SUPPLEMENTARY FOR

Pixel-Aligned Recurrent Queries for Multi-View 3D Object Detection

Yiming Xie[1] Huaizu Jiang[1] Georgia Gkioxari[*,2] Julian Straub[*,3]
[1]Northeastern University [2]California Institute of Technology [3]Meta Reality Labs Research

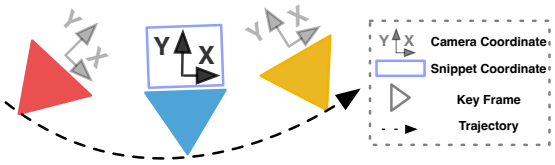

Figure 1: **Snippet coordinate.**

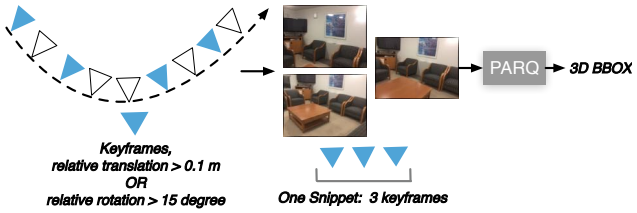

Figure 2: **Snippet generation.**

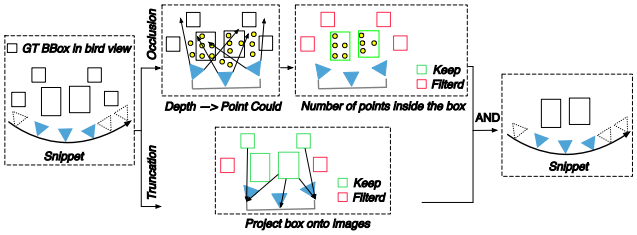

Figure 3: **Extract 3D box annotations for each snippet.**

1. Implementation Details

Snippet coordinate system. The camera coordinates of the middle snippet frame define the snippet coordinate system, as shown in Fig. 1. All 3D predictions are defined with reference to that snippet coordinate system.

2D image backbone. We use ResNet50 [2], pretrained on ImageNet, integrated with a feature pyramid network (FPN) [5]. The input image size is $(3 \times 240 \times 320)$. For simplicity, we omit the batch size and number of views here. The outputs of the image backbone are multi-level features whose sizes are *p0*: $(256 \times 60 \times 80)$, *p1*: $(256 \times 30 \times 40)$, *p2*: $(256 \times 15 \times 20)$, *p3*: $(256 \times 8 \times 10)$. We upsample the feature *p1*, *p2*, and *p3* to size 60×80 and concatenate the multi-level features together. Finally, we get the final image features $(1024 \times 60 \times 80)$.

Log-scale ray points sampling and encoding. We enhance the image features with 3D ray encodings, following [6]. For each image, we shoot rays originating at the camera center intersecting the image at each pixel. We sample D points along each ray, $P^{ray} \in \mathbb{R}^{H \times W \times (D \times 3)}$, with log-scale sampling:

$$d_j = e^{\log_e d_{min} + \frac{j}{D} \log_e \frac{d_{max}}{d_{min}}} \tag{1}$$

where d_j is the depth for j-th point along the ray, d_{min} is the minimal depth and d_{max} is the maximal depth. We follow the implementation of depth sampling in SimpleRecon [8] [1]. D is 64, d_{min} is $0.25m$, and d_{max} is $5.25m$.

The ray points are transformed to position encodings, $P \in \mathbb{R}^{H \times W \times C}$, via an MLP of the same hidden dimension C as the input feature maps. C is 1024 in our model.

Loss computation. We supervise the object detection output in each iteration. We use Hungarian matching [3] to match the predictions and ground truth. The cost matrix $\mathcal{C}$ consists of center distance $\mathcal{C}_{center}$ and classification cost $\mathcal{C}_{class}$,

$$\mathcal{C} = \alpha_0 \mathcal{C}_{center} + \alpha_1 \mathcal{C}_{class} \tag{2}$$

$$\mathcal{C}_{center} = |x_{ref} - x_{gt}| \tag{3}$$

$$\mathcal{C}_{class} = -score[class_{gt}] \tag{4}$$

where x_{ref} is the position of the reference point and x_{gt} is the center position of ground-truth boxes. α_0 is 0.25, and α_1 is 2. Aside from Hungarian matching, we also match the GT box and the predictions whose corresponding reference points are in close proximity to this GT box ($\mathcal{C}_{center} < 0.2m$), since for two adjacent reference points which have the similar queries, they should both detect nearby objects.

∗ Equal advising.

[1] External content: SimpleRecon Github

Table 1: **Quantitative results (Prec./Rec./F1)** on ScanNet using the detection-based metric proposed in [4].

@IoU > 0.25	chair	table	cabinet	trash bin	bookshelf	display	sofa	bathtub	other	**average**
ODAM [4]	39.1/71.7/50.6	44.8/40.4/42.5	**56.5**/5.0/9.3	25.3/25.3/32	35.8/13.7/19.9	9.2/38.7/14.8	53.6/39.8/39.8	28.6/28.3/28.5	0.0/0.0/0.0	33/47.1/38.8
imVoxelNet [7]	59.8/73.6/66	57.1/**54.5**/**55.8**	48.1/40.9/44.2	**51.7**/53.9/**52.8**	27.4/8.5/13.0	0.0/0.0/0.0	47.3/**49.1**/**48.1**	**60.7**/14.4/23.3	**44.6**/**24.8**/**31.9**	**55.2**/**48.6**/**51.7**
DETR3D [9]	40.5/71.2/51.6	31.6/40.2/35.4	27.2/34.6/30.4	15.1/43/22.4	16.4/21.1/18.5	9.7/40.5/15.6	45.7/28.1/34.8	15.3/25.4/19.1	8.1/17.7/11.2	24.4/44.7/31.6
PETR [6]	64.1/79.5/71.0	60.3/41.7/49.3	48.3/**44.4**/**46.4**	40.6/54.8/46.6	**37.6**/**23.6**/**29.0**	19.8/40.0/26.5	66.7/33.3/44.4	44.8/36.4/40.2	26.5/20.6/23.2	49.6/50.5/50.0
Ours	**66.4**/**79.8**/**72.5**	**62.7**/36.5/46.2	56.0/36.6/44.3	47.4/**57.0**/51.8	35.1/14.3/20.4	**22.9**/**46.2**/**30.6**	**72.7**/28.1/40.5	56.6/**39.8**/**46.8**	32.2/16.2/21.6	54.2/48.2/51.1
@IoU > 0.5	chair	table	cabinet	trash bin	bookshelf	display	sofa	bathtub	other	**average**
ODAM [4]	17.4/31.9/22.5	10.4/9.4/9.9	30.4/2.7/5.0	6/10.3/7.6	8.6/3.3/4.8	1.7/7.1/2.7	19/14.2/16.2	6.7/6.7/6.7	0.0/0.0/0.0	12.1/17.3/14.2
imVoxelNet [7]	39.7/48.8/43.8	18/**17.1**/17.5	20.4/17.4/18.8	**21.0**/**21.9**/**21.5**	1.6/0.5/0.8	0.0/0.0/0.0	16.4/17/16.7	32.1/7.6/12.3	**19.2**/**10.7**/**13.8**	28.6/25.2/26.8
DETR3D [9]	19.9/19.9/25.4	6.0/7.6/6.7	10.4/13.2/11.6	11.6/7/3.7	2.6/3.4/3	1.3/5.4/2.1	15.7/9.6/12	3.6/5.9/4.5	1.4/3.1/1.9	8.8/16.1/11.4
PETR [6]	39.9/49.5/44.2	25.1/17.3/20.5	26.9/**24.7**/25.8	11.0/14.9/12.7	**12.1**/**7.6**/**9.3**	3.8/7.6/5.1	33.3/**16.7**/**22.2**	18.8/15.3/16.8	10.6/8.2/9.3	25.4/25.9/25.6
Ours	**47.6**/**57.2**/**52.0**	**28.1**/16.4/**20.7**	**35.2**/23.0/**27.9**	16.8/20.2/18.3	10.3/4.2/6.0	**5.2**/**10.7**/**7.1**	**34.1**/13.2/19.0	**25.3**/**17.8**/**20.9**	14.7/7.4/9.9	**31.8**/**28.3**/**30.0**
@IoU > 0.7	chair	table	cabinet	trash bin	bookshelf	display	sofa	bathtub	other	**average**
ODAM [4]	1.8/3.3/2.3	0.6/0.5/0.6	0.0/0.0/0.0	1.3/**2.2**/1.6	0.0/0.0/0.0	0.0/0.0/0.0	1.2/0.9/1	0.8/0.8/0.8	0.0/0.0/0.0	1.2/1.7/1.4
imVoxelNet [7]	6.1/7.5/6.7	1.8/1.7/1.8	2.4/2.1/2.2	0.8/0.9/0.9	0.0/0.0/0.0	0.0/0.0/0.0	3.6/**3.8**/3.7	**7.1**/**1.7**/**2.7**	2.3/**1.3**/**1.7**	4.1/3.6/3.8
DETR3D [9]	2.5/4.4/3.2	0.1/0.2/0.2	0.3/0.4/0.4	0.0/0.0/0.0	0.0/0.0/0.0	0.1/**0.6**/0.2	0.0/0.0/0.0	1.0/1.7/1.3	0.0/0.0/0.0	0.9/1.7/1.2
PETR [6]	8.2/10.2/9.1	2.5/1.7/2.0	8.1/**7.4**/**7.7**	0.3/0.4/0.4	1.3/**0.8**/0.1	0.0/0.0/0.0	1.8/0.9/1.2	2.1/1.7/1.9	1.6/1.2/1.4	4.6/4.6/4.6
Ours	**10.8**/**13.0**/**11.8**	**3.9**/**2.3**/**2.9**	**8.8**/5.8/7.0	**1.8**/**2.2**/**2.0**	**2.1**/**0.8**/**1.2**	**0.3**/**0.6**/**0.4**	**6.8**/2.6/**3.8**	2.4/**1.7**/2.0	**2.4**/1.2/1.6	**6.5**/**5.8**/**6.1**

Table 2: **Quantitative results (Prec./Rec./F1)** on ARKitScenes using the detection-based metric proposed in [4].

@IoU > 0.25	cabinet	refrigerator	shelf	stove	bed	sink	washer	toilet	bathtub
imVoxelNet [7]	45.8/38.2/41.6	16.7/2.5/4.3	25.9/10.9/15.3	0.0/0.0/0.0	46.1/**70.7**/55.8	33.3/0.6/1.2	69.5/**64.1**/**66.7**	71.4/**85.9**/78.0	**84.5**/89.1/**86.7**
PETR [6]	40.4/**52.9**/45.8	34.7/**60.0**/44.0	18.5/**28.5**/**22.4**	**14.6**/31.8/**20.0**	50.8/62.6/**56.1**	40.3/**63.8**/49.4	58.2/60.9/59.5	63.5/84.3/72.5	61.9/**94.5**/74.8
Ours	**64.2**/42.2/**50.9**	**80.8**/52.5/**63.6**	**30.9**/15.3/20.5	11.9/**38.6**/18.2	**87.9**/29.3/43.9	**54.7**/**63.8**/**58.9**	**88.9**/50.0/64.0	**88.1**/81.3/**84.6**	78.2/78.2/78.2
	oven	dishwasher	fireplace	stool	chair	table	tv monitor	sofa	**average**
imVoxelNet [7]	0.0/0.0/0.0	30.8/23.5/26.7	45.2/45.2/45.2	16.0/**28.9**/**20.6**	49.7/69.3/57.9	32.8/**57.4**/41.7	0.0/0.0/0.0	51.9/**66.8**/58.4	44.5/40.3/42.3
PETR [6]	77.2/55.7/64.7	83.3/29.4/43.5	57.6/**61.3**/**59.4**	18.8/20.0/19.4	38.9/**61.0**/47.5	36.9/52.5/43.3	**4.7**/**16.7**/**7.3**	53.0/64.4/58.1	36.6/**53.2**/43.4
Ours	**81.4**/**60.8**/**69.6**	**100.0**/**35.3**/**52.2**	**61.1**/35.5/44.9	**29.2**/15.6/20.3	**68.6**/**61.0**/**64.6**	**58.6**/41.6/**48.6**	4.0/10.4/5.8	**93.3**/47.1/**62.6**	**54.1**/44.4/**48.8**
@IoU > 0.5	cabinet	refrigerator	shelf	stove	bed	sink	washer	toilet	bathtub
imVoxelNet [7]	11.4/9.5/10.4	0.0/0.0/0.0	0.0/0.0/0.0	0.0/0.0/0.0	21.1/**32.3**/**25.5**	0.0/0.0/0.0	44.1/**40.6**/42.3	37.7/45.3/41.1	20.7/21.8/21.2
PETR [6]	14.6/**19.1**/16.5	15.9/27.5/20.2	4.7/**7.3**/**5.7**	**2.1**/4.5/2.9	22.1/28.3/24.4	19.0/**30.1**/23.3	35.8/37.5/36.6	34.1/45.3/38.9	29.8/45.5/36.0
Ours	**27.3**/17.9/**21.6**	**57.7**/**37.5**/**45.5**	**5.9**/2.9/3.9	**2.1**/**6.8**/**3.2**	**42.4**/14.1/21.2	**22.1**/25.8/**23.8**	**66.7**/37.5/**48.0**	**67.8**/**62.5**/**65.0**	**58.2**/**58.2**/**58.2**
	oven	dishwasher	fireplace	stool	chair	table	tv monitor	sofa	**average**
imVoxelNet [7]	0.0/0.0/0.0	23.1/17.6/20.0	0.0/0.0/0.0	6.1/**11.1**/**7.9**	22.7/31.7/26.4	9.2/16.2/11.8	0.0/0.0/0.0	26.1/**33.7**/29.4	14.8/21.5/17.5
PETR [6]	42.1/30.4/35.3	50.0/17.6/26.1	18.2/**19.4**/18.8	5.2/5.6/5.4	17.2/26.9/21.0	12.3/**17.5**/14.4	**0.6**/**2.1**/**0.9**	26.9/32.7/29.5	14.8/21.5/17.5
Ours	**45.8**/**34.2**/**39.1**	**100.0**/**35.3**/**52.2**	**27.8**/16.1/**20.4**	**8.3**/4.4/5.8	**38.1**/**33.8**/**35.8**	**24.7**/**17.5**/**20.5**	0.5/1.4/0.8	**59.0**/29.8/**39.6**	**26.7**/**21.9**/**24.1**
@IoU > 0.7	cabinet	refrigerator	shelf	stove	bed	sink	washer	toilet	bathtub
imVoxelNet [7]	0.6/0.5/0.5	0.0/0.0/0.0	0.0/0.0/0.0	0.0/0.0/0.0	0.7/1.0/0.8	0.0/0.0/0.0	5.1/4.7/4.9	1.3/1.6/1.4	1.7/1.8/1.8
PETR [6]	2.8/3.7/3.2	4.3/7.5/5.5	0.0/0.0/0.0	0.0/0.0/0.0	0.8/1.0/0.9	2.7/4.3/3.3	10.4/**10.9**/10.7	8.2/11.0/9.4	9.5/**14.5**/**11.5**
Ours	**6.3**/**4.1**/**5.0**	**19.2**/**12.5**/**15.2**	0.0/0.0/0.0	0.0/0.0/0.0	**12.1**/**4.0**/**6.0**	**4.7**/**5.5**/**5.1**	**19.4**/**10.9**/**14.0**	**23.7**/**21.9**/**22.8**	**10.9**/10.9/10.9
	oven	dishwasher	fireplace	stool	chair	table	tv monitor	sofa	**average**
imVoxelNet [7]	0.0/0.0/0.0	7.7/5.9/6.7	0.0/0.0/0.0	0.0/0.0/0.0	3.1/4.3/3.6	0.0/0.0/0.0	0.0/0.0/0.0	3.4/4.3/3.8	1.3/1.2/1.3
PETR [6]	12.3/8.8/10.3	**16.7**/**5.9**/**8.7**	0.0/0.0/0.0	0.0/0.0/0.0	2.1/3.3/2.6	2.1/**3.0**/**2.5**	0.0/0.0/0.0	3.2/3.8/3.5	2.5/3.7/3.0
Ours	**18.6**/**13.9**/**15.9**	0.0/0.0/0.0	**5.6**/**3.2**/**4.1**	0.0/0.0/0.0	**9.4**/**8.3**/**8.8**	**2.3**/1.7/1.9	0.0/0.0/0.0	**14.3**/**7.2**/**9.6**	**6.3**/**5.2**/**5.7**

Extracting video snippets. We provide two figures to illustrate the details of snippet generation from a video in Fig. 2 and 3D box annotations for each snippet in Fig. 3.

2. Full Precision/Recall/F1 for All Classes

We provide the full Prec./Recal./F1 performance on ScanNet in Table 1. We also provide the complete performance table on ARKitScenes in Table 2.

3. Full Qualitative Results

We provide more qualitative results in Fig. 4, Fig. 5 and Fig. 6.

4. Limitations

We provide some failure cases in Fig. 7. Our approach is prone to failure when detecting large objects (Fig. 7 a2, a4, e3, e4), detecting objects with the same color as the background e.g. black object in the dark (Fig. 7 b4, c2, d3, e2), and detecting objects with high occlusion (Fig. 7 b3, c1, d4, e1).

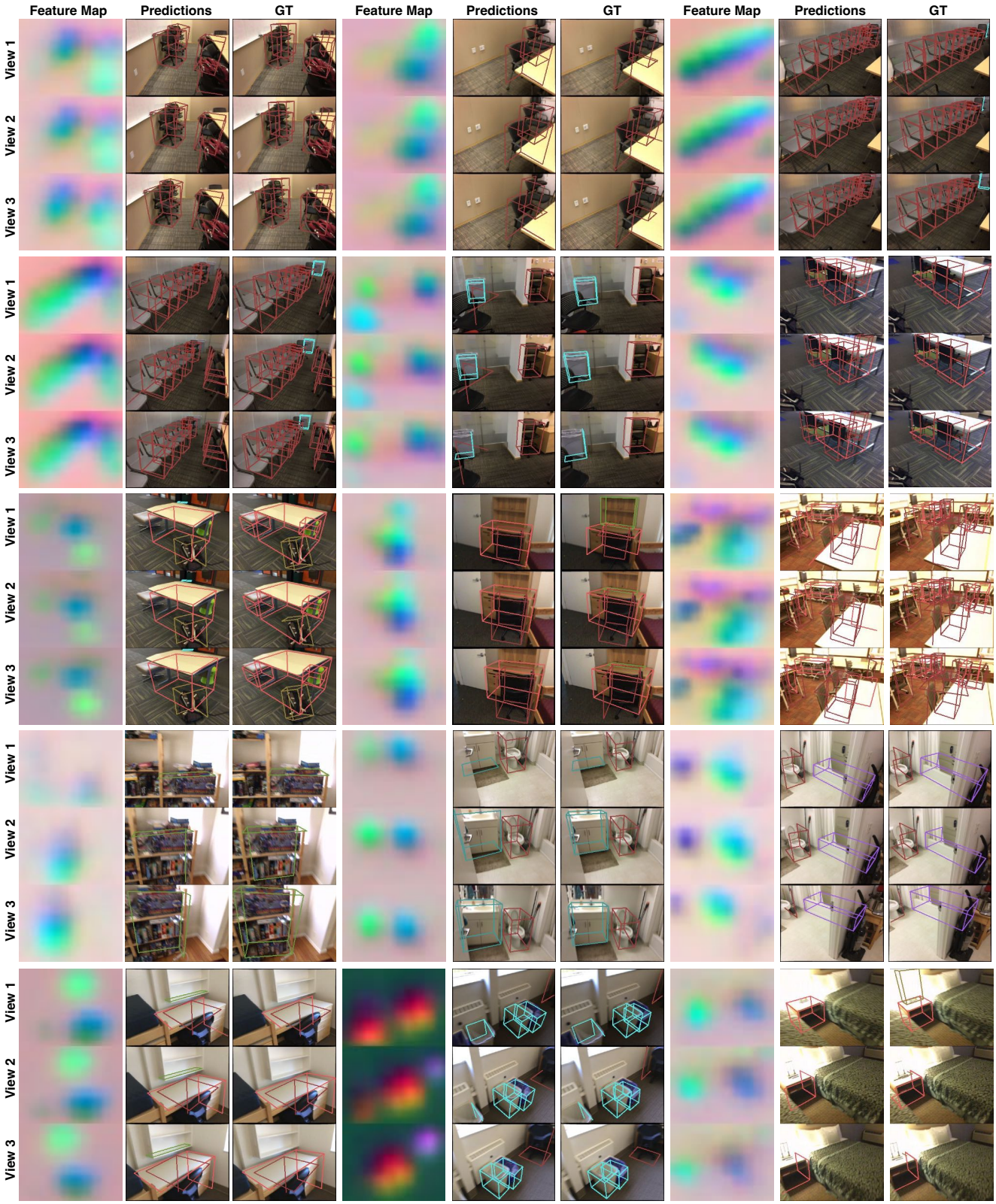

Figure 4: **Qualitative results on ScanNet (1).** *Zoom in for details.* We compress the image feature maps using Linear PCA [1]. Note that the learned feature maps are multi-view consistent.

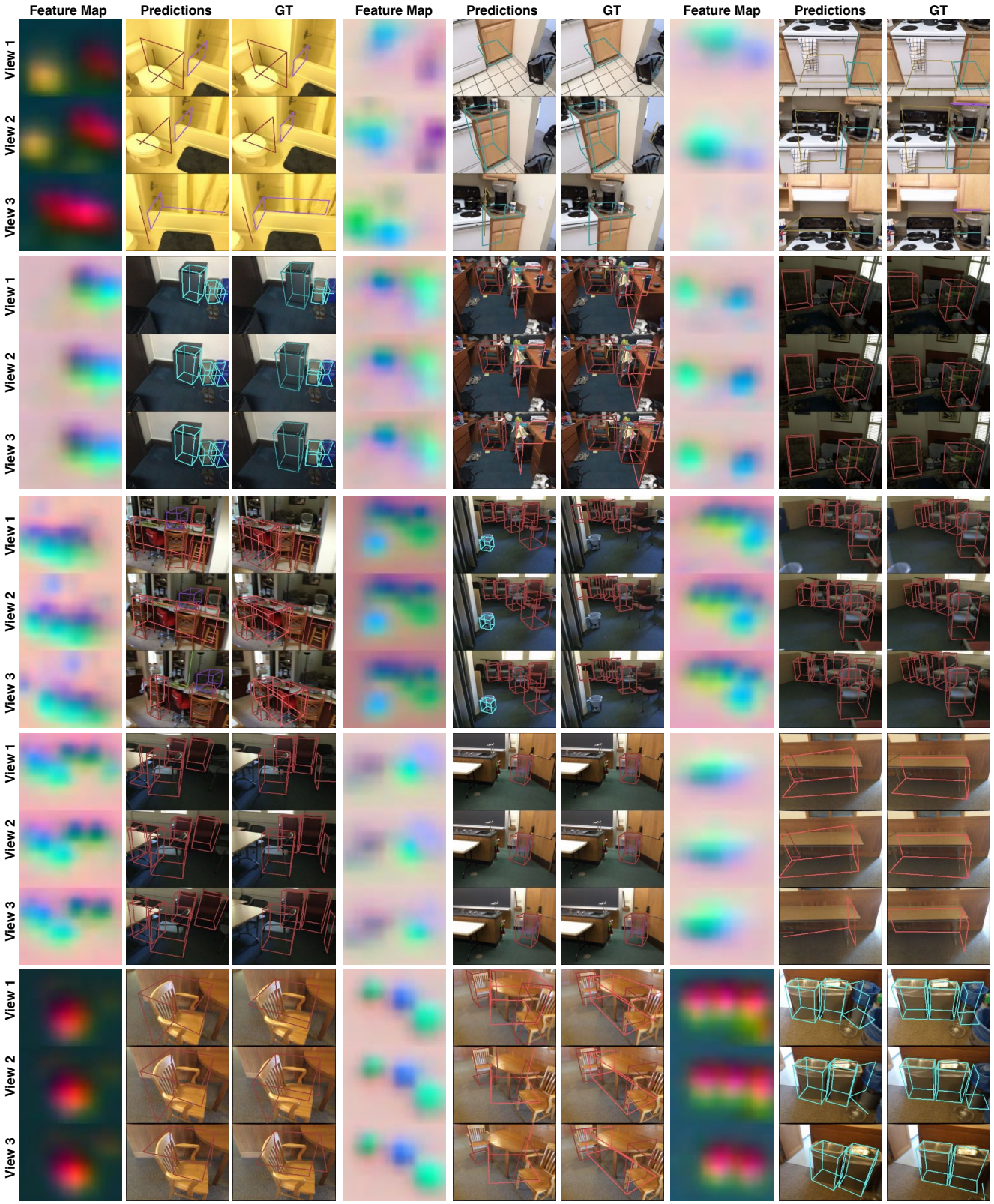

Figure 5: **Qualitative results on ScanNet (2).** *Zoom in for details.* We compress the image feature maps using Linear PCA [1]. Note that the learned feature maps are multi-view consistent.

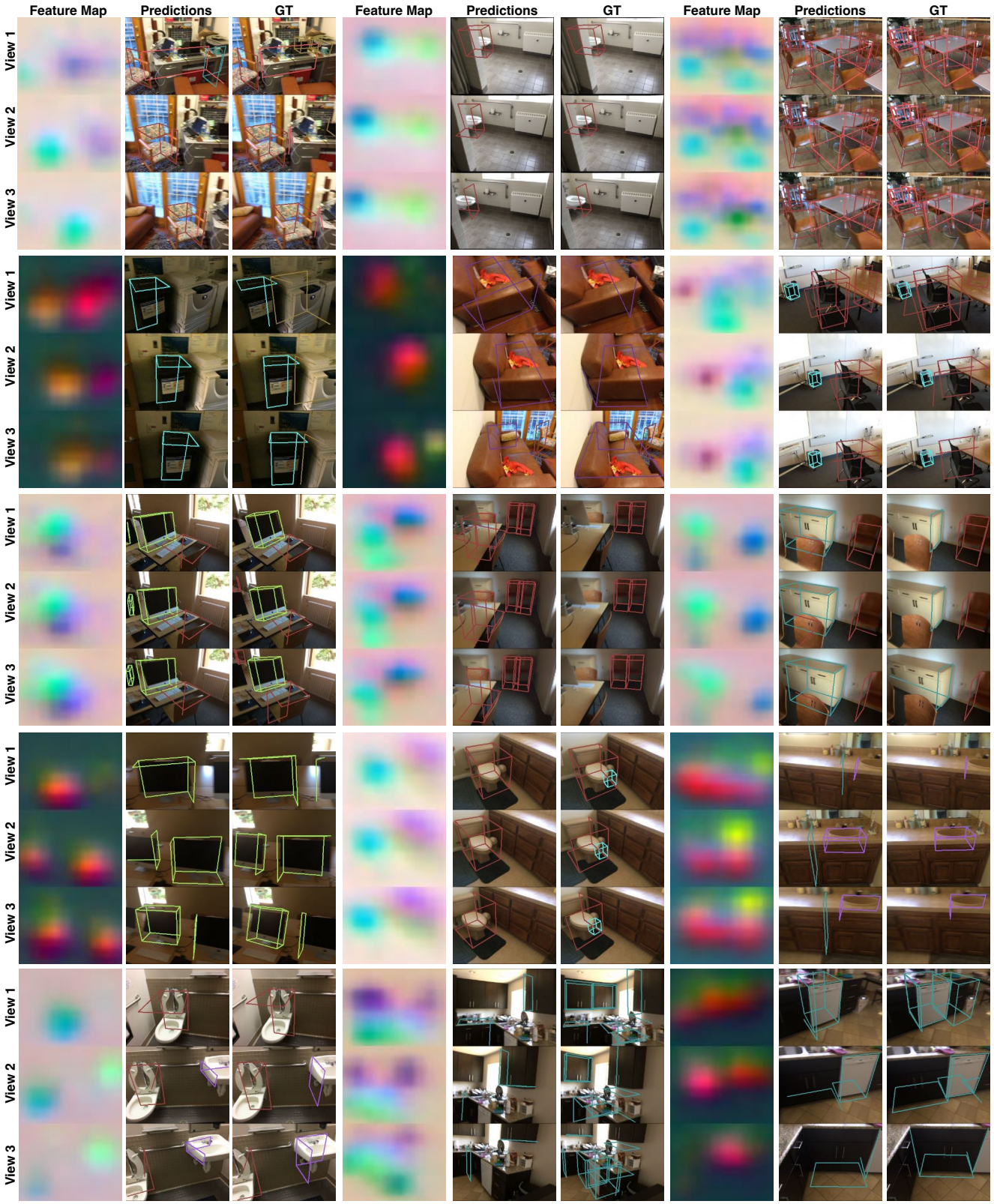

Figure 6: **Qualitative results on ScanNet (3).** *Zoom in for details.* We compress the image feature maps using Linear PCA [1]. Note that the learned feature maps are multi-view consistent.

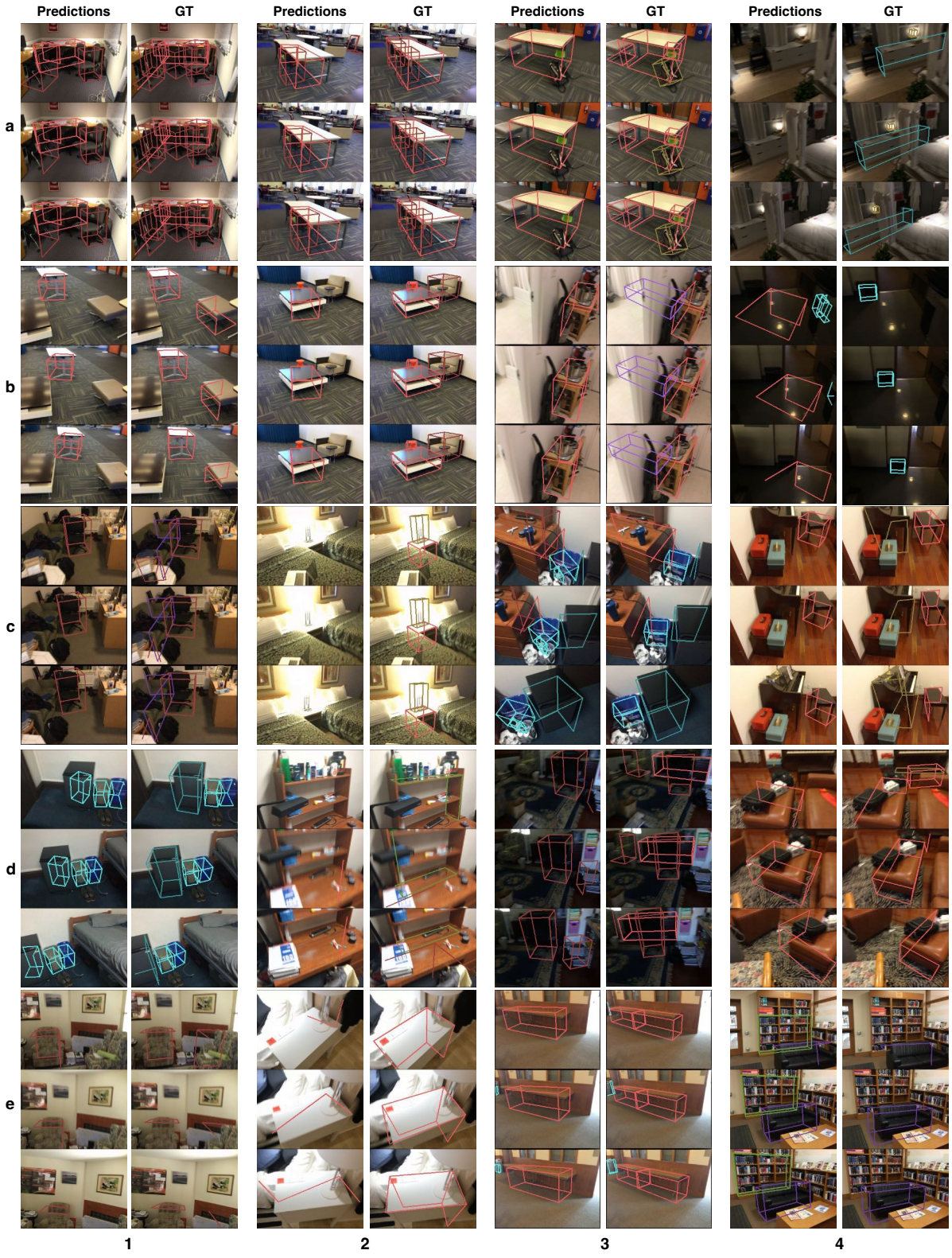

Figure 7: **Failure cases.** *Zoom in for details.* Our approach is prone to failure when detecting large objects (a2, a4, e3, e4), detecting objects with the same color as the background e.g. black object in the dark (b4, c2, d3, e2), and detecting objects with high occlusion (b3, c1, d4, e1).

References

[1] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 2011. 3, 4, 5

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[3] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 1955. 1

[4] Kejie Li, Daniel DeTone, Steven Chen, Minh Vo, Ian Reid, Hamid Rezatofighi, Chris Sweeney, Julian Straub, and Richard Newcombe. ODAM: Object Detection, Association, and Mapping using Posed RGB Video. In *ICCV*, 2021. 2

[5] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *CVPR*, 2017. 1

[6] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. PETR: Position Embedding Transformation for Multi-View 3D Object Detection. In *ECCV*, 2022. 1, 2

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

[8] Mohamed Sayed, John Gibson, Jamie Watson, Victor Prisacariu, Michael Firman, and Clément Godard. Simplerecon: 3d reconstruction without 3d convolutions. In *ECCV*, 2022. 1

[9] Yue Wang, Vitor Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, , and Justin M. Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *CoRL*, 2021. 2