] rely primarily on sparse point-cloud input from a depth scan to detect a fixed set of classes. EVL [42] operates on a multi-camera stereo rig such as Project Aria [13,17] and follows a similar detection strategy as ImVoxelNet [38] that lifts 2D features into a 3D voxel grid. More recent Transformer-based multi-view methods like DETR3D [45], PARQ [47], or PETR [24] directly query 3D objects from calibrated image sets. SceneScript [3] formulates 3DBB estimation in an autoregressive way using a custom object language. SpatialLM [28] is another example of a 3D-LLM capable of detecting objects in 3D, but detection is still limited to a closed set of 59 common categories and trained only on synthetic data. While the aforementioned works focus on indoor scenes, 3D detection for autonomous driving typically adopts Bird's-Eye-View (BEV) approaches [21,22,32] to localize fixed-set classes like cars and pedestrians.

Open-world 2D detection. Open-world 2D object detection is a well-studied and extensive field. Here we focus only on the most relevant approaches that take an input text prompt such as "red car" and localize it in the image with a 2D bounding box and, optionally, a segmentation mask. DETIC [52] is an example capable of detecting 21k+ categories using weak image-level annotations. Powerful models such as OWLv2 [29] can detect tens of thousands of classes and rely on pseudo-annotations to self-label over 1 billion weakly annotated web image-text pairs. SAM3 [8] is another approach that uses a large and diverse corpus of 4 million object-text concepts and includes segmentation masks. Vision–language models (VLMs) like Qwen3-VL [4] do not mention an exact number of 2D bounding box annotations but are pretrained on web-scale image–text corpora involving hundreds of billions of tokens, which supports robust multimodal grounding and open-vocabulary reasoning.

Open-world 3D detection. Open-world 3DBB detection is a less mature field. A promising shift in open-set 3D detection treats localization as a generative "next-token prediction" task [10], leveraging the broad semantic knowledge and reasoning capabilities of VLMs. Building on this, LocateAnything3D [27] introduced a Chain-of-Sight (CoS) decoding strategy that mirrors human reasoning by emitting 2D grounding tokens as a visual anchor before predicting 3D coordinates. While large-scale foundation models like Qwen3-VL [4] and N3D-VLM [46] have recently incorporated 3D grounding into their unified multimodal output via JSON-like tokens, their performance is often hampered by 3D hallucinations and a lack of metric precision compared to dedicated geometric systems.

Similar to Boxer, 3D-MOOD [48] and DetAny3D [50] also lift open-world 2D detections to 3D conditioned on monocular images. One limitation of these approaches is that they cannot condition on commonly available known depth from a depth sensor or sparse point clouds because they compute dense depth internally. OVMono3D [49] proposes a framework for open-vocabulary monocular 3D detection that decouples 2D semantic recognition from 3D spatial estimation, leveraging pretrained vision foundation models to lift 2D bounding boxes into metric 3D space for novel categories. These approaches focus on monocular detection, whereas Boxer supports optional metric-scale depth from a calibrated stereo rig or RGB-D sensor. The closest to Boxer is CuTR [20], which trains a DETR-style 3DBB detector using open-world 3DBB-annotated data from ARKit Scenes [5]. In Boxer, we follow a similar architecture to CuTR, but we replace the pixel-level dense-depth ViT with a more flexible depth-patch encoder to enable scaling to different depth input types. Since CuTR is built on the DETR framework, a confidence score is available via the foreground/background classifier score. However, this couples both the 2D and 3D detection scores. In our work we factorize this into a separate 3D uncertainty. SAM3D [9] goes a step beyond 3DBB detection (which it describes as layout estimation) and estimates a full 3D shape plus texture of each object. Combined with an open-world detection and segmentation model such as SAM3 [8], it can serve a similar function to Boxer, while lacking the ability to process depth input and fuse estimates from multi-view video.

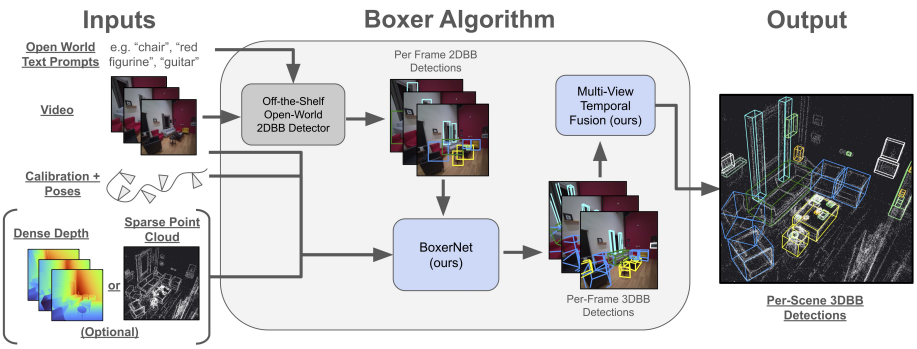

Fig. 2: Boxer algorithm overview. Boxer operates on a set of posed and calibrated images with optional dense depth or sparse point clouds to produce metric, static 3D bounding boxes for open-set objects.

To bridge the gap between 2D reasoning and 3D geometric consistency, recent works such as ConceptGraphs [14] and EgoLifter [15] lift open-world 2D segmentation masks into 3D, constructing object-centric scene representations by associating class-agnostic segments across multi-view sequences. In contrast, Boxer functions as a flexible detector designed for efficient 3DBB estimation, and uses a post-processing temporal fusion step to achieve 3D consistency.

3 Boxer

Boxer is an algorithm that takes as input a set of posed, calibrated images and produces a final set of global, metric 3DBBs. As shown in Fig. 2, BoxerNet (Sec. 3.2) is the core module and produces per-frame 3DBB detections from a single image and open-world 2DBBs from an off-the-shelf detector (Sec. 3.1). BoxerNet can also take optional depth input to improve performance. Per-frame 3DBB detections are merged and filtered using a hand-engineered fusion algorithm (Sec. 3.3).

3.1 2D Detection

The Boxer pipeline first uses an off-the-shelf open-world 2D bounding-box (2DBB) detector to localize text queries in the image. We denote by $\mathcal{T} = \{t_j\}_{j=1}^{M}$ a set of natural-language text prompts used to query an open-world 2D bounding-box detector. Given an image $I \in \mathbb{R}^{H \times W \times 3}$ and a prompt t_i, the detector produces a set of 2D bounding boxes $\mathcal{B}^{2D} = \{b_i\}_{i=1}^{N}$ corresponding to image regions semantically aligned with the query, as a single text query can produce multiple detections. Note we trivially handle grayscale images by repeating the single channel three times. Each box b_i is parameterized by its corner coordinates and an associated 2D confidence score $s_i^{2D} \in [0, 1]$.

3.2 BoxerNet: Lifting 2D to 3D

Overview. We show the BoxerNet network design in Fig. 3. Each 2D detection $b_i^{2D} \in \mathcal{B}^{2D}$ is mapped to a 3D detection $b_i^{3D} \in \mathcal{B}^{3D}$ using a learned model f_{lift} that predicts a 3D object hypothesis conditioned on the image and the 2D box. Specifically, the model outputs a 7-DoF 3DBB

$$b_i^{3D} = (x_i, y_i, z_i, w_i, h_i, d_i, \theta_i) \ , \tag{1}$$

where $(x_i, y_i, z_i) \in \mathbb{R}^3$ denotes the object center in the gravity-aligned camera coordinate frame $\mathcal{F}_g$, $(w_i, h_i, d_i) \in \mathbb{R}^3_+$ its physical extent, and $\theta_i \in [-\pi, \pi)$ a single rotation about the gravity axis. Each lifted prediction is additionally associated with a confidence score $s_i^{3D} \in [0, 1]$, indicating the model's confidence in the 3D hypothesis. We choose a 7-DoF representation because the datasets commonly provide a gravity estimate via an IMU, but it is straightforward to extend this to a 9-DoF 3DBB representation (*i.e.*, estimating a quaternion instead of a single scalar).

Lifting encoder. The lifting model f_{lift} is implemented as a self-attention encoder that jointly reasons over appearance, scene depth, and camera calibration. Given an input image I, we extract dense visual features $F^{\text{img}} \in \mathbb{R}^{H' \times W' \times D}$ using a pretrained DINOv3 [41] backbone. To encode metric scale into the model, we optionally project sparse point clouds or dense depth points from a depth sensor into the image plane using known camera intrinsics and compute the mean depth within each image patch, yielding a per-patch depth feature $F^{\text{depth}} \in \mathbb{R}^{H' \times W' \times 1}$. If no point projects into a patch, we set that patch to -1. Camera calibration and pose information are encoded as ray features $F^{\text{ray}} \in \mathbb{R}^{H' \times W' \times 3}$ following [44], where each ray encoding represents the normalized 3D ray associated with an image patch in the gravity-aligned coordinate frame $\mathcal{F}_g$, computed by unprojecting the patch center. Camera translation is omitted because it is always set to the origin $(0, 0, 0)$. Sparse point clouds with per-image visibility are readily available in SLAM or SfM systems [7, 13, 40]. This differs from CuTR, which requires a specialized ViT that processes dense depth only.

If 6-DoF pose is not available, the 2-DoF gravity direction (obtained by an IMU or off-the-shelf estimator [43]) could be used instead of the full pose for single-image box lifting, though all datasets we tested with provide the full pose. The three feature modalities are concatenated per patch $F^{\text{enc}} \in \mathbb{R}^{H' \times W' \times D+R+1}$ and fed as tokens into a shared self-attention encoder which allows the semantic features to mix with the ray and optional depth values. Including the rays and depth as separate channels enables conditioning on intrinsics with or without depth.

2D-to-3D box decoder. Following self-attention encoding, the lifted 3D predictions are obtained by conditioning on the 2D detections via cross-attention. Each 2D box $b_i^{(j)}$ is first lifted to the same feature dimension as the patch tokens using a linear layer (4 to hidden dimension), and then cross-attended over all encoder tokens (*i.e.* all patches), producing a box-specific latent representation.

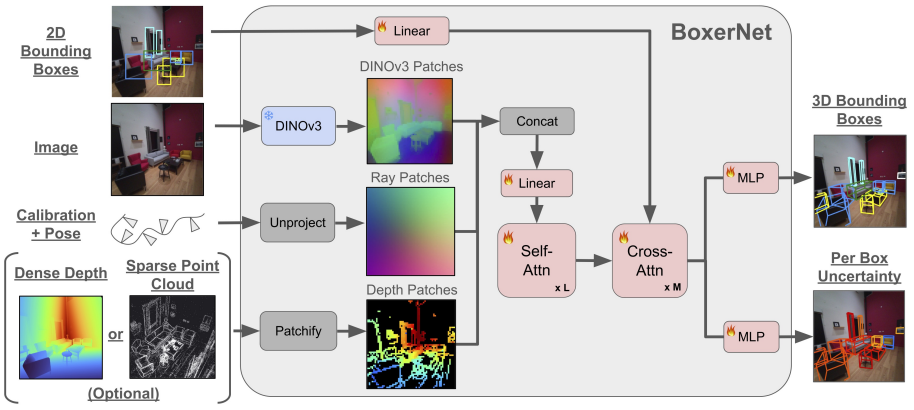

Fig. 3: BoxerNet lifting module. BoxerNet conditions on the image, camera calibration and poses and an optional depth input to lift 2D bounding boxes into metric 7-DoF 3D bounding box predictions.

Each box attends independently to the patch tokens, with no attention between box tokens, making the formulation permutation invariant. This representation is passed to two prediction heads. Each head is a two-layer MLP with ReLU and 128 hidden dimensions.

Outputs. The first head regresses the 7-DoF 3D bounding box parameters $\hat{b}_i^{3D} = (\hat{x}_i, \hat{y}_i, \hat{z}_i, \hat{w}_i, \hat{h}_i, \hat{d}_i, \hat{\theta}_i)$. The second head predicts an aleatoric uncertainty value $\hat{\sigma}_i$ [16], which captures observation noise and ambiguity in the lifting process. The final confidence score s_i for each box is the mean of the 2D and 3D confidences:

$$s_i = (s_i^{2D} + s_i^{3D})/2 \ , \tag{2}$$

which is used during inference to rank the final detections for filtering and generating the PR curves needed in the mAP computation.

Training objective. The lifting model is trained using a loss function that combines geometric alignment with aleatoric uncertainty [16] modeling. Specifically, we supervise the predicted 3D bounding box using a Chamfer loss $\mathcal{L}_{\text{chamfer}}$ between the predicted and ground-truth box corners. We also add the 3D uncertainty term as in [6, 25]. The final training objective is given by

$$\mathcal{L} = \mathcal{L}_{\text{chamfer}} \cdot \exp(-\hat{\sigma}) + \hat{\sigma} \ , \tag{3}$$

where $\hat{\sigma}$ denotes the predicted aleatoric uncertainty from above.

3.3 Multi-View Temporal Fusion

The final step estimates a set of per-scene de-duplicated, global, static 3DBBs from per-frame 3DBB estimates across time and views. Per-frame 3D detections are

Table 1: Comparison of datasets used for developing Boxer. We use both internal and public data sources.

Dataset Name	Device	Availability	Unique 3DBB	Images
-	Project Aria	Internal	65k	1.1M
-	Quest	Internal	400k	6.4M
NymeriaPlus	Project Aria	Public	240k	8.6M
CA-1M	iPad	Public	440k	13M
ScanNet	iPad	Public	14k	2.5M
SUN-RGBD	Kinect	Public	65k	10.5k

aggregated into a consistent set of scene-level object hypotheses via a geometric and semantic fusion pipeline. Given the set of lifted 3D bounding boxes across time, merging is first restricted to geometrically compatible detections using a 3D IoU threshold, preventing fusion of spatially distant objects. Semantic consistency is enforced by allowing merges only between detections with similar category labels computed on the prompt text via a text embedding [35]. A graph is then constructed whose nodes correspond to detections and whose edges connect geometrically and semantically compatible pairs. Connected components define object-level clusters spanning multiple frames. Within each cluster, confidence-weighted fusion of box parameters is performed. To account for the 90° rotational symmetry of gravity-aligned boxes, orientation ambiguity is resolved before averaging position, dimensions, and yaw (using a circular mean). Finally, 3D non-maximum suppression is applied to the fused hypotheses to remove residual duplicates, yielding a compact set of scene-level 3D object detections. Additional implementation details are provided in the supplementary material.

4 Training Dataset

Overall statistics. We train Boxer on both internal and public data. The internal non-public dataset consists of Project Aria (Gen1 [13] and Gen2 [33]) and Quest3 data. For public data, we use NymeriaPlus [12], which has additional annotation on top of [26] and CA-1M [20]. We also source data from Omni3D by including SUN-RGBD and ScanNet [11] (using 3DBBs from Scan2CAD [2]). The included datasets are shown in Tab. 1. We focus on indoor datasets because they are an important use case for everyday human and robotic applications. We exclude other Omni3D indoor datasets: ARKit [5] due to overlap with CA-1M, Objectron [1] due to its single-object focus, and Hypersim [36] because it is synthetic. Summed across these sources, this yields roughly 1.22M unique 3DBBs and 42.1 million image views.

Note that we focus on *unique* 3D bounding boxes when quantifying the scale of such datasets. This is because with posed video datasets, it is trivial to record a long static video, annotate a single 3DBB, and annotate a thousand 3DBB *observations*. Thus we separate the statistics into unique 3DBBs and unique image views.

Per-view visibility. Since per-view 2D annotation is costly and the training scenes are static, some datasets only have static 3D bounding boxes with human annotation. Naively projecting all 3D bounding boxes into each image does not account for occlusion (*i.e.* boxes will be visible through walls). To compute the per-frame visibility of each 3D bounding box with respect to each posed image, we use the observability of the available depth, making sure at least two visible depth points lie inside the 3DBB. In addition, we require sufficient visibility of the 3D bounding box geometry by enforcing that 80% of sampled points along the box edges are visible in the valid region of the image.

Data augmentation. We apply four types of data augmentation to the encoder inputs to improve robustness and generalization: photometric, camera, depth, and 2D box augmentation. See the supplemental material for details.

Training and inference details. The model is trained on the dataset described in Sec. 4. It is trained for roughly two weeks on 16 H100 GPUs using the AdamW optimizer with a cosine-decaying learning rate that starts at 1e-4 and decays to 1e-5. BoxerNet lifting, given 2DBB detection, takes roughly 20 ms for a forward pass on 960×960 images using bfloat16 on an NVIDIA RTX 4090 and has about 25 million trainable parameters (excluding frozen DINO weights).

5 Experiments

Our experiments are split into three parts: (1) per-frame single-image evaluation with optional depth (dense depth or sparse point clouds), which computes metrics on 3DBBs predicted independently for each frame; (2) per-scene evaluation, which runs the fusion described in Sec. 3.3 on each method and computes metrics on the final set of estimated 3DBBs; and (3) sensitivity studies to analyze the key factors in Boxer's design.

5.1 Testing Datasets

NymeriaPlus [12]. The NymeriaPlus dataset provides a test set that is exhaustively annotated with open-world 3D bounding boxes. The data comes from real-world homes captured by egocentric glasses. This dataset does not have external ground-truth sensors (e.g., no dense depth). Thus, we can only compare methods that operate on image-only input or image + sparse point clouds from the SLAM system.

Aria Digital Twin [31]. We use the Aria Digital Twin (ADT) dataset, which contains ground-truth depth, to compare models that can use depth as input (e.g., CuTR (Image-D)). This dataset contains one indoor apartment scene collected from egocentric data. We select a single sequence from this dataset. The sequence includes a small amount of dynamics (<5% of observed 3DBBs). We exclude such dynamic objects from evaluation. For GT2D, we prompt only with 2DBBs from static objects.

CA-1M [20]. We use the first ten sequences from the validation set to measure performance on CA-1M. We use the provided rendered/cropped 3DBBs for testing, as well as provided 2DBBs for GT2D prompt inputs. We exclude very large structural objects such as floors and walls, or any object with a dimension > 3 m, as these are typically not observable in a single image. We additionally expand the dimensions of thin objects (e.g., lights) to at least 5 cm for both predictions and ground truth.

Omni3D [6]. We also report results for our model on the closed-world Omni3D dataset, focusing on the SUN-RGBD split. We use the provided 2D boxes for GT2D prompts, as well as provided dense depth for depth-based models.

5.2 Benchmarking Protocols

Open-world 2DBB detectors. We primarily experiment with two open-world 2DBB detectors: DETIC [52] and OWLv2 [29]. We also evaluate SAM3 [8], but its compute cost is high with 1000+ prompts, taking 45 s per image on an NVIDIA RTX 4090, versus 35 ms for DETIC and 120 ms for OWLv2, making it less suitable for long sequences with thousands of images. We also use ground-truth 2D bounding boxes (GT2D) in some experiments to focus evaluation on the 3D lifting capability of each method.

Text prompts. For open-world datasets (NymeriaPlus, ADT, and CA-1M), we use a generic open-world prompt consisting of LVIS (1202 categories) plus 17 additional common indoor-focused categories. See the supplemental material for the exact list. In Omni3D, which is a closed-set 3DBB dataset, we prompt open-world models with the limited taxonomy of 50 classes, as done in [6].

Metrics and protocol. We report the performance of 3D detectors at the scene level after per-frame detections are fused into a global static 3D coordinate frame. We compute precision-recall curves for 3D IoU at thresholds [0.05, 0.1, 0.15, ..., 0.5] and average them to compute mAP, as done in prior work [6]. Unlike some prior setups, we do not limit detections to 100; instead, we run each detector at a relatively low threshold and report all resulting detections. We report class-agnostic results, meaning we discard semantic labels from each method and treat all boxes as a single "Anything" class.

Explanation of combinations in Tab. 2. To make comparisons as fair as possible, we focus evaluation on each model's ability to take a 2D box and lift it to 3D. BoxerNet does not provide 2D detections itself, so it requires 2DBB inputs. For methods that already use a 2DBB detector, this is straightforward (e.g., OWLv2+BoxerNet). For methods that are 3DBB detectors, such as 3D-MOOD [48], we first run their 3DBB detector, then project each 3DBB back into the 2D image using known camera pose and calibration to generate a 2DBB, and then prompt BoxerNet with it. We follow the same procedure for CubeRCNN. This reduces compounding variables such as taxonomy and training-set differences.

Table 2: Per-frame 3D detection results. All numbers report class-agnostic mAP. Methods are grouped by Image-only and Image+Depth. For NymeriaPlus, since depth is not available, sparse point cloud from SLAM is used. **Bold** indicates the best method excluding GT2D and underline the best lifting method for a given 2DBB input.

		Test Set (mAP per-frame)			
Method	Modalities	NymeriaPlus	ADT	CA-1M	$\text{Omni3D}_{\text{SUN}}$
DETIC+BoxerNet	RGB	0.110	0.026	0.053	0.098
3D-MOOD	RGB	0.014	0.003	0.046	0.271
3D-MOOD+BoxerNet	RGB	**0.092**	0.013	**0.099**	**0.340**
CubeRCNN	RGB	0.030	0.015	0.042	0.264
CubeRCNN+BoxerNet	RGB	**0.092**	0.030	0.075	0.286
OWLv2+CuTR	RGB	0.005	0.001	0.064	0.078
OWLv2+BoxerNet	RGB	0.061	**0.041**	0.081	0.144
GT2D+CuTR	RGB	0.010	0.004	0.119	0.160
GT2D+BoxerNet	RGB	0.296	0.077	0.126	0.293
DETIC+BoxerNet	RGB+D	0.193	0.120	0.153	0.200
3D-MOOD+BoxerNet	RGB+D	0.166	0.040	0.174	0.449
CubeRCNN+BoxerNet	RGB+D	0.120	0.039	0.104	**0.450**
EVL	RGB+D	0.100	-	-	-
OWLv2+CuTR	RGB+D	-	0.052	0.178	0.160
OWLv2+BoxerNet	RGB+D	**0.297**	**0.124**	**0.204**	0.290
GT2D+CuTR	RGB+D	-	0.102	0.250	0.219
GT2D+BoxerNet	RGB+D	0.532	0.317	0.412	0.585

Table 3: Per-scene (fused per-frame) 3D detection results. All numbers report class-agnostic mAP. Methods are grouped by Image-only and Image+Depth inputs (NymeriaPlus uses sparse point cloud from SLAM). We focus on video datasets and the CuTR baseline as it is most similar to BoxerNet and capable of running in real-time. Underline indicates the best lifting method for a given 2DBB input.

		Test Set (mAP per-scene)		
Method	Modalities	NymeriaPlus	ADT	CA-1M
OWLv2+CuTR	RGB	0.013	0.004	0.064
OWLv2+BoxerNet	RGB	0.145	0.077	0.081
GT2D+CuTR	RGB	0.013	0.005	0.087
GT2D+BoxerNet	RGB	0.219	0.107	0.089
OWLv2+CuTR	RGB+D	-	0.198	0.178
OWLv2+BoxerNet	RGB+D	0.278	0.215	0.204
GT2D+CuTR	RGB+D	-	0.244	0.305
GT2D+BoxerNet	RGB+D	0.400	0.307	0.434

5.3 Per-Frame Experiments

For all baseline methods, BoxerNet improves 3DBB estimation. Compared to CuTR, BoxerNet significantly outperforms it on NymeriaPlus in all configurations, even in image-only settings with ground-truth 2DBBs (0.296 vs. 0.010). One likely factor is that CuTR was not trained on egocentric data. Since NymeriaPlus only has sparse point-cloud depth, we do not run the CuTR RGB-D model and leave it as a dash (−). However, even on CA-1M, where both methods are trained and dense depth is available, we still see a gap (0.412 vs. 0.250). A qualitative

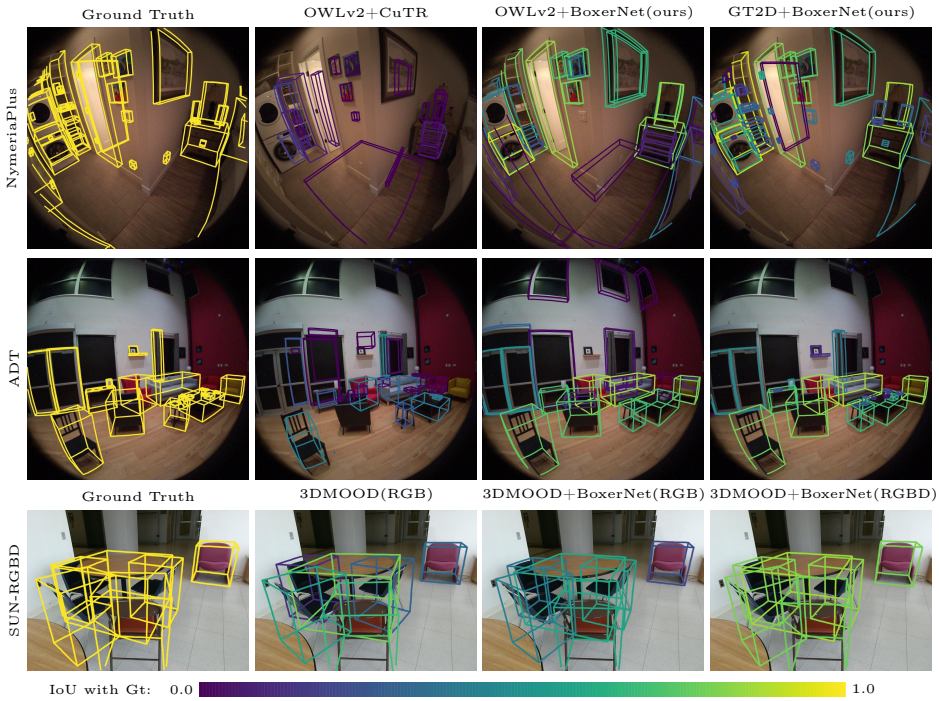

Fig. 4: Per-frame 3D IoU visualization. First two rows: We visualize the 3D IoU overlap of the ground truth (left) with OWLv2+CuTR (middle left), OWLv2+BoxerNet (middle right), and GT2D+BoxerNet (right). Predictions are colored with a viridis colormap (yellow is better). Boxer estimates have higher 3D IoU with the ground-truth boxes, as shown by the more yellow boxes. **Last row:** We compare 3D-MOOD and BoxerNet using 3D-MOOD 2D boxes on the SUN-RGBD dataset.

visualization comparing CuTR and BoxerNet is shown in Fig. 4. See the ablation study in Sec. 5.6 for further analysis.

Compared to 3D-MOOD, BoxerNet also improves performance in both image-only and image+depth settings. BoxerNet uses a DINOv3 backbone, which has been shown to exhibit strong monocular depth priors across large datasets [41]. This may help explain why it outperforms 3D-MOOD, which explicitly learns an auxiliary depth map on a smaller dataset.

5.4 Per-Scene Experiments

We focus on video datasets (excluding SUN-RGBD) and compare primarily to CuTR, since it is the closest baseline to BoxerNet and runs in real time. As shown in Tab. 3, the trend is consistent with the per-frame results: BoxerNet improves performance in all cases. Fig. 5 compares the per-frame predictions between CuTR and Boxer lifted into a consistent global map, demonstrating that Boxer exhibits more consistent predictions between frames.

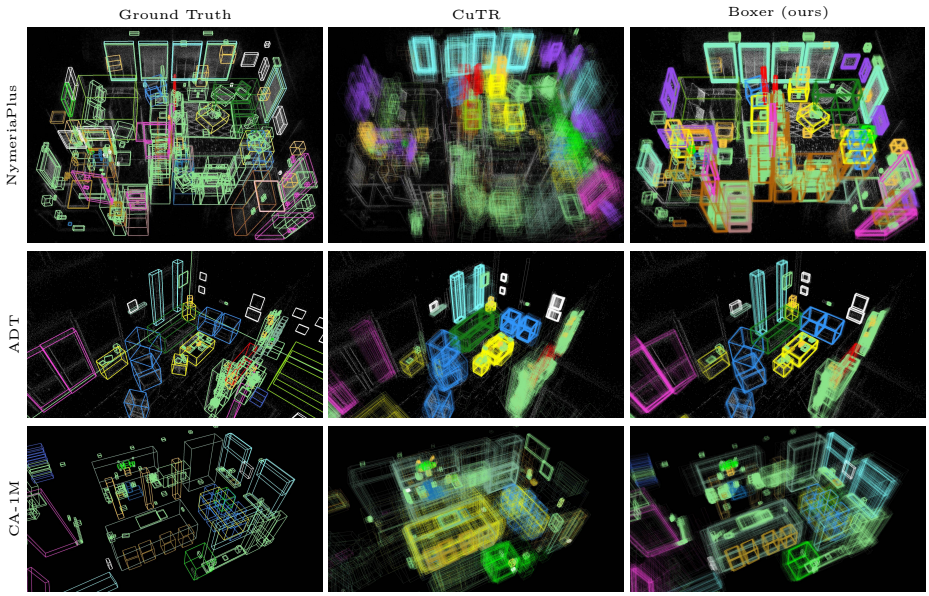

Fig. 5: Lifted per-frame 3DBB pseudo-heatmaps. We compare ground-truth per-scene 3DBBs (left) with per-frame 3DBBs from CuTR (middle) and Boxer (right), prompted with GT 2DBB input and lifted into a consistent coordinate frame. Boxes are rendered on top of one another to create a pseudo-heatmap. Boxer exhibits a sharper heatmap than CuTR, indicating more consistent predictions. Colors loosely correspond to object semantic class.

5.5 Multi-View Fusion Experiments

We compare Boxer's clustering + geometric averaging fusion with BoxFusion's stochastic fitting approach of [19]. Both are non-learned. On CA-1M scenes, Boxer fusion outperforms BoxFusion in terms of mAP, precision and recall. BoxFusion was not designed for BoxerNet's output distribution, and extra tuning may improve it.

Method	mAP	P@0.2	R@0.2
OwlV2 + BoxerNet + Fusion (ours)	**0.204**	**0.547**	**0.421**
OwlV2 + BoxerNet + BoxFusion	0.176	0.321	0.309

5.6 Ablation and Sensitivity Studies

We study key design decisions in the BoxerNet training recipe. We evaluate the 2D-to-3D lifting capability of the model on both egocentric and non-egocentric data sources. Ground-truth (oracle) 2D bounding boxes are used as input to best isolate the performance of the 3D lifting model.

Table 4: Ablation and Sensitivity Study. We compare removing individual design components in BoxerNet using GT 2DBB inputs and show per-frame mAP ↑.

Variant	NymeriaPlus	CA-1M
Full BoxerNet	**0.518**	**0.412**
median (instead of mean) depth aggregation	0.505	0.402
w/o data augmentation	0.497	0.400
w/o aleatoric uncertainty head	0.485	0.401
half image resolution (480x480 vs 960x960)	0.466	0.395
only public training data	0.463	0.376
w/o depth	0.279	0.126
only CA-1M	0.002	0.357

Analysis of Tab. 4. We ablate different design choices in BoxerNet. Removing point-cloud input has a significant effect, reducing performance from 0.518 to 0.279 mAP on the NymeriaPlus test set. Replacing the mean aggregation with a median aggregation in the depth patch aggregation reduces NymeriaPlus mAP by 0.013 mAP. Removing aleatoric uncertainty also hurts performance, from 0.518 to 0.485. There is a large domain gap between CA-1M and Aria data, since performance drops from 0.518 to 0.002 when training only on CA-1M. Since BoxerNet is trained partly on internal data, we also measure the impact of removing internal data from training: performance drops from 0.518 to 0.463 on NymeriaPlus and from 0.412 to 0.376 on CA-1M.

5.7 Limitations

BoxerNet was trained on static objects, and the fusion system assumes a static world, so dynamic objects (*e.g.,* in-hand objects) do not work very well. Highly non-cuboidal objects (*e.g.,* wires, plant vines) are not well represented by a 3DBB, and the model is not expected to perform well in these circumstances. BoxerNet lifting requires calibrated input and gravity. If not available they could be acquired from off-the-shelf methods such as [43], which we leave to future work to explore. OWLv2 and DETIC make mistakes and are not trained heavily on egocentric data. Boxer inherits these limitations since it is based on these open-set 2D detectors.

6 Conclusion

We presented Boxer, an algorithm for generating global, static, and de-duplicated 3D object bounding boxes from posed video sequences with optional depth. By combining an open-vocabulary 2D detector with BoxerNet, a learned 2D-to-3D lifting model trained on a large dataset consisting of multiple camera types, Boxer produces stable 3D object bounding boxes for open-world scenes. BoxerNet itself can be used as a drop-in replacement to existing 2D or 3D object detection pipelines to improve 3DBB accuracy. A multi-view temporal fusion module integrates per-frame predictions into a unified global map, using 3D IoU and

heuristic rules to merge redundant instances and refine object estimates. Overall, Boxer provides a practical and scalable approach for constructing consistent scene-level 3D object representations from posed image data.

7 Acknowledgments

We thank Pierre Moulon for feedback in group discussions, Manuel Lopez Antequera for support with internal annotations, Dan Barnes and Raul Mur-Artal for support with tooling for annotation, and Yawar Siddiqui for valuable discussions and feedback. We also thank Austin Kukay, Rowan Postyeni, Ruosha Pang, Mu Cheng, William Sun, Chen Zhang, Qinyue He, Aaron Deguzman, Yao Zhi, Luis Pesqueira, Abha Arora, and Rana Hayek for annotation support.

References

1. Ahmadyan, A., Zhang, L., Ablavatski, A., Wei, J., Grundmann, M.: Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. CVPR (2021)
2. Avetisyan, A., Dahnert, M., Dai, A., Savva, M., Chang, A.X., Niessner, M.: Scan2CAD: Learning CAD model alignment in rgb-d scans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2619–2628 (2019)
3. Avetisyan, A., Xie, C., Howard-Jenkins, H., Yang, T.Y., Aroudj, S., Patra, S., Zhang, F., Frost, D., Holland, L., Orme, C., Engel, J., Miller, E., Newcombe, R., Balntas, V.: Scenescript: Reconstructing scenes with an autoregressive structured language model. In: Proceedings of the European Conference on Computer Vision (ECCV) (2024)
4. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
5. Baruch, G., Chen, Z., Dehghan, A., Dimry, T., Feigin, Y., Fu, P., Gebauer, T., Joffe, B., Kurz, D., Schwartz, A., et al.: Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. arXiv preprint arXiv:2111.08897 (2021)
6. Brazil, G., Kumar, A., Straub, J., Ravi, N., Johnson, J., Gkioxari, G.: Omni3D: A large benchmark and model for 3D object detection in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9630–9641 (2023)
7. Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multi-map SLAM. IEEE Transactions on Robotics **37**(6), 1874–1890 (2021)
8. Carion, N., Gustafson, L., Hu, Y.T., Debnath, S., Hu, R., Suris, D., Ryali, C., Alwala, K.V., Khedr, H., Huang, A., Lei, J., Ma, T., Guo, B., Kalla, A., Marks, M., Greer, J., Wang, M., Sun, P., Rädle, R., Afouras, T., Mavroudi, E., Xu, K., Wu, T.H., Zhou, Y., Momeni, L., Hazra, R., Ding, S., Vaze, S., Porcher, F., Li, F., Li, S., Kamath, A., Cheng, R., Dollar, P., Ravi, N., Saenko, K., Zhang, P., Feichtenhofer, C.: Sam 3: Segment anything with concepts. In: International Conference on Learning Representations (ICLR) (2026)
9. Chen, X., Chu, F.J., Gleize, P., Liang, K.J., Sax, A., Tang, H., Wang, W., Guo, M., Hardin, T., Li, X., et al.: Sam 3d: 3dfy anything in images. arXiv preprint arXiv:2511.16624 (2025)
10. Cho, J.H., Ivanovic, B., Cao, Y., Schmerling, E., Wang, Y., Weng, X., Li, B., You, Y., Krähenbühl, P., Wang, Y., et al.: Language-image models with 3d understanding. arXiv preprint arXiv:2405.03685 (2024)
11. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5828–5839 (2017)
12. DeTone, D., Bogo, F., Le, E.T., Frost, D., Straub, J., Siddiqui, Y., Ye, Y., Engel, J., Newcombe, R., Ma, L.: NymeriaPlus: Enriching nymeria dataset with additional annotations and data. arXiv preprint arXiv:2603.18496 (2026)
13. Engel, J., Somasundaram, K., Goesele, M., Sun, A., Gamino, A., Turner, A., Talattof, A., Yuan, A., Souti, B., Meredith, B., et al.: Project aria: A new tool for egocentric multi-modal ai research. arXiv preprint arXiv:2308.13561 (2023)

14. Gu, Q., Kuwajerwala, A., Morin, S., Jatavallabhula, K.M., Sen, B., Agarwal, A., Rivera, C., Paul, W., Ellis, K., Chellappa, R., et al.: Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 5021–5028. IEEE (2024)
15. Gu, Q., Lv, Z., Frost, D., Green, S., Straub, J., Sweeney, C.: Egolifter: Open-world 3d segmentation for egocentric perception. In: European conference on computer vision. pp. 382–400. Springer (2024)
16. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: Advances in Neural Information Processing Systems (2017)
17. Kong, C., Fort, J., Kang, A., Wittmer, J., Green, S., Shen, T., Zhao, Y., Peng, C., Solaira, G., Berkovich, A., et al.: Aria gen 2 pilot dataset. arXiv preprint arXiv:2510.16134 (2025)
18. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., et al.: The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. International journal of computer vision **128**(7), 1956–1981 (2020)
19. Lan, Y., Zhu, C., Gao, Z., Zhang, J., Cao, Y., Yi, R., Wang, Y., Xu, K.: Boxfusion: Reconstruction-free open-vocabulary 3d object detection via real-time multi-view box fusion. arXiv preprint arXiv:2506.15610 (2025)
20. Lazarow, J., Griffiths, D., Kohavi, G., Crespo, F., Dehghan, A.: Cubify anything: Scaling indoor 3d object detection. arXiv preprint arXiv:2412.04458 (2024)
21. Li, Y., Ge, Z., Yu, G., Yang, J., Wang, Z., Shi, Y., Sun, J., Li, Z.: Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In: Proceedings of the AAAI conference on artificial intelligence. vol. 37, pp. 1477–1485 (2023)
22. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Yu, Q., Dai, J.: Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. IEEE Transactions on Pattern Analysis and Machine Intelligence **47**(3), 2020–2036 (2024)
23. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 740–755 (2014)
24. Liu, Y., Wang, T., Zhang, X., Sun, J.: Petr: Position embedding transformation for multi-view 3d object detection. In: European conference on computer vision. pp. 531–548. Springer (2022)
25. Lu, Y., et al.: Geometry uncertainty projection network for monocular 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
26. Ma, L., Ye, Y., Hong, F., Guzov, V., Jiang, Y., Postyeni, R., Pesqueira, L., Gamino, A., Baiyya, V., Kim, H.J., Bailey, K., Fosas, D.S., Liu, C.K., Liu, Z., Engel, J., Nardi, R.D., Newcombe, R.: Nymeria: A massive collection of multimodal egocentric daily motion in the wild (2024), https://arxiv.org/abs/2406.09905, accessed: 2026-06-29
27. Man, Y., Wang, S., Zhang, G., Bjorck, J., Li, Z., Gui, L.Y., Fan, J., Kautz, J., Wang, Y.X., Yu, Z.: Locateanything3d: Vision-language 3d detection with chain-of-sight. arXiv preprint arXiv:2511.20648 (2025)
28. Mao, Y., Zhong, J., Fang, C., Zheng, J., Tang, R., Zhu, H., Tan, P., Zhou, Z.: Spatiallm: Training large language models for structured indoor modeling. In: Advances in Neural Information Processing Systems (2025)
29. Minderer, M., Gritsenko, A., Houlsby, N.: Scaling open-vocabulary object detection. In: Advances in Neural Information Processing Systems. vol. 36, pp. 72832–72859 (2023)

30. Misra, I., et al.: 3detr: An end-to-end transformer model for 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
31. Pan, X., Charron, N., Yang, Y., Peters, S., Whelan, T., Kong, C., Parkhi, O., Newcombe, R., Ren, Y.C.: Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 20133–20143 (2023)
32. Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: European conference on computer vision. pp. 194–210. Springer (2020)
33. Project Aria Team: Aria gen 2: An advanced research device for egocentric ai research. https://www.projectaria.com/ariagen2devicepaper (2025), accessed: 2026-03-02
34. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Votenet: Deep hough voting for 3d object detection in point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
35. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP). pp. 3982–3992 (2019)
36. Roberts, M., Ramapuram, J., Ranjan, A., Kumar, A., Bautista, M.A., Paczan, N., Webb, R., Susskind, J.M.: Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In: ICCV (2021)
37. Rukhovich, D., Vorontsova, A., Konushin, A.: Fcaf3d: Fully convolutional anchor-free 3d object detection. In: European Conference on Computer Vision. pp. 477–493. Springer (2022)
38. Rukhovich, D., Vorontsova, A., Konushin, A.: Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 2397–2406 (2022)
39. Rukhovich, D., Vorontsova, A., Konushin, A.: Tr3d: Towards real-time indoor 3d object detection. In: 2023 IEEE International Conference on Image Processing (ICIP). pp. 281–285. IEEE (2023)
40. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4104–4113 (2016)
41. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., et al.: Dinov3. arXiv preprint arXiv:2508.10104 (2025)
42. Straub, J., DeTone, D., Shen, T., Yang, N., Sweeney, C., Newcombe, R.: Efm3d: A benchmark for measuring progress towards 3d egocentric foundation models. arXiv preprint arXiv:2406.10224 (2024)
43. Veicht, A., Sarlin, P.E., Lindenberger, P., Pollefeys, M.: Geocalib: Learning single-image calibration with geometric optimization. In: European Conference on Computer Vision. pp. 1–20. Springer (2024)
44. Wang, Y., Doersch, C., Arandjelovic, R., Carreira, J., Zisserman, A.: Input-level inductive biases for 3d reconstruction. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6166–6176 (2022)
45. Wang, Y., Guizilini, V.C., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In: Conference on robot learning. pp. 180–191. PMLR (2022)

46. Wang, Y., Ke, L., Zhang, B., Qu, T., Yu, H., Huang, Z., Yu, M., Xu, D., Yu, D.: N3d-vlm: Native 3d grounding enables accurate spatial reasoning in vision-language models. arXiv preprint arXiv:2512.16561 (2025)
47. Xie, Y., Jiang, H., Gkioxari, G., Straub, J.: Pixel-aligned recurrent queries for multi-view 3D object detection. In: ICCV (2023)
48. Yang, Y.H., Piccinelli, L., Segu, M., Li, S., Huang, R., Fu, Y., Pollefeys, M., Blum, H., Bauer, Z.: 3d-mood: Lifting 2d to 3d for monocular open-set object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7429–7439 (2025)
49. Yao, J., Gu, H., Chen, X., Wang, J., Cheng, Z.: Open vocabulary monocular 3d object detection. arXiv preprint arXiv:2411.16833 (2024)
50. Zhang, H., Jiang, H., Yao, Q., Sun, Y., Zhang, R., Zhao, H., Li, H., Zhu, H., Yang, Z.: Detect anything 3d in the wild. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5048–5059 (2025)
51. Zhang, Z., Sun, B., Yang, H., Huang, Q.: H3dnet: 3d object detection using hybrid geometric primitives. In: European conference on computer vision. pp. 311–329. Springer (2020)
52. Zhou, X., Girdhar, R., Joulin, A., Krähenbühl, P., Misra, I.: Detecting twenty-thousand classes using image-level supervision. In: Proceedings of the European Conference on Computer Vision (ECCV) (2022)