3DiffTection: 3D Object Detection with Geometry-Aware Diffusion Features

Supplementary Material

A. Additional Details

A.1. Implementation Details.

The Omni3D-ARKitscenes dataset includes 51K training images, 7.6K test images and a total of 420K oriented 3D bounding boxes annotation. Omni3D-SUN-RGBD includes 5.2K training images, 5K test images and a total of 40K 3D bounding box annotations. Omni-Indoor is a combination of ARKitScenes, SUN-RGBD and HyperSim datasets. Our geometric ControlNet is trained at a resolution of 256 $\times$ 256, following the camera pose annotations provided by the ARKitScenes dataset [3]. The novel-view synthesis training is mainly developed based on ControlNet implementation from Diffusers [1]. We use the same training recipe as Diffusers. To report results at a resolution of 512 $\times$ 512, we directly input images with a resolution of 512 $\times$ 512 into the backbone. Note that different to both original ControlNet [59] and DIFT [48] that use text as input, we input empty text into the diffusion model thorough the experiments.

A.2. 3D-Detection Head and Objective.

3D detection head. We use the same 3D detection head as Cube-RCNN [5]. Specifically, Cube-RCNN extends Faster R-CNN with a cube head to predict 3D cuboids for detected 2D objects. The cube head predicts category-specific 3D estimations, represented by 13 parameters including (1) projected 3D center (u, v) on the image plane relative to the 2D Region of Interest (RoI); (2) object's center depth in meters, transformed from virtual depth (z); (3) log-normalized physical box dimensions in meters ($\bar{w}$, $\bar{h}$, $\bar{l}$); (4) Continuous 6D allocentric rotation ($p \in \mathcal{R}$); and (5) predicted 3D uncertainty μ. With parameterized by the output of the cube head, the 3D bounding boxes can be represented by

$$B_{3D}(u, v, z, \bar{w}, \bar{h}, \bar{l}, p) = R(p) \cdot d(\bar{w}, \bar{h}, \bar{l}) \cdot B_{\text{unit}} + X(u, v, z), \tag{9}$$

where R(p) is the rotation matrix and d is the 3D box dimensions, parameterized by $\bar{w}$, $\bar{h}$, $\bar{l}$. $X(u, v, z)$ is the bounding box center, represented by

$$X(u, v, z) = \left(\frac{z}{f_x}\right)\left(rx + u \cdot r_w - p_x, \frac{z}{f_y}\right)(ry + v \cdot r_h - p_y) \tag{10}$$

where: $[r_x, r_y, r_w, r_h]$ are the object's 2D bounding box. (f_x, f_y) are the known focal lengths of the camera. (p_x, p_y) represents the principal point of the camera. Given the representation of the 3D bounding box, our detection training objective is

$$L = L_{\text{RPN}} + L_{2D} + \sqrt{2} \cdot \exp(-\mu) \cdot L_{3D} + \mu, \tag{11}$$

where L_{RPN} and L_{2D} are commonly used in 2D object detection such as Faster-RCNN [36], here L_{3D} is given by

$$L(u, v)_{3D} = ||B_{3D}(u, v, z_{\text{gt}}, \bar{w}_{\text{gt}}, \bar{h}_{\text{gt}}, \bar{l}_{\text{gt}}, p_{\text{gt}}) - B^{3D}_{\text{gt}}||_1 \tag{12}$$

A.3. Semantic ControlNet Illustration

The block diagram of Semantic ControlNet is depicted in Fig. 6. Please refer to the main description of Semantic ControlNet in the method part.

B. Additional Experimental Results

B.1. Table of Label Efficiency

The label efficiency table is shown in Tab. 4. Please refer to the Experiment section of the main text for the analysis of label efficiency experiments.

B.2. Latency

We evaluate the latency of our proposed method on one RTX 3090 GPU. The latency comparison is shown in Tab. 5. We can see that our method is indeed slower than our baseline. It is our future work to improve the latency of our proposed method.

B.3. Detection Results on SUN-RGBD Common Classes

We also evaluate our method on the common SUN-RGBD 10 classes, as shown in Tab. 6, as following [5] and [32]. Experiments demonstrate that our proposed 3DiffTection significantly improves the previous method by a large margin.

B.4. Correspondences Evaluation

We quantitatively evaluate the correspondence results on ScanNet dataset [12], as following SuperGlue [41]. In particular, we directly use the ScanNet image data and ground truth provided by SuperGlue official library [2], and we select the keypoints estimated from the SuperGLue pretrain models. Then we use the keypoints to evaluate the correspondence accuracy for target images. The keypoint correspondences evaluation metric is precision and matching

[1] https://github.com/huggingface/diffusers/blob/main/examples/controlnet/train_controlnet.py

[2] https://github.com/magicleap/SuperGluePretrainedNetwork

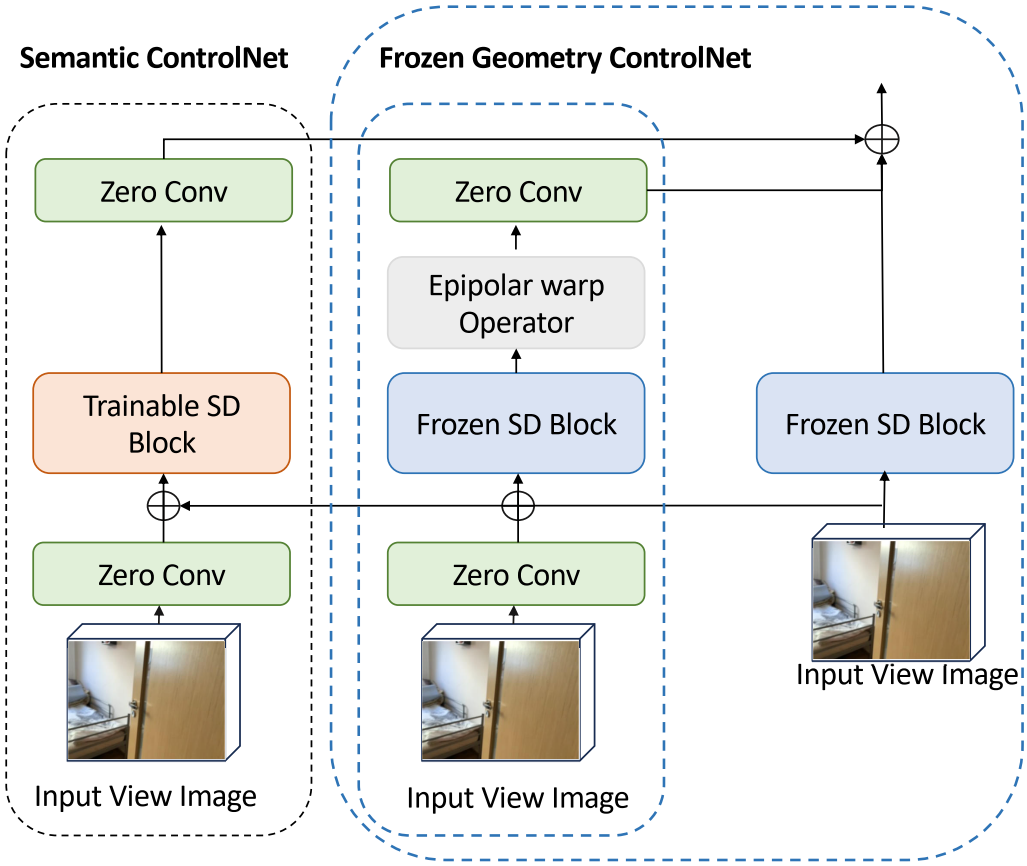


Figure 6. **Semantic ControlNet.** When tuning Semantic ControlNet, we freeze the pre-trained Geometric ControlNet and the original Stable Diffusion block. For both training and inference, we input the identity pose into the Geometric ControlNet by default.

Methods	**Backbone**	**Pre-training**	**Tuned Module.**	**100% data**	**50% data**	**10% data**
CubeRCNN	DLA34	ImageNet cls.	DLA34+3D Head	31.75	25.32	7.83
DreamTchr	ResNet50	SD distill	Res50+3D Head	33.20	26.61	8.45
DIFT-SD	StableDiff	LION5B gen.	3D Head	28.86	24.94	7.91
3DiffTection	StableDiff+Geo-Ctr	Aktsn nvs.	3D Head	30.16	27.36	14.77
3DiffTection	StableDiff+Geo-Ctr	Aktsn nvs.	Sem-Ctr+3D Head	39.22	35.48	17.11

Table 4. Label efficiency in terms of AP3D.

Method	Latency (s)
3DiffTection (w/o SemanticControlNet)	0.104
3DiffTection	0.133
3DiffTection (w/ 6 virtual view Ensemble)	0.401
Cube-RCNN-DLA34	0.018

Table 5. Latency comparison on one 3090Ti GPU.

score (MS), as shown in Tab. 7. We also evaluate the performance of pose estimation via AUC. Experiment results also demonstrate that our method can improve the pose estimation.

Moreover, we apply the 3DiffTection feature to SuperGlue pipeline by augmenting the features onto the Super-

Method	AP3D
Total3D [32]	27.7
ImVoxelNet [38]	30.6
Cube-RCNN	35.4
3DiffTection	38.8

Table 6. Comparison on common categories of SUN-RGBD dataset.

Methods	**AUC@5**	**AUC@10**	**AUC@20**	**Precision**	**MS**
DIFT-SD	5.61	6.15	14.73	16.95	4.57
3DiffTection	7.49	8.92	17.66	19.11	6.93

Table 7. Correspondence evaluation on ScanNet dataset.

SuperGLUE	SuperGLUE+DIFT	SuperGLUE+Ours
13.26	14.77	20.86

Table 8. AUC@5 camera pose estimation on ScanNet subset

Point features, we find that compared to previous method, our 3DiffTEction+SuperGlue can achieve superior performance in terms of pose estimation metric AUC@5.

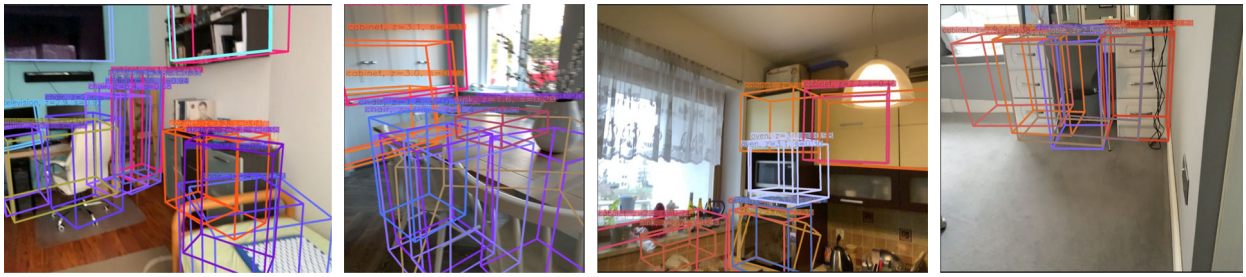
C. Additional Results Visualizations

Visualization of 3D bounding box. We show more 3D bounding box visualization results in Fig. 7 and Fig. 8, which are the results on the test sets of Omni3D-ARKitScenes and Omni3D-SUN-RGBD, respectively. We observe that our 3DiffTection can predict more accurate 3D bounding boxes compared to Cube-RCNN. More importantly, 3DiffTection does not fail even in very challenging cases, such as the second column of Fig. 7 and the first column of Fig. 8. They show that 3DiffTection is able to handle occlusions where the chairs are occluded by the tables.

Visualization of novel-view synthesis. We then provide more visualization results about novel-view synthesis, as shown in Fig. 9. We randomly chose the images that were never seen during the training of geometric ControlNet. To provide how the camera rotates, we present the warped images as well. Even though novel-view synthesis at the scene level is not our target task, it can be seen that our model can still generalize well under the challenging setting of synthesising novel views from one single image.

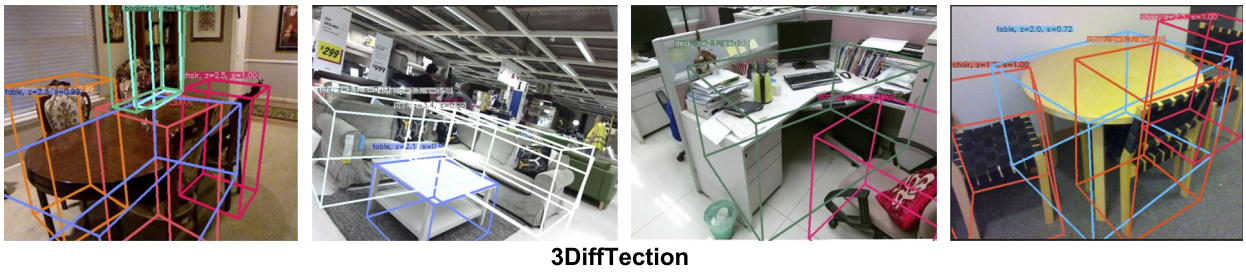
Visualization of 3D correspondences. We provide more visualization results about 3D correspondences, as shown in Fig. 10. We can observe that our method can find very accurate 3D correspondences in challenging cases. For instance, in the second example, the camera translation is large and this leads to drastically near/far movement and multi-scale correspondences. Our method can successfully find all the correspondences even if the camera near-far changes a lot.

3DiffTection

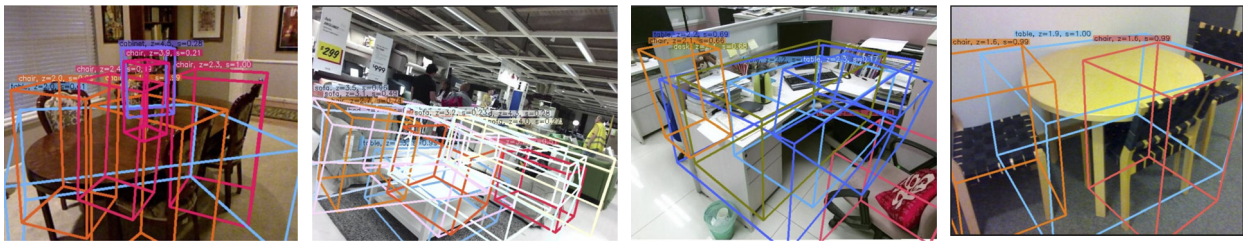


Cube-RCNN

Figure 7. Visualization of 3D bounding boxes on the Omni3D-ARKitScenes test set.



3DiffTection



Cube-RCNN

Figure 8. Visualization of 3D bounding boxes on the Omni3D-SUNRGB-D test set.

Figure 9. Visualization of novel-view synthesis. We rotate the camera by 15 deg anchoring to different axises. The warp image can be used to indicate the camera rotated directions.

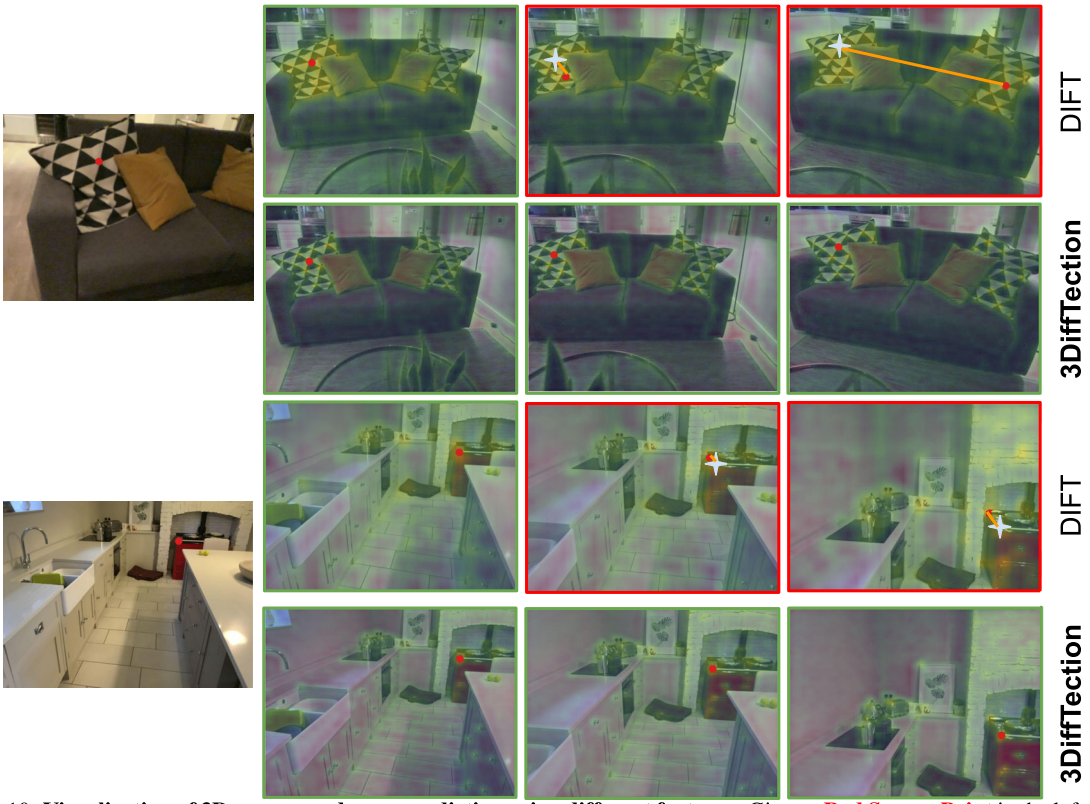


Figure 10. **Visualization of 3D correspondences prediction using different features**. Given a **Red Source Point** in the leftmost reference image, we predict the corresponding points in the images from different camera views on the right (**Red Dot**). The ground truth points are marked by **Blue Stars**. Our method, 3DiffTection, is able to identify precise correspondences in challenging scenes with repetitive visual patterns. The orange line measures the error of the prediction and ground truth points.