DetAny4D: Detect Anything 4D Temporally in a Streaming RGB Video

Jiawei Hou[1], Shenghao Zhang[2,*], Can[3] Wang, Zheng Gu[4], Yonggen Ling[2],
Taiping Zeng[1], Xiangyang Xue[1,*], Jingbo Zhang[2,*]

Abstract

Reliable 4D object detection, which refers to 3D object detection in streaming video, is crucial for perceiving and understanding the real world. Existing open-set 4D object detection methods typically make predictions on a frame-by-frame basis without modeling temporal consistency, or rely on complex multi-stage pipelines that are prone to error propagation across cascaded stages. Progress in this area has been hindered by the lack of large-scale datasets that capture continuous reliable 3D bounding box (b-box) annotations. To overcome these challenges, we first introduce DA4D, a large-scale 4D detection dataset containing over 280k sequences with high-quality b-box annotations collected under diverse conditions. Building on DA4D, we propose DetAny4D, an open-set end-to-end framework that predicts 3D b-boxes directly from sequential inputs. DetAny4D fuses multi-modal features from pre-trained foundational models and designs a geometry-aware spatiotemporal decoder to effectively capture both spatial and temporal dynamics. Furthermore, it adopts a multi-task learning architecture coupled with a dedicated training strategy to maintain global consistency across sequences of varying lengths. Extensive experiments show that DetAny4D achieves competitive detection accuracy and significantly improves temporal stability, effectively addressing long-standing issues of jitter and inconsistency in 4D object detection. Data and code will be released upon acceptance.

1. Introduction

3D object detection serves as a pivotal task to perceive the world. Previous methods have made significant strides in predicting the 3D bounding box (b-box) from single snapshots [5, 29, 41] and pre-scanned point clouds [21, 33]. However, when applied to streaming videos, these methods are either prone to error propagation and accumulation or depend on prior global environment scanning. Since streaming 3D bounding box prediction is crucial for reliable long-term reasoning, predicting future states, and ensuring stable decision-making in real-world applications, we focus on this task and formulate it as 4D object detection, which integrates 3D information with temporal awareness.

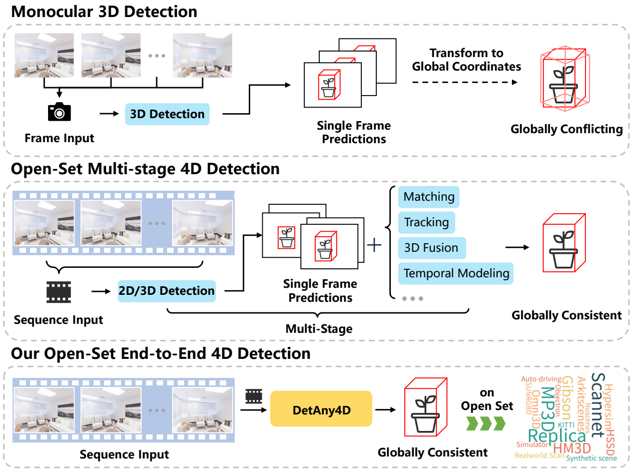


Figure 1. Comparison with existing methods. Existing 3D detectors predict on a frame-by-frame basis, which causes inconsistency when transforming 3D b-boxes into global coordinates. Current open-set 4D detectors typically address 3D predictions and cross-frame relationships in a multi-stage manner, which is complex and prone to error propagation across cascaded stages. In contrast, we propose an open-set end-to-end 4D detection benchmark that directly predicts globally consistent 3D b-boxes.

An intuitive solution is to directly apply the single-frame 3D detection methods [5, 29, 41] to 4D detection (Figure 1-top), but this would lead to inconsistency because temporal relationships are not considered. In response to this, another approach [13, 34] to 4D object detection is a multi-stage pipeline that post-processes single-frame detections with tracking and 3D association modules to enforce temporal coherence (Figure 1-middle). Although feasible, these approaches often suffer from cumbersome design and limited

[*] Co-corresponding author

[1] Jiawei Hou and Xiangyang Xue are with School of Computer Science, Fudan University; Taiping Zeng is with Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University. *jwhou23@m.fudan.edu.cn* {*xyxue, zengtaiping*}*@fudan.edu.cn*

[2] Authors are with Tencent Robotics X, Tencent Binhai Building, Shenzhen, Guangdong 518057, China. {*popshzhang, eckertzhang, rolandling*}*@tencent.com*

[3] The University of Hong Kong. *canwang@hku.hk*

[4] Shenzhen University. *guzheng@szu.edu.cn*

robustness, leading to fragile behavior and error accumulation. Several autonomous-driving studies have extended 3D detection to the spatiotemporal setting by incorporating ego-motion estimation, velocity prediction, and multi-stage detectors with temporal-consistency learning [4, 14]. However, these methods are typically tailored to specific driving scenarios and cover only a limited range of object categories. Consequently, they either lack open-set detection capability or still depend on multi-stage pipelines that are susceptible to error propagation.

Despite these explorations, 4D object detection remains an open challenge: 1) A large-scale, high-quality 4D dataset with 3D b-box annotations is still lacking, largely due to the prohibitive cost and complexity of spatiotemporal annotation; 2) In temporal modeling, it remains unresolved how to build mechanisms that maintain long-term memory yet still detect objects accurately in newly arriving frames; 3) It is essential to devise methods that handle arbitrarily long sequences and deliver consistent, stable 3D detections under dynamically changing viewpoints.

We tackle the first challenge by introducing DA4D, a large-scale 4D object detection dataset. First, we collect large-scale RGB sequences with depth and camera poses by moving the camera in the Habitat simulator [32]. Next, we project object b-boxes in global coordinates to ego views and filter severely-occluded or out-of-view objects. Finally, we implement an adaptive strategy to adjust the 3D b-box annotations, taking into account factors such as visibility, occlusion, and historical observations, especially for objects with complex geometry. The DA4D dataset unifies 12 datasets of more than 280k sequences with spatiotemporal-aligned annotations.

Leveraging this large-scale dataset, we propose DetAny4D, an open-set end-to-end framework that addresses both remaining challenges, rather than relying on a complex multi-stage design (Figure 1-bottom). DetAny4D fuses visual features extracted by SAM [17] and DINO [22], which are then fed into the proposed Spatiotemporal Decoder, along with multi-modal prompt tokens and b-box tokens. DetAny4D further introduces a multi-head architecture for multi-task learning, enabling camera, depth, and 3D b-box estimations, while ensuring effective model training.

To enable effective training on sequential data and handle variable-length sequences during inference, we propose a dedicated training strategy with random cropping and object query padding for newly appearing objects. Furthermore, a dedicated loss function is designed to enforce global consistency among the b-boxes across the sequence.

Our contributions are summarized as follows:

- We introduce DA4D, a large-scale 4D object detection dataset consisting of 280k sequences, each with spatiotemporal-aligned 3D b-box annotations, depth data, and camera pose information.
- We propose DetAny4D, an open-set end-to-end 4D detection benchmark which predicts spatiotemporally aligned 3D b-boxes across frames.
- We propose a SpatioTemporal Decoder module integrated with a multi-task head framework and a tailored sequence-based training strategy, enabling robust cross-frame detections.
- Extensive evaluations show that DetAny4D achieves competitive 3D detection performance and a 10–30% reduction in cross-frame variance compared to single-frame detection solutions, while also exhibiting strong open-set detection capability, demonstrating its overall effectiveness on the 4D detection task.

2. Related Works

2.1. Monocular 3D Object Detection

Recently, advances in autonomous driving and embodied AI have made object detection a prominent focus in visual perception research. Early work primarily addressed the detection of specific categories of 3D objects within either indoor or outdoor scenes [8, 10, 15, 20, 37, 43]. Cube R-CNN [5] utilizes the comprehensive Omni3D dataset to pioneer unified monocular 3D object detection, significantly improving generalization across diverse scene types. UniMODE [19] further advances the field by introducing a BEV-based monocular 3D object detector capable of seamless operation in both indoor and outdoor environments, thereby enhancing adaptability. OVMono3D [40] innovates by decoupling 3D detection; it leverages a powerful 2D open-vocabulary detector for recognition, followed by a class-agnostic network for 3D lifting, enabling zero-shot detection of previously unseen categories. More recently, DetAny3D [41] proposes a foundation model for monocular 3D object detection by distilling knowledge from 2D foundation models. Although such methods achieve satisfactory results on single frames, their lack of temporal context in continuous world perception often causes jitter and inconsistency. To address this, our approach focuses on video-based 3D detection, modeling spatiotemporal relationships to produce globally consistent 3D b-boxes.

2.2. Video Object Detection

To achieve continuous world perception, many existing works focus on predicting continuous 3D bounding boxes in video sequences [4, 13, 14, 29]. For example, Kinematic3D [4], developed for autonomous driving, predicts 3D boxes frame by frame and subsequently applies ego-motion compensation and a Kalman filter to fuse detections across frames. BEVFormer [18] creates BEV representations by aggregating multi-camera features with a spatiotemporal transformer. TR3D [29] enables real-time 3D object detection for limited categories in indoor environ-

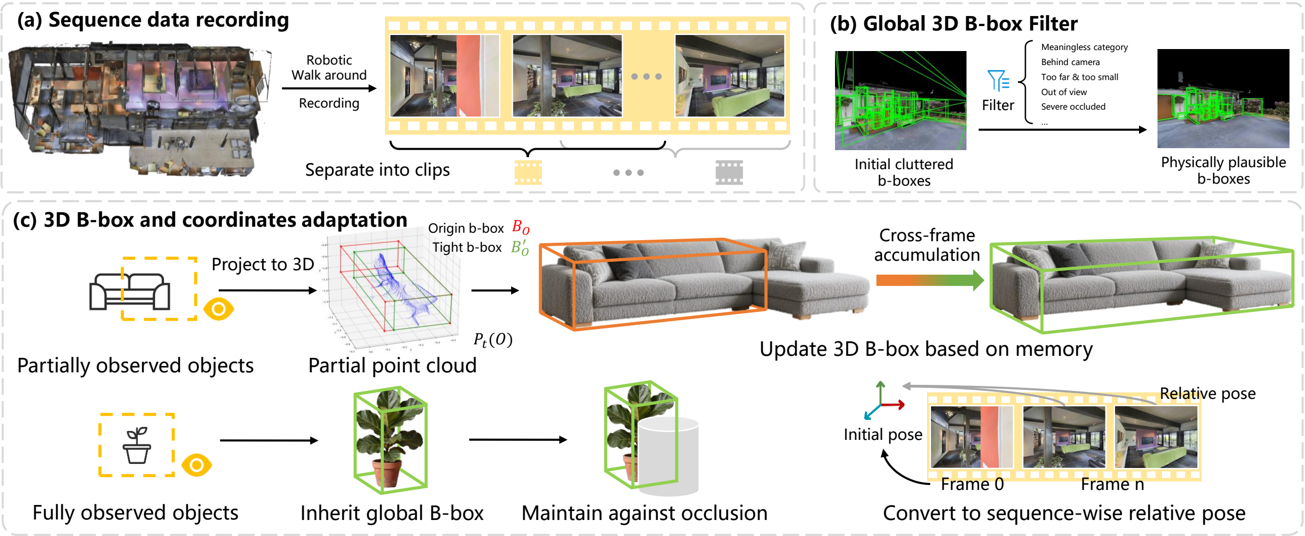


Figure 2. The data processing pipeline for 4D detection task. We record posed RGB frames sequentially and separate the records into fixed-length sequences. Objects in global coordinates are projected into ego view and filtered with policies to delete occluded and out-of-view objects. Objects b-boxes are then recalculated according to the visibility and accumulated considering the point cloud within the sequence. Finally, the coordinates of a sequence is adapted referring to the first frame.

ments through sparse convolution and lightweight design. BADet [14] adopts a two-stage process of initial detection and multi-frame fusion, incorporating global optimization of object centers and bundle adjustment of feature metrics to learn temporal dependencies. ConceptGraphs [13] and TAIROS [34] project 2D detections into 3D using depth, enabling category-wise fusion of object point clouds. However, these methods often rely on multi-stage pipelines or are restricted to specific scenarios, limiting their generalization. In contrast, our approach leverages open-vocabulary setting and an end-to-end framework for efficient, globally consistent 3D object detection in video sequences.

2.3. Point-Cloud-Based 3D Detection

To achieve globally consistent detections, another pipeline leverages complete 3D point clouds of an entire scene. Methods such as VoteNet [25] use Hough voting for object center proposals, H3DNet [42] integrates hybrid geometric primitives for greater accuracy, and RGBNet [36] models surface geometry with ray-based grouping. Multi-modal approaches like MTC-RCNN [23] fuse RGB images and point cloud features, while SpatialLM [21] employs large language models to convert point clouds to structured text, enabling interpretable and zero-shot detection. However, these methods require complete pre-scanned point clouds, resulting in high data and computational costs that limit scalability. In contrast, our approach produces consistent 3D bounding boxes directly from RGB sequences, achieving stronger generalizability without these constraints.

3. Data Generation Pipeline

We employ habitat simulator [32] to generate fine-labeled data for the 4D object detection task. As shown in Figure 2, the data processing pipeline consists of three main parts. First, we drive the robot to walk around in the virtual environment, record video sequences with an on-board camera, and divide them into fixed-lengh clips (Section 3.1). Meanwhile, Global coordinates object b-boxes are projected into ego frames and filtered according to designed strategy (Section 3.2). Then, the filtered b-boxes are adapted sequence-wise to get plausible ground-truth according to current view and memory. Coordinates are also adapted to relative coordinates in sequence (Section 3.3).

3.1. Sequence Data Recording

As shown in Figure 2(a), we move an onboard camera through the scene in a random walk pattern, recording RGB observations and corresponding depth and poses. The recorded sequences are then segmented into fixed-length clips using a stride that is shorter than the clip length itself. This ensures an overlap between consecutive clips. Such design is intentional to facilitate the learning of object states under diverse initial visibility conditions within a sequence. For instance, it allows the model to learn from scenarios where an object may start fully within the field of view and gradually exit it, or conversely, begin outside the frame and progressively enter it.

3.2. Global objects 3D B-box Filtering

To acquire globally consistent 3D object b-box as ground truth for the recorded clips, a natural idea it to project all

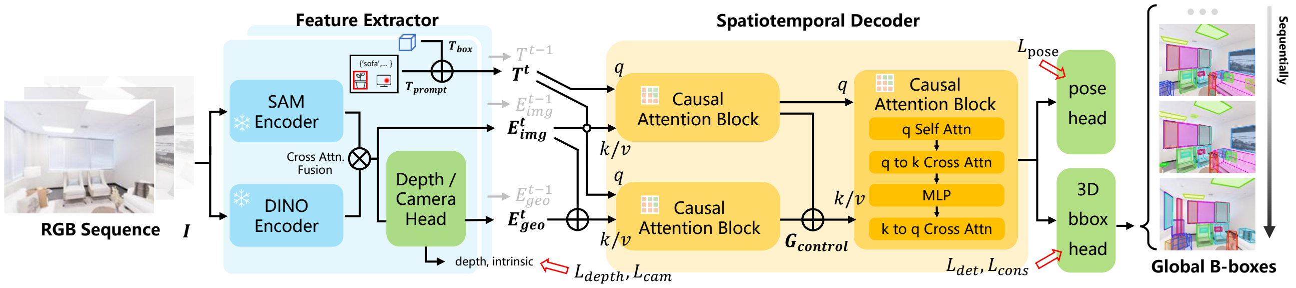
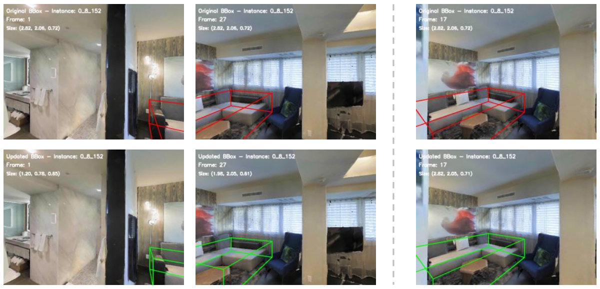


Figure 3. Pipeline of the proposed DetAny4D model. RGB sequence with prompts are encoded with the feature extractor, generating tokens T^t, image embeddings E^t_{img}, and depth and camera-related embeddings $E^t_{d,m,c}$. A Geometry Context Transformer then inject 3D space embeddings in a transformer control manner, and together with embeddings decoded by the Spatiotemporal Transformer to generate prediction results. Multi-task heads are employed for effective training.

objects from the world coordinates into the camera coordinates. However, such bare projection does not take severely occluded or out-of-view objects into consideration, which introduces significant artifacts (as shown in Figure 2(b)). Therefore, we apply a set of filtering rules to the initial cluttered b-boxes, keeping the physical plausibility while preserving the model's predictive capability on partially-occluded objects.

We first filter out objects according to the semantic information. We remove background and non-object categories (e.g., floor, wall, void) in each frame. Next, we exclude objects that are located behind the camera or far beyond a depth threshold. Then, we filter out objects with severe occlusion. We calculate the projected 2D b-box in the ego view. For each 2D b-box, we count the semantic-map pixels belonging to the object. Objects with pixel counts below a threshold are removed, indicating strong occlusion. We also project the eight vertices of each 3D b-box onto the current image plane and compare their depths with the depths of the corresponding 2D pixels. If more than five vertices lie behind their corresponding pixels, the object is considered heavily occluded and is filtered out.



(a) Partial Observed GT (b) Fully visible GT

Figure 4. Visualization of GT b-box adaptation strategy (Section 3.3). Taking an L-shaped sofa as example, red b-box denotes original GT and green denotes the adapted one. (a) shows adaptation of b-box when camera moves from left to right (column 1) and right to left (column 2) when sofa is partially observed. (b) shows when fully observed, adapted b-box aligns with original GT.

3.3. 3D B-box and Coordinates Adaptation

Unlike 3D detection, 4D detection exposes camera motion and changing viewpoints. As shown in Figure 2(c), considering an L-shaped sofa with a corner section, the initial global 3D b-box encompasses the entire sofa, including the corner. However, if the corner is not visible at the beginning, this global 3D b-box is intuitively unreasonable, and predicting it in early frames is physically implausible. A naive employment of such data leads to error propagation.

To further ensure physical credibility of the data, we propose a coordinate adaptation strategy to dynamically adjust the 3D b-boxes. If an object is fully visible at the start of a sequence, we maintain its original global 3D b-box until it completely disappears from the view or becomes permanently occluded. Conversely, if an object O is partially visible at the beginning, we back-project the masked visible pixels into 3D space using their depth values. From this partial point cloud $P_t(O)$, we compute a tight-fitting b-box B'_O, which is smaller but shares the same orientation as the global 3D b-box. As the sequence progresses, we incrementally maintain and expand this point cloud as

$$P_t(O) = P_{t-1}(O) \cup \pi^{-1}(Mt(O), D_O), \tag{1}$$

where π^{-1} is the projection function. At the same time, we enlarge the computed b-box correspondingly. Once the object has been completely observe, we use the original global b-box as the ground truth and maintain it in remaining frames. Such adaptation offers clear advantage for data construction. It enables our system to intelligently handle 3D objects with complex geometries, and also unlocks the model's capability to handle occlusions that evolve over time, which is crucial in temporal-aware 4D detection task. Figure 4 visulize the ground truth adaptation (Section 2) that leverages a more reasonable accumulative b-boxes strategy.

Furthermore, regarding coordinate system handling, we define the camera pose of the first frame in a sequence as the initial reference pose. The camera poses for all subsequent frames are then calculated and stored as relative transformations with respect to this initial pose, and these relative poses are provided for the training process.

Thus, processed by our pipeline, each sequence serves as an independent data instance. Its coordinate system is defined relative to the first frame, and its b-boxes are physically plausible for the current viewpoint. These boxes are updated incrementally based on the current observation and historical memory, which fully aligns with the objectives and logic of the 4D detection task.

4. Model Design

As illustrate in Figure 3, DetAny4D takes an RGB sequence as input. We first employ a feature extractor to encode frame-wise inputs into image embeddings and introduce prompt and box tokens (Section 4.1). These embeddings are then injected into a unified 3D space and aggregated across frames using a spatiotemporal decoder (Section 4.2). Multi-task heads are subsequently applied to supervise training and to produce temporally consistent 3D b-boxes for the queried objects throughout the sequence (Section 4.3). The overall training strategy and loss functions are specifically designed to address the challenges of sequence-level 4D object detection (Section 4.4)

4.1. Feature Extractor

To extract semantic and geometry information from the input images, we follow [41] and employ a feature extractor. Specifically, we adopt encoder modules from the pretrained SAM [17] and DINO [22] to capture priors from these 2D RGB sequence. These encoded features are further fused with a cross-attention-formed feature aggregator [41] to integrate multi-level embeddings E^t_{img}. To obtain geometric priors, we employ a depth and camera module, such as UniDepth-V2 [24], generating geometry-aware embedding $E^t_{geo} = \{E^t_{depth}, E^t_{cam}\}$, including depth embeddings E^t_{depth} and camera parameter embeddings E^t_{cam}. At the same time, it predicts metric depth and camera parameters. Meanwhile, the prompts are encoded to tokens T_{prompt} and concatenated with 3D b-box tokens T_{box} to form the prompt embedding T^t, which serves as query input for subsequent transformer modules.

4.2. Spatiotemporal Decoder

The spatiotemporal decoder accepts tokens and embbedings from sequence frames and generate spatiotemporal-aware implicit states for the following heads. The decoder compromises three Causal Attention Blocks (CAB). Simultaneously, one CAB module integrates tokens $T^{1,\dots,t}$ and image embeddings $E^{1,\dots,t}_{img}$ and another module leverages tokens $T^{1,\dots,t}$ and geometry-aware embeddings $E^{1,\dots,t}_{geo}$, generating geometry control embedding $G_{control}$. Then a CAB module fuse the output from the former two modules and generating the final implicit features for following heads.

Causal Attention Block. The causal attention block is a core component designed for modeling causal dependencies in sequential data, structured as a sequential stack of self-attention and bidirectional cross-attention, with causal masks integrated to strictly regulate information flow. Causal mask, implemented as a lower triangular matrix sized according to the input sequence length and feature count, is applied to ensure information from current frame can be aware of historical information but blind to future values, preventing information leakage. First, the self-attention module applies causal masking to q to perceive the historical status. Next, the bidirectional cross-attention guide causal-constrained information interaction between the tokens and embeddings. Overall, this design balances modeling flexibility and causal compliance with strict causal flow regulation, making it suitable for temporal-aware tasks.

4.3. Multi-task Heads

To enable effective model training, we design a multi-head architecture for multi-task supervision. In the feature extractor stage, we employ the depth and camera head to estimate metric depth and camera intrinsics [24]. Additionally, following the spatiotemporal decoding stage, we introduce a camera pose head and a 3D b-box head, respectively. These heads infer the relative camera pose and 3D b-boxes for each frame based on the fused features. This design allows the model to account for camera view transformations and predict 3D b-boxes in global coordinates.

4.4. Training

Training Strategy. To enable effective training on sequential data and provide flexibility for variable-length sequences during inference, we introduce a dedicated training strategy. For input sequences, we randomly crop them within a fixed maximum sequence length, which promotes diversity in sequence lengths during training. To maintain the model's ability for frame-wise detection, we also interleave single-frame 3D detection samples (represented as sequences of length one) into training batches. On the other hand, since the number of objects per frame is variable, we unify the number of prompts across all frames in a sequence before model input. We apply padding to the prompts to ensure the tokens from each frame have the same dimension in a sequence batch to be fed into the transformer. The loss function is calculated to disregard predictions associated with these padded prompts. During inference, the presence of these redundant tokens enables the model to detect newly appearing objects. Further details can be found in the supplementary materials.

Loss Functions. To achieve spatiotemporal attention across input frames, we design a series of loss functions that balance the various capabilities of our multi-task heads. we adopt widely used Scale-Invariant Logarithmic (SILog) loss [11, 35], denoted as L_{depth}. Similarly, following the

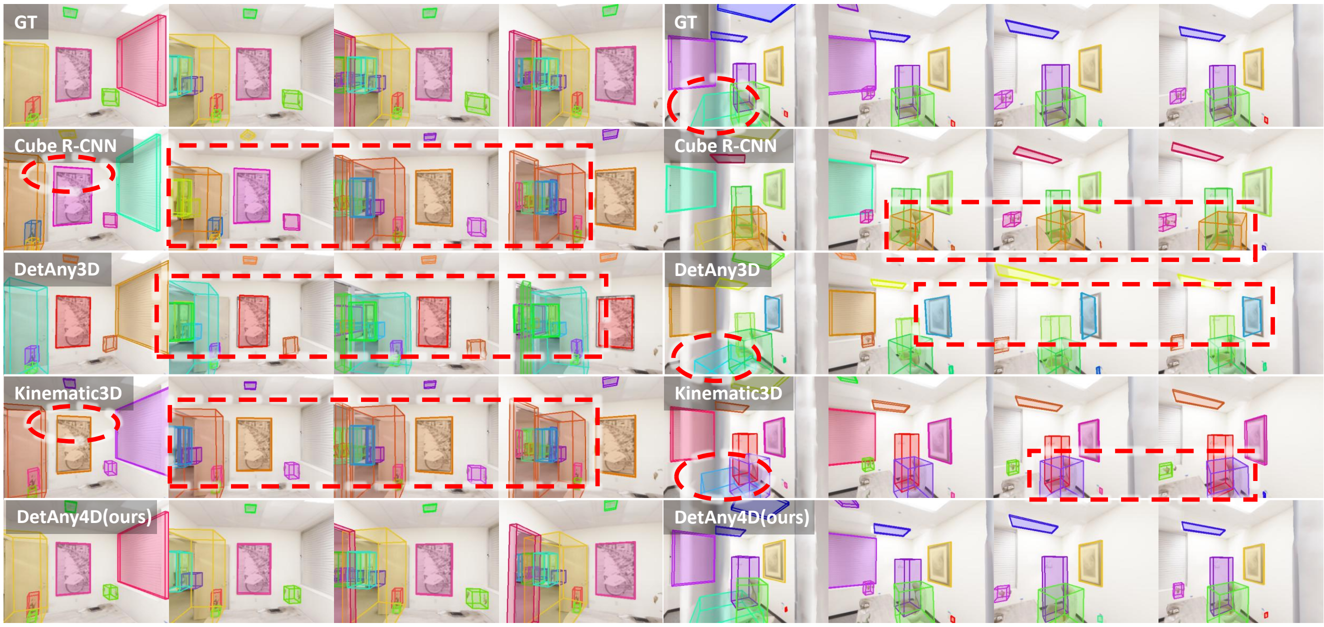


Figure 5. Qualitative comparison with other methods on 3D b-box predictions across consecutive frames in a sequence. Our proposed DetAny4D predicts spatiotemporally aligned 3D b-boxes, while red circles and rectangles show inaccurate and inter-frame jtter predictions.

approach in UniDepth [24], we supervise the camera intrinsics using the same SILog loss, denoted as L_{cam}.

The 3D b-box predictions are supervised with a combination of losses:

$$L_{det} = L_{center} + L_d + L_{IoU}^{2D} + L_{IoU}^{3D} + L_{corner} + L_{dim}, \quad (2)$$

where L_{center} and L_d are both L1 losses, used to constrain the center position and depth of the 3D b-box, respectively. L_{IoU}^{2D} and L_{IoU}^{3D} represent the 2D and 3D IoU losses [33], respectively, which are used to supervise the overall prediction quality. L_{corner} denotes a permutation-invariant Chamfer Distance loss [5, 39] applied to enforce rotation consistency. L_{dim} is a soft loss designed to supervise the box dimensions. Unlike previous 3D detection methods that predict 3D b-boxes from a single image, our 4D detection objective aims to predict globally consistent 3D boxes across consecutive time frames. This temporal consistency makes it challenging to directly constrain the length, width, height, and rotation angle for geometrically symmetric objects observed from different viewpoints, since the definition of box dimensions can vary and may not be globally consistent across views. Therefore, for dimension supervision, the height is directly constrained using an L1 loss L_h. For the length and width, we alternately compute the differences between the predictions and ground truth, then apply a softmin operation to form a robust loss term. The overall dimension loss is thus defined as:

$$L_{dim} = L_h + \sum_{k=1,2} w_k l_{wl}^{(k)}, \quad (3)$$

$$w_k = \frac{\exp(-l_{wl}^{(k)}/\tau)}{\sum_{m=1}^{2} \exp(-l_{wl}^{(m)}/\tau))}, \tau = 0.1 \quad (4)$$

where

$$\begin{aligned} \Pi_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \Pi_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \\ l_{wl} = \Pi_1 \cdot (\mathrm{L1}(\mathrm{w}_{pred}, \mathrm{w}_{GT}) + \mathrm{L1}(\mathrm{l}_{pred}, \mathrm{l}_{GT})) + \\ \Pi_2 \cdot (\mathrm{L1}(\mathrm{w}_{pred}, \mathrm{l}_{GT}) + \mathrm{L1}(\mathrm{l}_{pred}, \mathrm{w}_{GT})) \end{aligned} \quad (5)$$

For the camera pose, following StreamVGGT [44], we apply L1 loss L_{pose} separately to the translation, rotation, and field of view components. Furthermore, to address the sequential nature of the task, we design a consistency loss $L_{cons} = L_{spatial} + L_{temp}$ to enforce spatiotemporal consistency of the model's predictions in the global coordinate system across different time frames. Here, $L_{spatial}$ is a Chamfer Distance loss computed between the transformed 3D boxes and the ground truth in the global world coordinates, while L_{temp} is another Chamfer Distance loss that measures the discrepancy between per-frame predictions and their temporal average $\bar{B}_w$ across the sequence. Formally, these losses are defined as:

$$L_{spatial} = \sum_i \mathrm{chamfer}(B_w^i, B_w^{GT}), \quad (6)$$

$$L_{temp} = \frac{1}{T} \sum_i^T \mathrm{chamfer}(B_w^i, \bar{B}_w), \quad (7)$$

where $B_w^i = T_{c2w}^i \cdot B_c^i$ represents the 3D boxes transformed into the global coordinate system using the predicted box

B_c^i and camera pose T_{c2w}^i for each frame in the sequence. $\bar{B}_w$ denotes the temporal average of these transformed 3D boxes. By leveraging these two losses, our model effectively ensures the consistency of its predictions in both spatial and temporal dimensions.

The total loss function is composed as follows:

$$L = L_{depth} + L_{cam} + L_{det} + L_{pose} + L_{cons}. \quad (8)$$

Implementation Details. We employ ViT-L DINOv2 [22] and ViT-H SAM [17] as image encoders. We employ pretrained prompt encoder and 2D feature aggregator from DetAny3D [41]. The encoders and feature aggregator are frozen during training. The model is trained for 200 epochs taking 2 weeks. We leverage AdamW optimizer with initial learning rate of 0.0001, following cosine annealing policy. The image input is resized and padded to the resolution of 448. The maximum sequence length in training is set to 10.

5. Experiments

5.1. Setups

DA4D Benchmark. The proposed DA4D benchmark is a unified 4D detection dataset that integrates 12 datasets covering tasks such as 4D detection, depth estimation, and reconstruction. It builds upon the Omni3D [5] dataset, which includes ArkitScenes [3], Hypersim [27], KITTI [12], nuScenes [6], Objectron [1], and SUNRGBD [30], and adds sequential datasets including Replica [31], MP3D [7], HM3D [26], HSSD [16], Gibson [38], and Scannet [9], enriching resources for comprehensive 4D perception research. Sequences are provided with depth maps, intrinsics, camera poses, prompts, and 3D b-boxes. The dataset includes 170k sequences with multiple frames for 4D task settings and 110k single frames to validate the single frame capacity. All licenses are considered according to policies.

Baselines. We compare following three types of methods:

- **Monocular 3D Detection.** We employ DetAny3D [41], OV-Mono3D [40], Cube R-CNN [5], ImVoxelNet [28] as single frame 3D detection baselines. DetAny3D [41] is a recently open-sourced open-set 3D detector, reaching state-of-the-art performance on hybrid datasets. We train baselines with our DA4D datasets in single frame manner and aligned training strategy.
- **Video-based 4D Detection.** We employ ConceptGraph* [13], which projects frame-by-frame 2D detection results into 3D space using depth maps and performs object-level fusion to predict 3D bounding boxes. To ensure a fair comparison, we use depth predictions from [24], consistent with both [41] and our own method. Kinematic3D* [4] is a close-vocabulary 4D detection approach specifically designed for autonomous driving. For fairness, we adapt its original rotation decomposition strategy to match the rotation supervision used in other baselines.
- **Point-Cloud-Based 3D Detection.** We conduct comparison with SpatialLM [21], a structured scene understanding approach powered by large language models.

Metrics. We employ the widely used Average Precision (AP) for 3D b-boxes evaluation in each frame, using IoU3D thresholds range from $\tau \in [0.05, 0.10, \ldots, 0.50]$. To evaluate the cross-frame b-box consistency, we transfer predicted 3D b-boxes of each object across frames $B_{1,\ldots,t}$ into world coordinates $B'_{1,\ldots,t}$ using the camera poses. We then compute the average b-box $\bar{B}$ for each object in the world coordinates. Subsequently, we calculate the Chamfer Distance between eight vertices and the center of the b-box in each frame and the average, denoted as

$$\begin{aligned} \mathrm{Var_v} &= 1/t \sum \{\mathrm{chamfer}(B'_i, \bar{B})\}_{1,\ldots,t}, \\ \mathrm{Var_c} &= 1/t \sum \{||\mathrm{mean}(B'_i) - \mathrm{mean}(\bar{B})||_2\}_{1,\ldots,t}. \end{aligned} \quad (9)$$

Since not all methods predict camera poses, we employ the ground-truth camera poses for transformation to ensure a fair comparison. When compared with ConceptGraph* [13] and SpatialLM [21], we employ F1 scores under IoU of 0.25 and 0.5 following [21] instead of mAP, as suggested in [2]. Because SpatialLM [21] does not produce confidence score and ConceptGraph* [13] maps predictions from various frames, which is hard to evaluate cross-frame confidence. Cross-frame variance is also not suitable to evaluate b-boxes predicted from point cloud.

5.2. Main Results

As shown in Table 1, we evaluate several approaches on the proposed DA4D dataset, including monocular 3D detection, video-based 4D detection, and our end-to-end 4D detection method. All methods are provided with the same globally-labeled 3D bounding boxes and trained with strategies kept as consistent as possible. The results demonstrate that our method achieves competitive 3D detection performance compared to state-of-the-art (SOTA) approaches on individual frames. Furthermore, in terms of inter-frame consistency, DetAny4D reduces temporal variance by 10% to 30% compared to leading single-frame 3D detectors and exhibits less jitter than previous video-based 4D detection methods. Figure 5 includes qualitative comparison with other methods, showing better temporal consistency of 3D b-boxes across consecutive frames. These findings confirm that our architectural design and training strategy effectively and efficiently enhance spatiotemporal consistency.

5.3. Further Comparison

In Table 2, we compare with SpatialLM [21] and ConceptGraphs [13], two open-set global b-box prediction approaches. While our DetAny4D performs end-to-end 4D

Methods	Approach	Replica			MP3D			HM3D			Full DA4D		
		AP$^{\uparrow}_{3D}$	Var$^{\downarrow}_{v}$	Var$^{\downarrow}_{c}$	AP$^{\uparrow}_{3D}$	Var$^{\downarrow}_{v}$	Var$^{\downarrow}_{c}$	AP$^{\uparrow}_{3D}$	Var$^{\downarrow}_{v}$	Var$^{\downarrow}_{c}$	AP$^{\uparrow}_{3D}$	Var$^{\downarrow}_{v}$	Var$^{\downarrow}_{c}$
ImVoxelNet [28]	Monocular 3D Detection	10.03	1.24	1.36	12.99	1.80	1.79	11.32	1.33	1.28	11.98	1.50	1.43
Cube R-CNN [5]		24.49	0.82	0.86	20.90	1.71	1.68	21.84	0.94	0.89	21.76	1.27	1.20
OV Mono3D [40]		23.81	0.97	0.93	22.79	1.66	1.59	25.38	0.96	0.91	24.39	1.26	1.23
DetAny3D [41]		$\underline{27.25}$	0.62	0.52	**25.14**	1.28	1.23	**26.80**	0.83	0.70	$\underline{27.16}$	0.99	0.90
Kinematic3D* [4]	Video-Based 4D Detection	26.82	$\underline{0.58}$	$\underline{0.49}$	24.83	$\underline{1.03}$	$\underline{0.97}$	25.34	$\underline{0.74}$	$\underline{0.66}$	25.46	$\underline{0.85}$	$\underline{0.78}$
DetAny4D(ours)		**28.02**	**0.56**	**0.47**	$\underline{24.98}$	**0.90**	**0.86**	$\underline{26.79}$	**0.62**	**0.53**	**27.48**	**0.70**	**0.64**

Table 1. 4D detection evaluations on the DA4D dataset, comapred with 3D detection methods and multi-stage 4D detection methods. **Bold** and underlined indicates the best and second-best results. Results on three sub-datasets and full DA4D are evaluated. AP$_{3D}$ reports Average Precision of predicted 3D b-boxes evaluated on each frame compared with GT b-boxes. Var$_v$ and Var$_c$ report the temporal variance of b-box centers and vertices belonging to the same instance among a sequence as introduced in Section 5.1. Kinematic3D* indicates realization with adapted rotation supervision for fairness (Section 5.1).

Methods	Input	Strategy	F1$^{\uparrow}_{IoU@0.25}$	F1$^{\uparrow}_{IoU@0.5}$
SpatialLM [21]	Pre-scanned Point Cloud	End-to-End	27.1	12.8
ConceptGraph* [13]	RGB-D Seq.	Multi-Stage	45.6	41.9
DetAny4D(ours)	RGB Seq.	End-to-End	**49.7**	**45.5**

Table 2. 4D detection comparison with point-cloud-based and multi-stage method. SpatialLM employs pre-scanned point cloud and can only predict axis-aligned b-box, failing to perform well compared with GT. ConceptGraph* utilizes predicted depth for fairness (Section 5.1), which relies on RGB-D sequence and multi-stage inference.

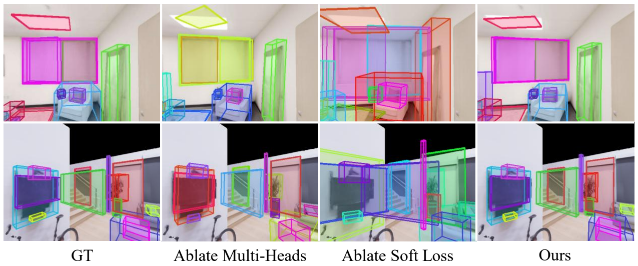


Figure 6. Visualize effect of ablated designs. Columns from left to right show the GT, prediction from multi-heads-ablated realization, soft-loss-ablated realization, and our original approach.

detection on RGB sequence, SpatialLM leverages pre-scanned global point-cloud for and ConceptGraphs relies on RGB-D sequence to perform multi-stage inference. SpatialLM works on the structured-scene assumption and predicts axis-aligned b-boxes, failing to perform well compared to the original GT. Results show that DetAny4D shows competitive performance against point-cloud based and multi-stage methods.

5.4. Ablation Studies

Ablated	Causal	Dim.	Vertices	Depth & Cam.	Pose & Cons.	AP$^{\uparrow}_{3D}$	Var$^{\downarrow}_{v}$	Var$^{\downarrow}_{c}$
Causal attn.	-	-	-	-	-	26.78	0.95	1.01
Soft loss	√	-	-	-	-	26.84	0.93	0.98
	√	√	-	-	-	26.88	0.91	0.96
Multi-heads	√	√	√	-	-	27.15	0.60	0.51
	√	√	√	√	-	27.28	0.59	0.50
Ours	√	√	√	√	√	**27.29**	**0.54**	**0.48**

Table 3. Ablation study of DetAny4D. Impact of each design on the AP$_{3D}$ and Var$_v$ is shown. Each component is progressively added to the origin model. The ablations are conducted on 10% of the full training dataset to save computational cost.

In Table 3, we ablate the key components of proposed DetAny4D, showing the performance degradation from DetAny4D to the original 3D detection model, which employs RGB sequence as input and performs single frame detection, ignoring temporal information. We also show qualitative comparison in Figure 6, which shows that without the multi-heads design, the center and rotation pose of the 3D b-box converge poorly, while the absence of the soft loss strategy leads to significant errors in the dimensions.

- **Causal attn.** By employing causal attention (Section 4.2), the model is enabled to manage temporal information and perform sequence-wise training and inference.
- **Soft loss.** The soft loss design described in 4.4 leads the model to learn the globally-annotated 3D b-boxes different from single frame 3D b-box annotation pattern. The order independent dimension and vertices chamfer distance loss make the training process a better convergence.
- **Multi-heads.** The multi-heads design described in Section 4.3 equips the model with geometric and temporal transformation awareness. The depth and camera parameter prediction heads help mitigate gaps in multiple datasets training and provide dense supervision of the scene geometry. The camera pose head and consistency head model the view transformation and relative pose relationship of observed objects and camera view.

6. Conclusion

We propose DetAny4D, the first open-set end-to-end 4D detection benchmark, which predicts spatiotemporally aligned 3D b-boxes across frames. Our solution features a large-scale 4D object detection dataset with unified data generation pipeline, a novel framework with multi-task heads for coherent spatiotemporal reasoning, and tailored training strategy with hybrid loss functions. Extensive evaluations demonstrate that DetAny4D achieves competitive 3D detection accuracy with significantly temporal stability.

References

[1] Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. *CVPR*, 2021. 7, 1

[2] Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins, Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan Frost, Luke Holland, Campbell Orme, Jakob Engel, Edward Miller, Richard Newcombe, and Vasileios Balntas. Scenescript: Reconstructing scenes with an autoregressive structured language model. In *European Conference on Computer Vision (ECCV)*, 2024. 7

[3] 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 (Round 1)*, 2021. 7, 1

[4] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele. Kinematic 3d object detection in monocular video. In *European Conference on Computer Vision*, pages 135–152. Springer, 2020. 2, 7, 8

[5] Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. Omni3D: A large benchmark and model for 3D object detection in the wild. In *CVPR*, Vancouver, Canada, 2023. IEEE. 1, 2, 6, 7, 8

[6] 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. 7, 1

[7] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 7, 1

[8] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2147–2156, 2016. 2

[9] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 7, 1

[10] Saumitro Dasgupta, Kuan Fang, Kevin Chen, and Silvio Savarese. Delay: Robust spatial layout estimation for cluttered indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 616–624, 2016. 2

[11] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014. 5

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 7, 1

[13] Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5021–5028. IEEE, 2024. 1, 2, 3, 7, 8

[14] Jiawei He, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. 3d video object detection with learnable object-centric global optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5106–5115, 2023. 2, 3

[15] 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. *Advances in Neural Information Processing Systems*, 31, 2018. 2

[16] Mukul Khanna*, Yongsen Mao*, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X. Chang, and Manolis Savva. Habitat Synthetic Scenes Dataset (HSSD-200): An Analysis of 3D Scene Scale and Realism Tradeoffs for ObjectGoal Navigation. *arXiv preprint*, 2023. 7, 1

[17] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 2, 5, 7

[18] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022. 2

[19] Zhuoling Li, Xiaogang Xu, Ser-Nam Lim, and Hengshuang Zhao. Towards unified 3d object detection via algorithm and data unification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 2

[20] Zechen Liu, Zizhang Wu, and Roland Tóth. Smoke: Single-stage monocular 3d object detection via keypoint estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 996–997, 2020. 2

[21] Yongsen Mao, Junhao Zhong, Chuan Fang, Jia Zheng, Rui Tang, Hao Zhu, Ping Tan, and Zihan Zhou. Spatiallm: Training large language models for structured indoor modeling. In *Advances in Neural Information Processing Systems*, 2025. 1, 3, 7, 8

[22] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 5, 7

[23] Jinhyung Park, Xinshuo Weng, Yunze Man, and Kris Kitani. Multi-modality task cascade for 3d object detection. *arXiv preprint arXiv:2107.04013*, 2021. 3

[24] Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. UniDepthV2: Universal monocular metric depth estimation made simpler, 2025. 5, 6, 7

[25] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019. 3

[26] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. 7, 1

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

[28] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. 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*, pages 2397–2406, 2022. 7, 8

[29] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Tr3d: Towards real-time indoor 3d object detection. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 281–285. IEEE, 2023. 1, 2

[30] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, 2015. 7, 1

[31] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 7, 1

[32] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2, 3

[33] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. `https://github.com/open-mmlab/OpenPCDet`, 2020. 1, 6

[34] Futian Laboratory Shenzhen Tencent Robotics X Team. Tairos: An embodied ai platform for robotics applications, 2025. 1, 3

[35] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5038–5047, 2017. 5

[36] Haiyang Wang, Shaoshuai Shi, Ze Yang, Rongyao Fang, Qi Qian, Hongsheng Li, Bernt Schiele, and Liwei Wang. Rbgnet: Ray-based grouping for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1110–1119, 2022. 3

[37] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 913–922, 2021. 2

[38] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: real-world perception for embodied agents. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018. 7, 1

[39] Jin Yao, Hao Gu, Xuweiyi Chen, Jiayun Wang, and Zezhou Cheng. Open vocabulary monocular 3d object detection. *arXiv preprint arXiv:2411.16833*, 2024. 6

[40] Jin Yao, Hao Gu, Xuweiyi Chen, Jiayun Wang, and Zezhou Cheng. Open vocabulary monocular 3d object detection. *arXiv preprint arXiv:2411.16833*, 2024. 2, 7, 8

[41] Hanxue Zhang, Haoran Jiang, Qingsong Yao, Yanan Sun, Renrui Zhang, Hao Zhao, Hongyang Li, Hongzi Zhu, and Zetong Yang. Detect anything 3d in the wild. *arXiv preprint arXiv:2504.07958*, 2025. 1, 2, 5, 7, 8

[42] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3dnet: 3d object detection using hybrid geometric primitives. In *European conference on computer vision*, pages 311–329. Springer, 2020. 3

[43] 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 *Proceedings of the Asian Conference on Computer Vision*, 2020. 2

[44] Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4d visual geometry transformer. *arXiv preprint arXiv:2507.11539*, 2025. 6

DetAny4D: Detect Anything 4D Temporally in a Streaming RGB Video

Supplementary Material

7. Dataset Composition

DA4D is a hybrid 4D detection dataset which includes 12 sub-datasets. The supporting tasks include sequence 4D detection, monocular 3D detection, depth estimation, and reconstruction tasks. DA4D is built on six datasets in Omni3D [5] (ArkitScenes [3], Hypersim [27], KITTI [12], nuScenes [6], Objectron [1], and SUNRGBD [30]) and expanded with a sequence dimension. Six more datasets are introduced for sequence-wise tasks, including Replica [31], MP3D [7], HM3D [26], HSSD [16], Gibson [38], and Scannet [9]. The format is standardized similar to the Omni3D structure with additional sequence information. Each sequence compromises a list of frames. Each frame includes monocular RGB image, camera intrinsics, camera pose, depth map, and object information. The object information includes 3D b-box attribute ($[x, y, z, w, h, l, yaw]$), rotation pose, 2D b-box prompt, category, instance ID, and score.

Dataset Composition. The dataset compromises original single frame 3D detection data as 3D detection capacity validation and multi-frame sequences data for 4D tasks. As shown in Figrue 7, DA4D consists of multi-frame sequences and Omni3D 3D detection data considered as sequences with length of 1.

Dataset Split. Sequences from Omni3D [5] follows the splitting strategy in prior works [5]. For newly compromised datasets to form multi-frame sequences, we split the scenes into the training and validation set, and separately record sequences from the selected scenarios, ensuring all scenes in the validation set have not been visited.

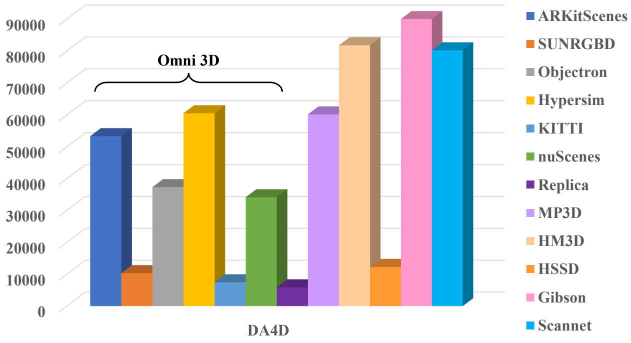


Figure 7. Visualization of the DA4D dataset composition.

8. Training Strategy Details

Here we illustrate the sequence crop and object padding strategy in Section 4.4 with more details. As shown in Figure 8, we crop each clip under a fixed maximum length. For the cropped sequence, we count the total objects number in each sequence and pad the objects in each frame to this number. During training, this strategy ensures each frame has the same object query dimension, and predictions generated with the padded queries will be masked which do not contribute to the loss. During inference, the padded queries enable the model to manage newly appeared objects during forecasting, where new objects are registered to the padded queries and maintained if the prediction result has a high score and differs against objects in the query memory list.

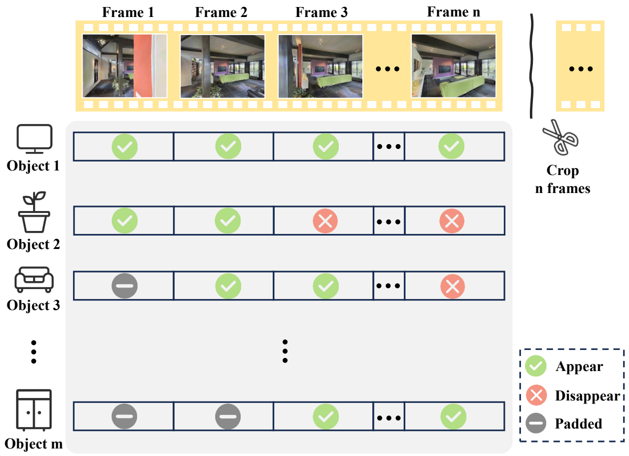


Figure 8. Visualization of the sequence crop and object padding strategy. The object query list maintains the objects and padding status. As frames in the sequence forecast, object status updates. The disappeared objects and padded objects do not contribute to the loss.

9. 3D B-box annotation Pattern

This section illustrates the differences in the b-box annotation pattern for 3D detection task and 4D detection task. As shown in Figure 9, when observing an object from different views, 3D detection annotation pattern turns to consider each view as a singular observation and annotate the box referring only to the current view. On the contrary, for 4D detection, each object is registered with a b-box in the global coordinates and this constant b-box is projected to various views ensuring consistency globally.

Given the annotation formats of the 4D detection task and our goal to leverage powerful prior knowledge from 3D detection, specifically by utilizing pre-trained 3D detection models, we designed a specialized loss function (Section 4.4) to constrain the predictions to align with 4D task

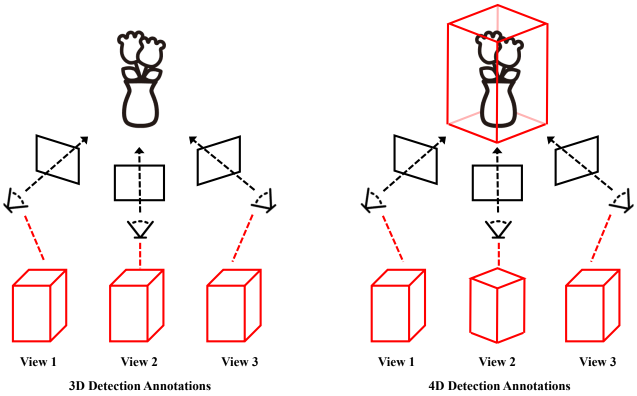


Figure 9. Visualization of the annotation pattern differences. 3D detection task turns to annotate each object reffering only to the current view. 4D detection requires each view predicts objects in the global coordinates.

annotations. Our composite loss function, detailed in Section 4.4, incorporates constraints on the center, dimensions, and rotation angle of the 3D bounding boxes. Figure 10 shows the supervision of the b-box. In contrast to the rigid constraints traditionally used in 3D detection, we employ a softened loss for the dimensions and rotation. This design is motivated by the potential misalignment between the outputs of 3D pre-trained models and the 4D annotations, as shown in Figure 11. Directly applying hard constraints can lead to slow and difficult convergence, whereas the softened loss facilitates a more effective and stable alignment of the predictions to the 4D ground truth.

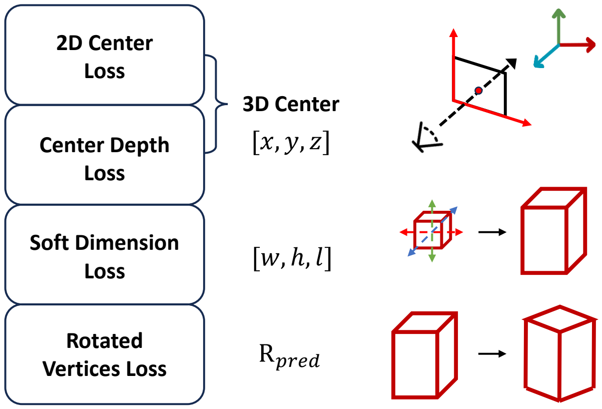


Figure 10. Visualization of the b-box supervision. The 3D bounding box is constrained sequentially by its center, dimensions, and rotation angle.

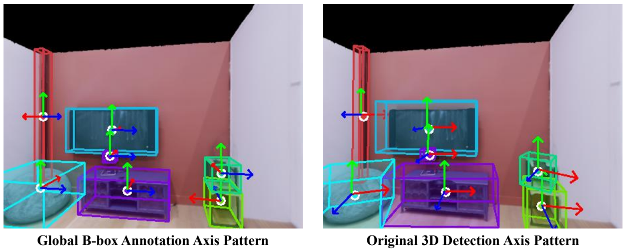

Global B-box Annotation Axis Pattern
Original 3D Detection Axis Pattern

Figure 11. Visualization of the axis pattern comparison between global annotations and 3D prediction results. Global annotated b-box dimensions and rotation differs from the predicted results of pre-trained 3D detection model.

Datasets	Categories Open-Set / All	Objects Num Open-Set / All	**Open-Set**		**None-Open-Set**	
			$\mathrm{AP}_{3\mathrm{D}}^{\uparrow}$	$\mathrm{F1}^{\uparrow}$	$\mathrm{AP}_{3\mathrm{D}}^{\uparrow}$	$\mathrm{F1}^{\uparrow}$
Replica	18 / 62	0.3k / 16k	27.9	46.8	28.0	47.1
MP3D	20 / 330	3k / 98k	24.7	41.9	25.0	43.2
HM3D	75 / 698	11k / 344k	27.2	44.4	26.7	43.7

Table 4. Evaluation of the open-set performance. We separately evaluate the AP and F1 score on three sub-datasets and also report the ratio of the categories and numbers of open-set objects.

10. Open-Set Validation

To provide a detailed assessment of our model's open-set detection capabilities, we separately evaluated the 4D detection metrics on objects from categories that were within the training set (seen categories) and those that were outside of it (unseen categories) in the validation set. Table 4 shows the AP and F1 score under threshold IoU@0.5 (the same metrics illustrated in Section 5.1) of open-set categories and none-open-set categories in the validation set. It can be seen that the model performance on open-set highly keeps align with the none-open-set performance.

11. More Results

We provide more visualization comparison to show the prior performance of our proposed DetAny4D on the spatiotemporal consistency validation.

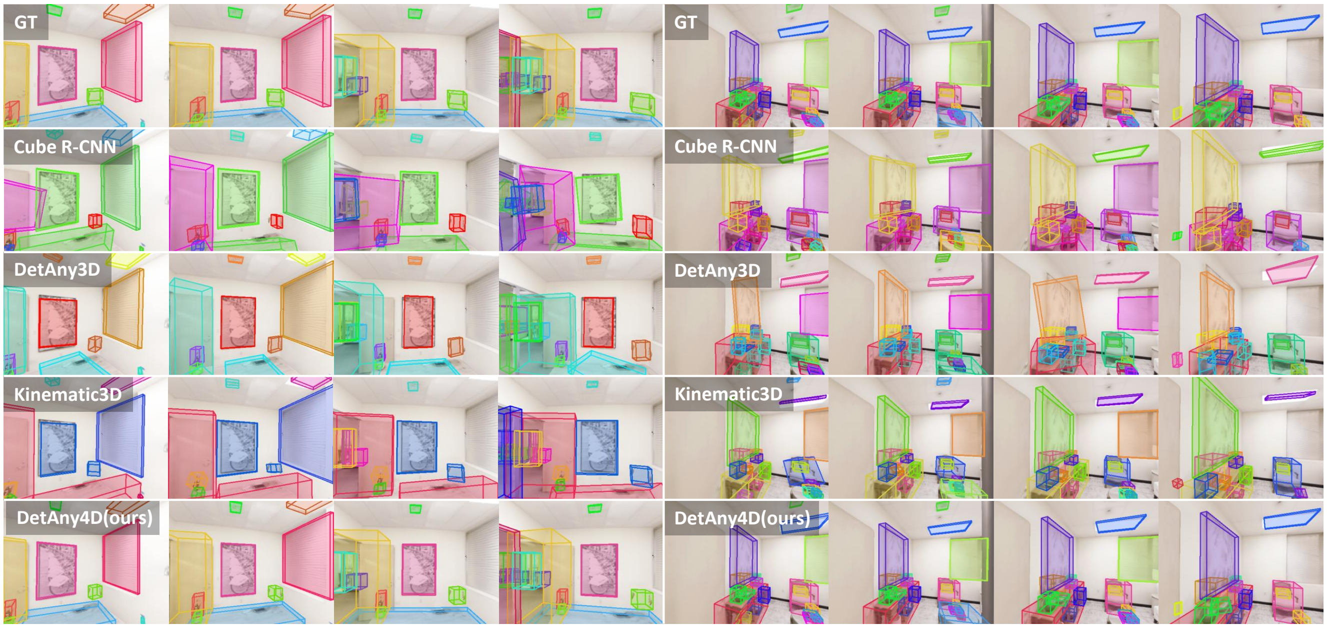


Figure 12. More qualitative comparison results.

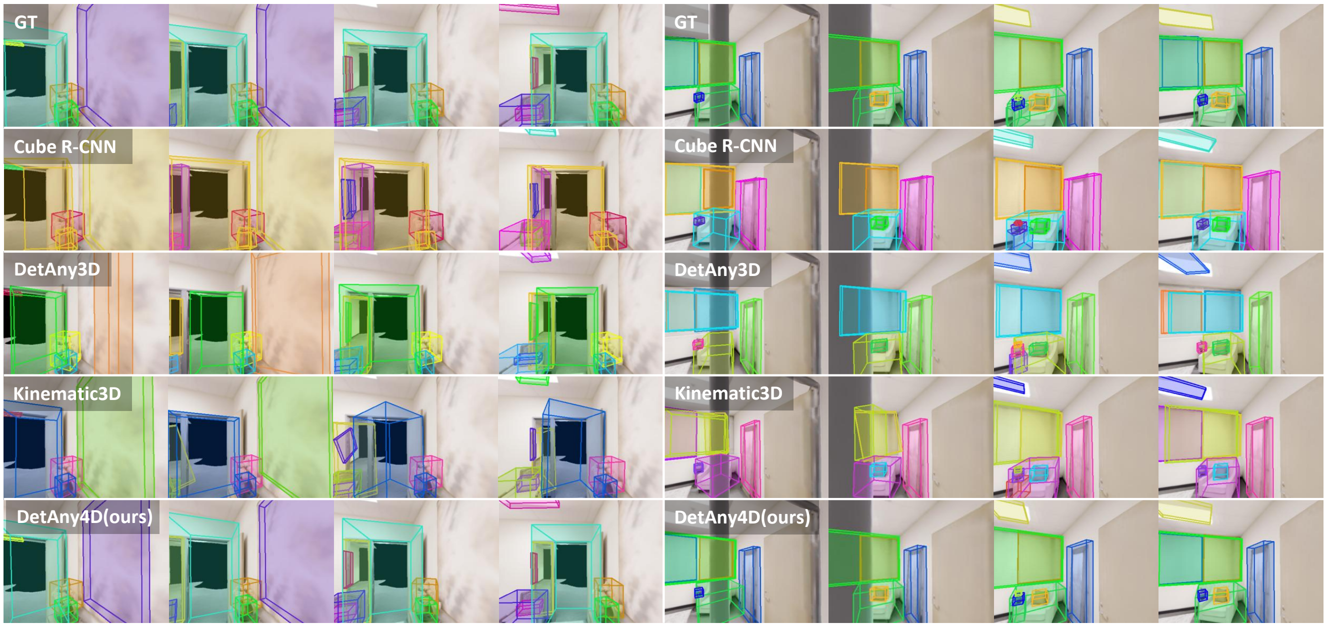


Figure 13. More qualitative comparison results.

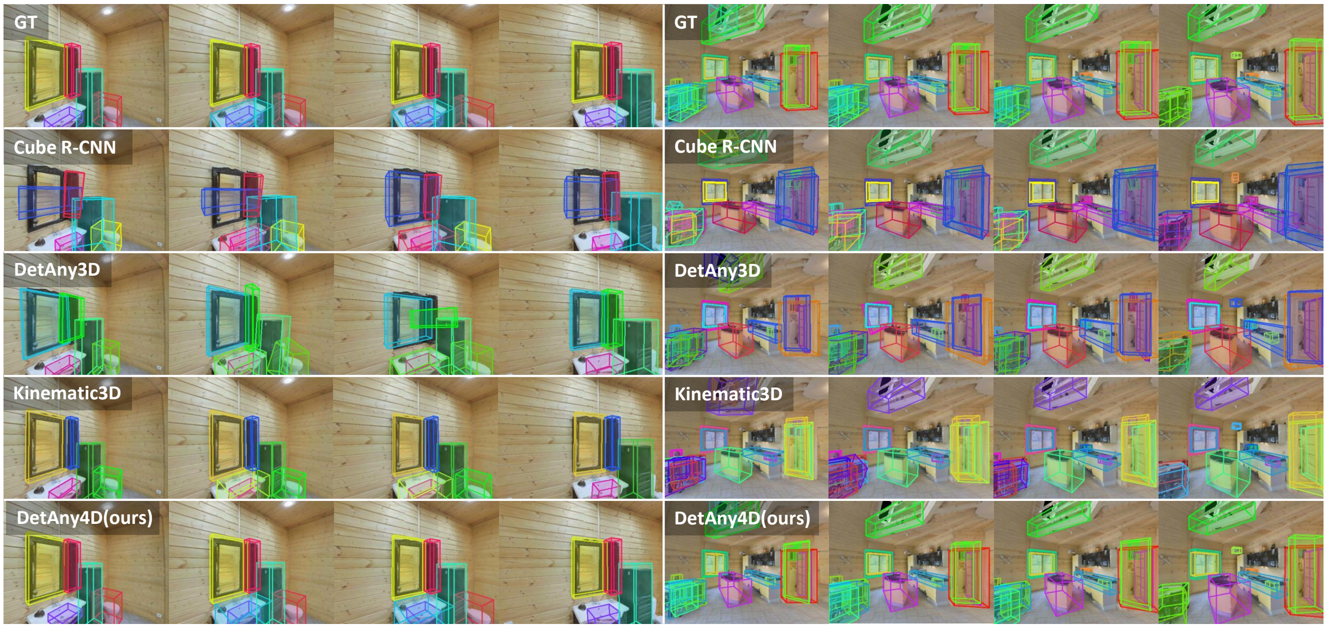


Figure 14. More qualitative comparison results.

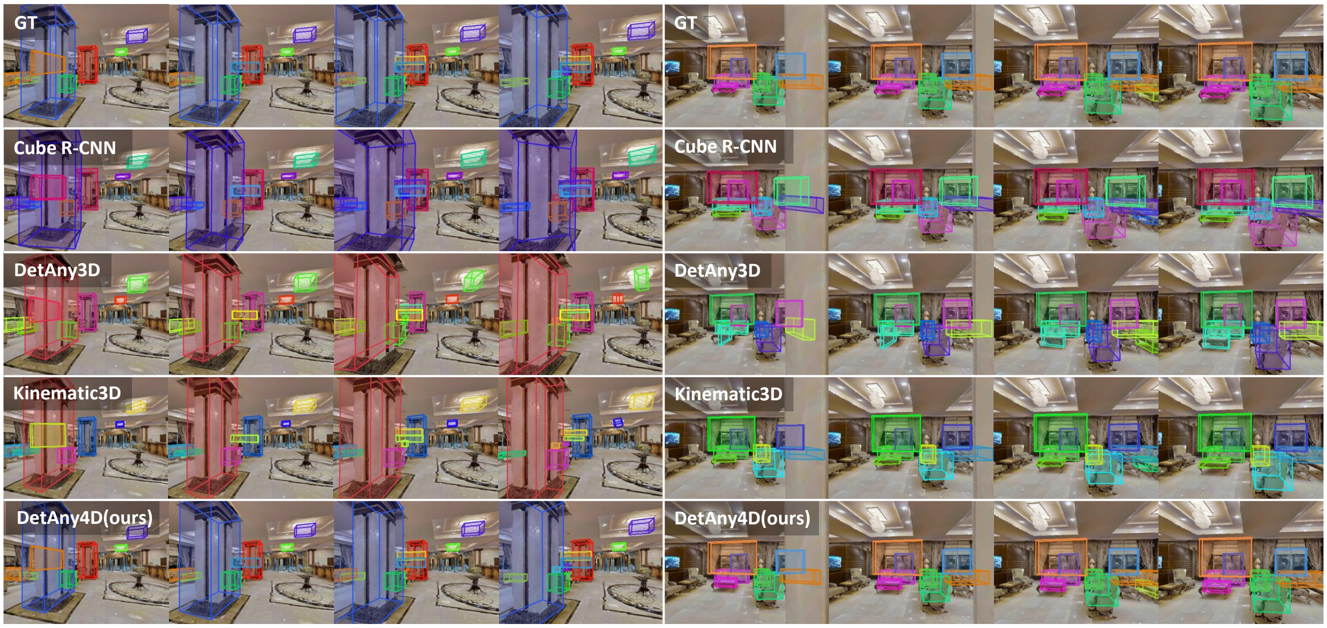


Figure 15. More qualitative comparison results.

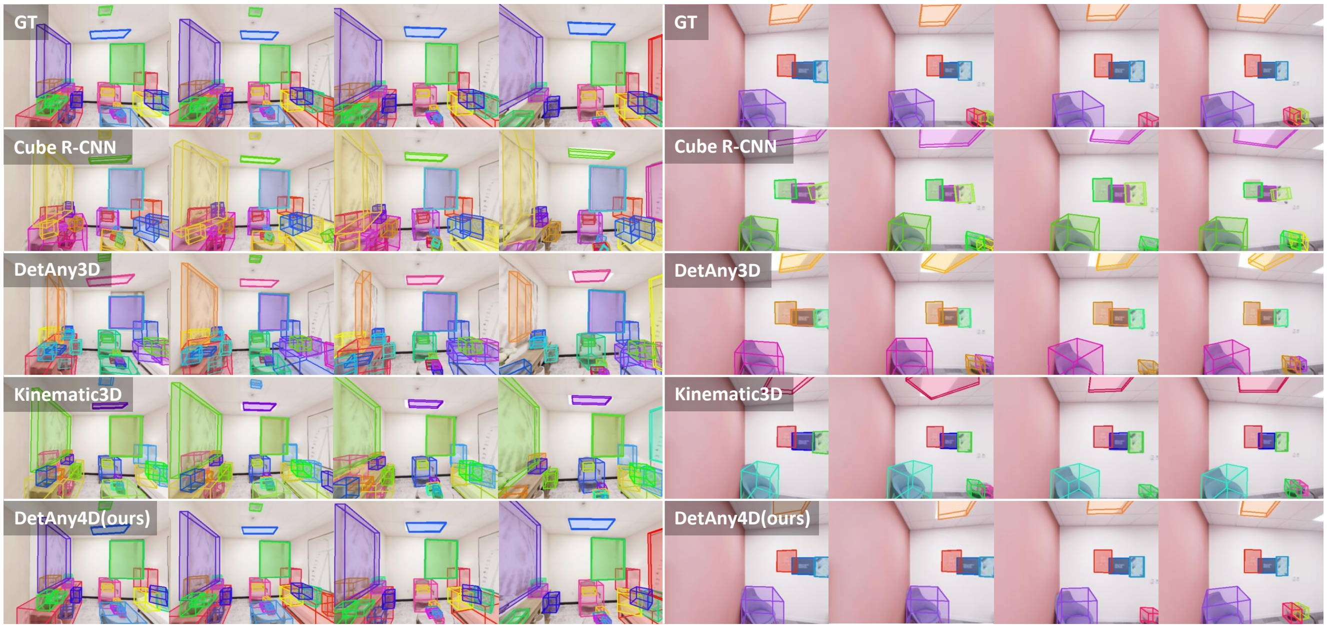


Figure 16. More qualitative comparison results.

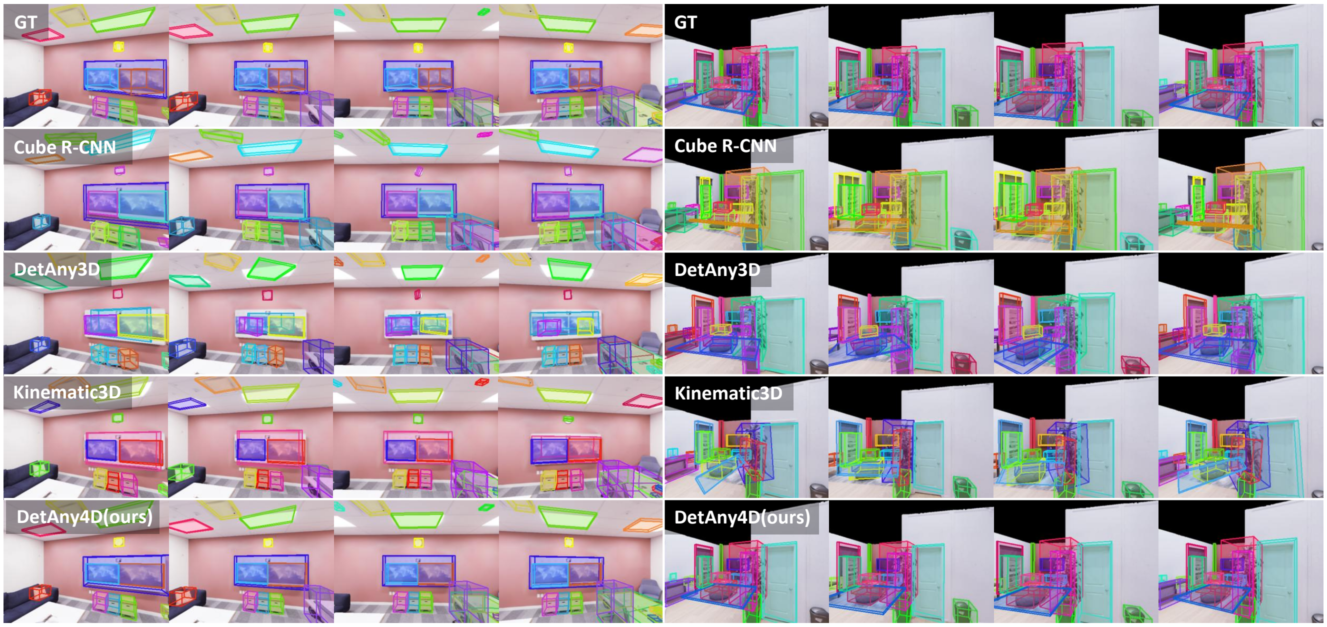


Figure 17. More qualitative comparison results.

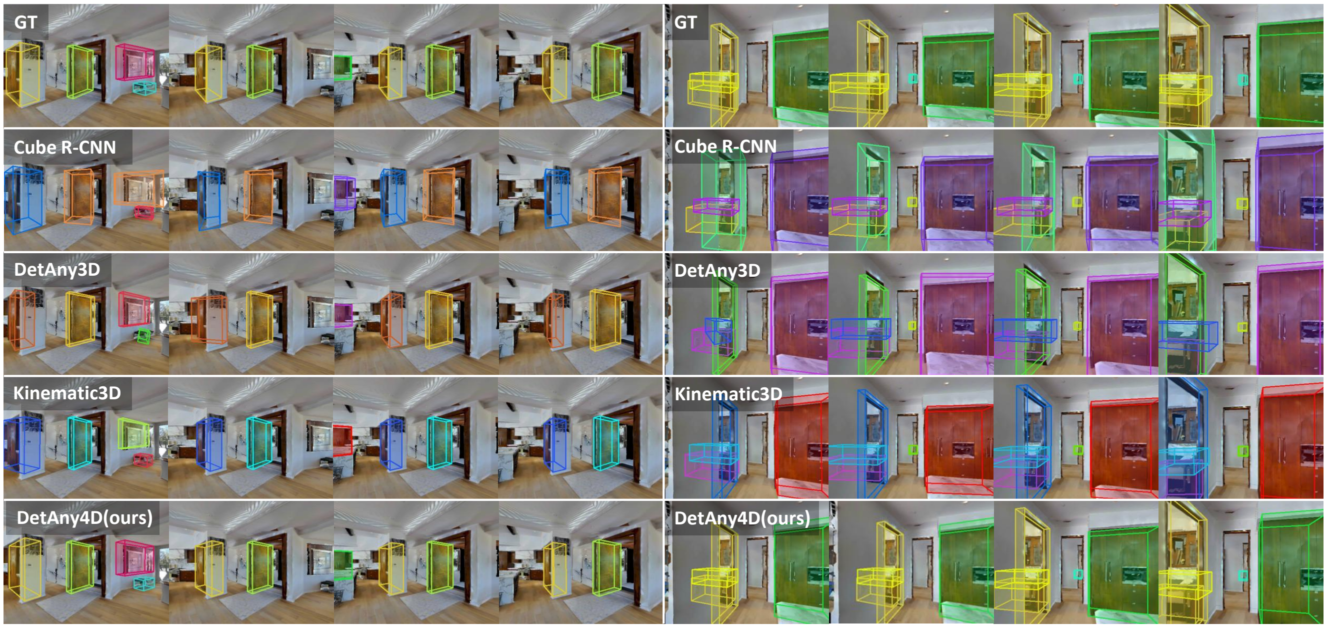


Figure 18. More qualitative comparison results.

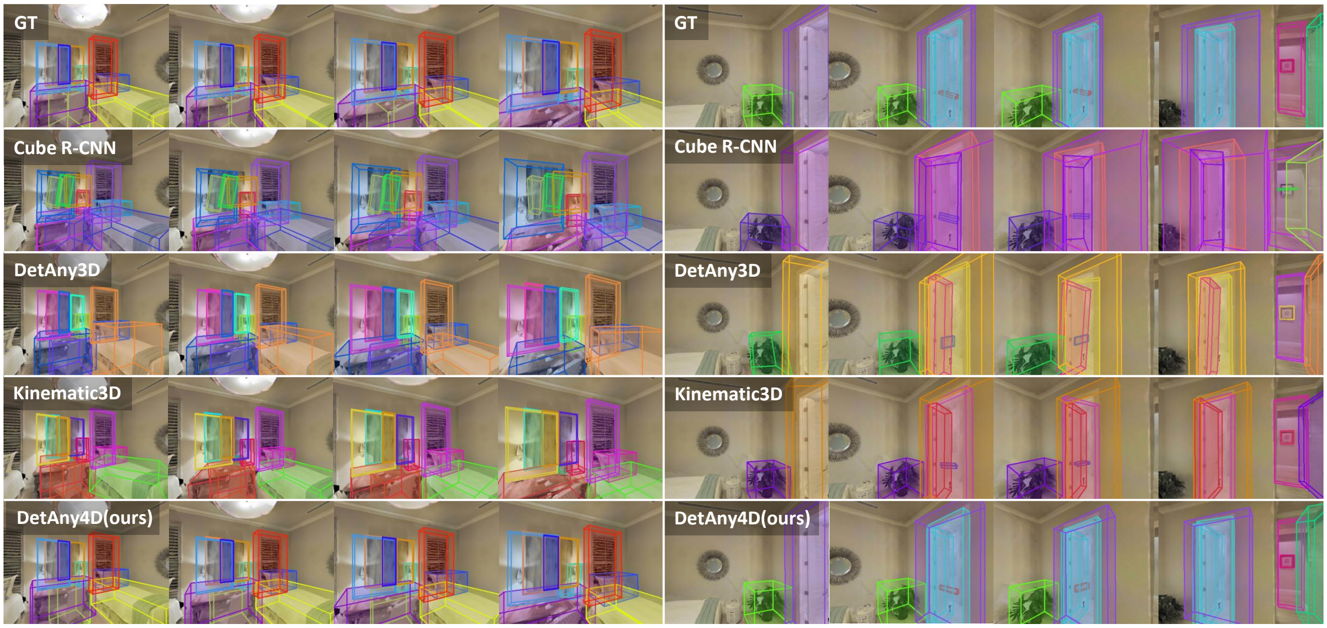


Figure 19. More qualitative comparison results.

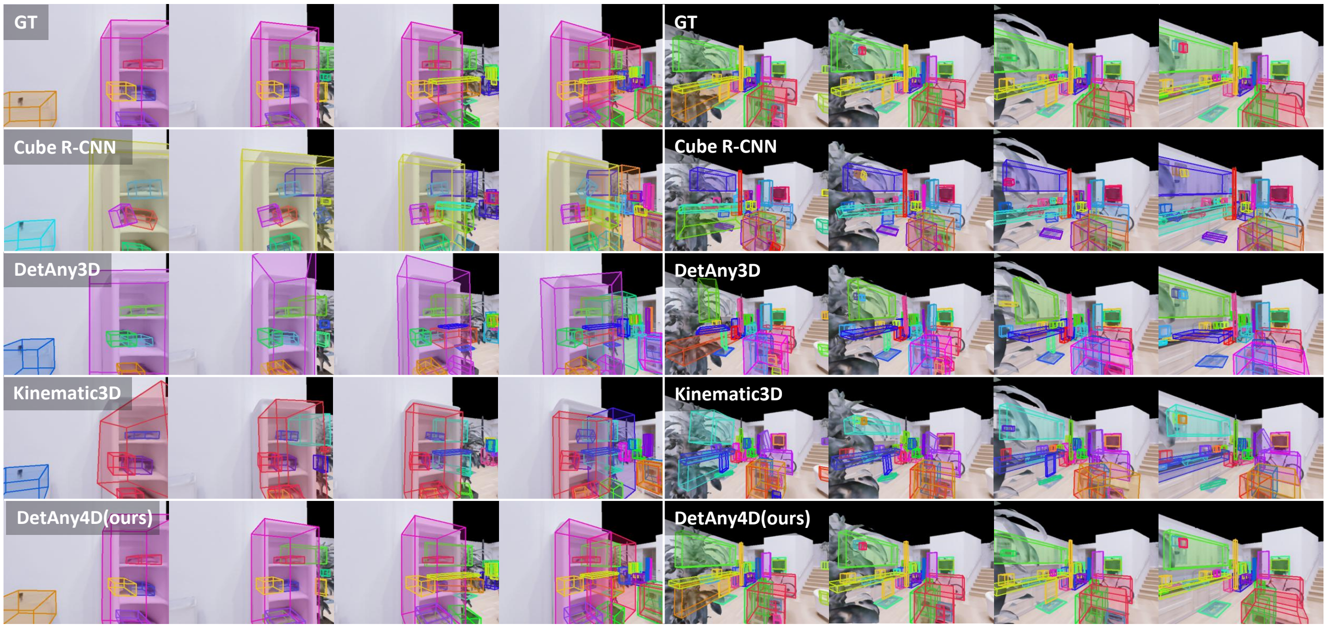


Figure 20. More qualitative comparison results.

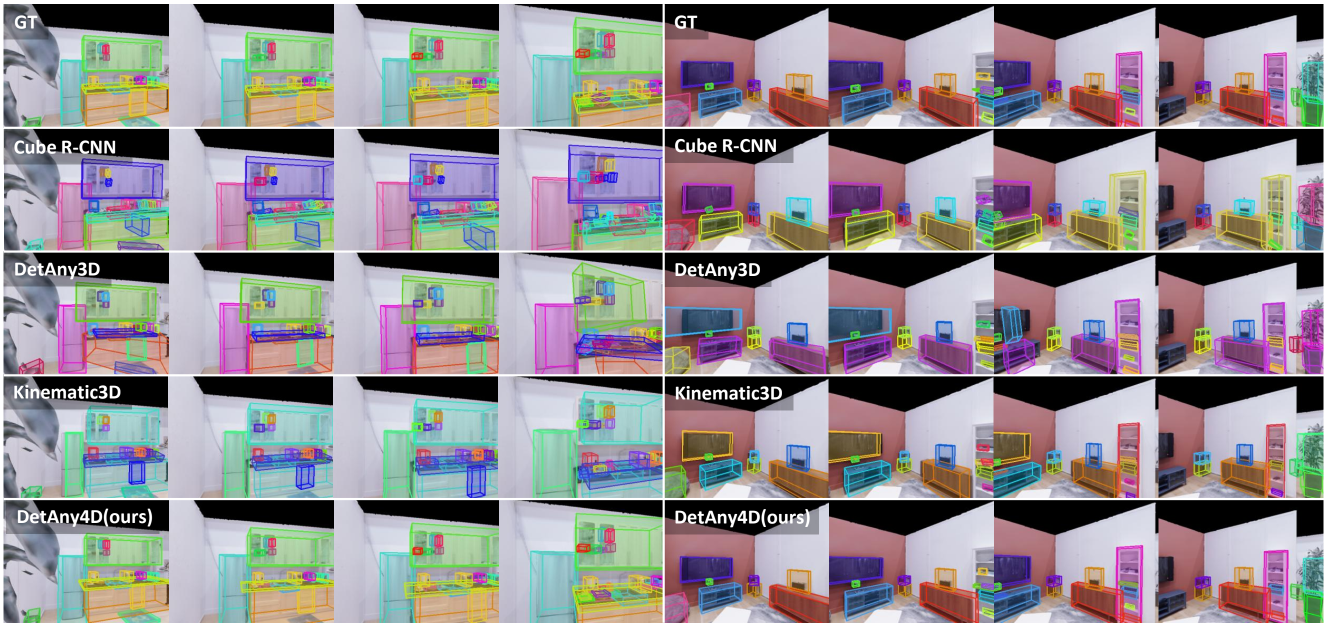


Figure 21. More qualitative comparison results.