Appendix

A. Approach Details

In this section, we show the details of our AShapeFormer, which is divided into naive shape encoding method, Channel Attention Module (CAM), ShapeFormer and semantics Guided Module (SGM).

A.1. Naive method and CAM

Naive method. In the naive shape encoding method, the backbone is based on the PointNet++ architecture [15], has four set abstraction layers and two feature upsampling layers. We follow the same layer parameters (e.g. ball-region radius, number of sample points, and MLP channels) as VoteNet [13]. We sample 1024 seed points and 256 candidate points. Before shape encoding, the candidate point aggregates n vote point features within a radius r. In the naive method, n is 16 and r is 0.3. The input tensor dimension of the candidate aggregation module is $(8, 256, 1024)$, 8 is the batch size, 256 is the dimension of the vote feature and 1024 is the number of vote points. We perform farthest point sampling (FPS) in vote points, and 256 candidates were sampled. We perform query and group [15] on 256 candidates, and obtain a 4D$(8, 512, 256, 16)$ tensor. Feature aggregation is performed on 16 vote points within each candidate neighborhood, and the output tensor dimension is $(8, 128, 256)$. Compared to seed points, voting points are more compact. Therefore, it is easier to obtain shape key points distributed on the same object surface by sampling the seed points with the same index as the candidate aggregation. Naive object-level shape features are obtained using MLP and Max-pooling like candidate aggregation. The output of the naive shape encoding module, like the candidate aggregate, is a $(8, 128, 256)$ tensor.

CAM. The input of our CAM is the candidate features $p(\mathcal{O})$ and object-level shape features $p(\mathcal{C})$, both of which have a feature channel of 256. The calculation of CAM is as follows:

$$p(\mathcal{O}) = \sigma(\mathrm{FC}_s(p(\mathcal{O}))) * p(\mathcal{O}), \quad (1)$$

$$p(\mathcal{C}) = \sigma(\mathrm{FC}_s(p(\mathcal{C}))) * p(\mathcal{C}), \quad (2)$$

where FC_s represent 2 layers of fully connected layer with [256,64,256]. $\sigma(\cdot)$ is the sigmoid activation.

A.2. ShapeFormer architecture details

ShapeFormer is a multi-layer multi-head self-attention [18] module. We use one layer of self-attention block and the number of heads is 4. The input of the ShapeFormer module is candidate points and the shape key point, the dimensions of the input data are shown in Table. 1. We regard the candidate feature as shape token [5, 8], shape token and shape key point features are concatenated together and fed into the self-attention module. The output dimension of the last layer of ShapeFormer is $(8, 256, 256, 17)$. The shape token of the output layer is used as the object-level shape feature, fused with the candidate feature, and fed to the detection head.

Input	coordinate	Feature
Candidate	(8, 256, 3, 1)	(8, 256, 256, 1)
SKP	(8, 256, 3, 16)	(8, 256, 256, 16)

Table 1. ShapeFormer input data dimension. SKP is shape key points.

Input Branch	Input MLP Channels
Joint	[512, 256, 64, 1]
Point	[256, 128, 64, 1]
Image	[256, 128, 64, 1]

Table 2. Semantic segmentation layer parameters of SGM module applied to imVoteNet.

Object-Scene Positional Encoding. The position encoding [18] is learned from point coordinates. Our position encoding is divided into two parts: object-level encoding and scene-level encoding. The input tensor dimension of scene-level encoding is $(8, 256, 3, 17)$, the last dimension 17 includes 1 candidate point and 16 shape key points. A 256-dimensional positional encoding is learned through a $[3, 128, 256]$ MLP$_s$. The object-level encoding is the position encoding in the canonical coordinate system centered on the candidate point. Since the candidate point is the origin of the coordinates in the object-level coordinate system, its position is encoded as $\mathbf{0}$ in the object-level position encoding. The input dimension is $(8, 256, 3, 17)$, the last dimension 17 includes a coordinate origin and 16 relative positions, and then also learns a 256-dimensional position encoding by $[3, 128, 256]$ MLP$_s$. Finally, The object-level positional encoding and scene-level positional encoding are added to the feature and fed to the ShapeFormer.

A.3. SGM architecture details

When we use SGM, the backbone follows the same layer as the naive method (§ A.1) and VoteNet [13] except for the second SA (set abstraction) [15] layer. To guide shape encoding with semantic information, we append a segmentation head for estimating the foreground confidence of each point. We keep more seed points in the SA layer, then sort these seed points by segmentation confidence, and finally pick out the highest quality the 1500 seed points are used for shape encoding. Since we sampled more foreground points, we could get more shape key points in ShapeFormer, so when we applied SGM to ShapeFormer, we chose 24 shape key points instead of 16. The segmentation head uses 3 layers of MLP$_s$, and finally outputs a 1-dimensional ten-

sor. After passing a Sigmoid function, the 1D tensor represents segmentation scores. When the SGM applied to the VoteNet [13], The segmentation head is composed of 3 layers of MLP [256, 128, 64, 1]. In the implementation based on imVoteNet [12], for the joint branch, point branch and image branch, the specific parameters are shown in Table 2. We implement our method on MMDetection3D [3] with one NVIDIA RTX 3090 GPU. Figure. 1 visualizes the positive impact of the SGM module on seed point sampling and voting during testing.

A.4. AShapeFormer loss function details

As mentioned in the main paper, our model is trained end-to-end with a multi-task loss including the SGM loss $\mathcal{L}_{\text{sgm}}$, voting regression loss $\mathcal{L}_{\text{vote}}$, objectness loss $\mathcal{L}_{\text{obj}}$, bounding box estimation loss $\mathcal{L}_{\text{box}}$, and semantic classification $\mathcal{L}_{\text{cls}}$ losses.

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{sgm}} + \lambda_2 \mathcal{L}_{\text{vote}} + \lambda_3 \mathcal{L}_{\text{box}} + \lambda_4 \mathcal{L}_{\text{cls}} + \lambda_5 \mathcal{L}_{\text{obj}}, \quad (3)$$

The balancing factors λ's are set as $\lambda_1 = 3.0$, $\lambda_2 = 10.0$, $\lambda_3 = 10.0$, $\lambda_4 = 1.0$ and $\lambda_5 = 5.0$.

$\mathcal{L}_{\text{sgm}}$ is used to supervise the foreground/background seed points prediction in SGM, which we define as follows

$$\mathcal{L}_{\text{sgm}} = -\frac{1}{M} \sum_{i=1}^{M} \left[\widehat{p_i} \ln (p_i) + (1 - \widehat{p_i}) \ln (1 - p_i) \right], \quad (4)$$

where p_i and $\widehat{p_i}$ denote the predicted segmentation score and the ground-truth score (1 for foreground and 0 for background). M is the total number of input points.

Following VoteNet [13] and imVoteNet [12], the vote loss $\mathcal{L}_{\text{vote}}$ is defined as

$$\mathcal{L}_{\text{vote}} = \frac{1}{M} \sum_{i=1}^{M} \|\Delta x_i - \Delta x_i^*\| \, \mathbb{1} \left[s_i \text{ on object } \right], \quad (5)$$

where $\mathbb{1} \left[s_i \text{ on object } \right]$ indicates whether a seed point s_i is a foreground point. the box loss $\mathcal{L}_{\text{box}}$ is defined as

$$\begin{aligned} \mathcal{L}_{\text{box}} &= \mathcal{L}_{\text{center-reg}} + 0.1 \mathcal{L}_{\text{angle-cls}} + \mathcal{L}_{\text{angle-reg}} \\ &\quad + 0.1 \mathcal{L}_{\text{size-cls}} + \mathcal{L}_{\text{size-reg}}, \end{aligned} \quad (6)$$

We refer readers to [13, 14] for more details about $\mathcal{L}_{\text{obj}}$, $\mathcal{L}_{\text{box}}$, and $\mathcal{L}_{\text{cls}}$.

A.5. Training details

When we trained our AShapeFormer based on the MMdetection3D [3], we used the AdamW [10] optimizer ($\beta 1 = 0.9, \beta 2 = 0.999$) with 48 epochs. We set the initial learning rate to 0.001 when training on the SUNRGBD dataset [16] and 0.008 when training on the Scannet V2 datase [4] with a batch size of 8, and is decayed $10\times$ at 24, 32 and 40 epochs. The learning rate of the ShapeFormer is set as 1/10 of that in the backbone network. Gradient normalized clipping is used, with maximum norm of 10.

Seed	Candidate	VoteNet	ours
1024	256	63.8	65.8
1200	256	63.4	66.0
1400	256	63.0	65.9
1500	256	63.9	66.6
1600	256	63.9	66.4

Table 3. Ablation study on the performance of AShapeFormer with different seed point number on SUN RGB-D dataset.

B. More Ablation Study and Discussion

B.1. Number of seed points

Since our AShapeFormer is guided by semantic information, we can sample more seed points in the backbone without worrying about introducing a large number of background points. More foreground points would be a great help to our shape encoding. Table 3 shows that our AShapeFormer improves with the increase in the number of sampling seed points. However, when we increased the seed points of vanilla VoteNet [13], there was no significant improvement in performance. This is because sampling more seed points without any restrictions will inevitably introduce a large number of background points, which is very unfavorable to the detection results. At the same time, it can be seen that even if we use the same number of 1024 seed points as VoteNet, AShapeFormer performance is considerably better than VoteNet.

B.2. Sampling Strategy

In order to sample more foreground points, in addition to SGM, we also analyzed various foreground point sampling methods, such as KPS [9], FBS [19], F-FPS [21] and Cls-aware [22].

KPS. K-Closest Points Sampling (KPS) is a candidate point sampling method proposed by GroupFree3D [9], a point is assigned a positive value if it lies within the ground truth object box and it is one of the k closest points to the object center. In inference, initial candidates are selected based on the classification scores of the points.

F-FPS. Feature-FPS (F-FPS) is a point cloud downsampling algorithm proposed by 3DSSD [21]. It utilizes the feature distance as the criterion in FPS, many similar useless negative points will be mostly removed.

FBS. Foreground Biased Sampling (FBS) is proposed by RBGNet [19]. It adds a semantic segmentation head to the backbone, and samples foreground and background points according to a point ratio. Its purpose is to sample more points on foreground object surfaces while still keeping the coverage rate of the whole scene.

Cls-aware. Class-aware Sampling (Cls-aware) is proposed by IASSD [22]. The segmentation network not only distinguishes foreground and background points, but also per-

Method	KPS	FBS	F-FPS	IASSD	Ours (SGM)
mAP@0.25	62.4	61.3	63.4	62.1	65.8

Table 4. Experimental results of imVoteNet [12] using different sampling algorithms on the SUNRGBD [16] dataset.

Model	Batchsize	Epoch	mAP@0.25
BRNet [2]	8	220	61.1
RBGNet [19]	8	360	64.1
DisArm [6]	8	220	65.1
AShapeFormer	8	48	65.8

Table 5. Comparison on training epoch on SUN RGB-D dataset.

forms semantic segmentation for each category. In experiments, we found that due to the limited accuracy of multi-category segmentation, it did not significantly help the detection results.

We apply several different sampling algorithms mentioned above in AShapeFormer. As shown in Table 3, the experimental results show that foreground sampling with our SGM is more helpful to the detection results. We observe that in the network sampling these algorithms, it takes a long training epoch, such as 360 epoch of FBS [19], we only need 48 epoch. Therefore, these algorithms may not be able to achieve their potential during the 10-fold shortened training process. We use one of the simplest foreground point bias sampling algorithms, and the results show its superiority. We keep more seed points 2048, and then select 1500 with the highest scores through semantic score. Although this method is simple, it is very effective.

B.3. Training Epoch

We have compared the training epoch and mAP of our AShapeFormer and other shape encoding methods. From Table 6, we can see that compared with the recent BRNet and RBGNet, when batch size is set to 8, we need very little training epoch but reach far beyond their mAP. This fully demonstrates the advantages of our AShapeFormer over other methods.

B.4. Positional Encoding

We conduct an extensive ablation study to analyze the efficacy of different Positional Encoding methods of our method. Table 6 compares the detection results of the Scene-level Positional Encoding, Object-level Positional Encoding combined with the AShapeFormer (VoteNet*) on the SUN RGB-D dataset when the IOU is 0.25.

B.5. Inference Speed

The realistic inference speed of our method is competitive with other state-of-the-art methods. For a fair comparison, all experiments are run on the same device (single NVIDIA RTX 3090 GPU, 62G RAM, and i9-10980XE CPU). The results are shown in Table 7. Our method achieves better performance with a competitive speed. Our method significantly improved expressiveness with little increase in reasoning time, as compared to VoteNet and imVoteNet. At the same time, as compared to other recently shape encoding method such as RBGNet [19], our method offers a dual advantage in speed and precision.

None	✓	✓	✓	✓
Scene-level		✓		✓
Object-level			✓	✓
mAP@0.25	57.7	58.9	59.5	62.2

Table 6. Contribution of Positional Encoding of AShapeFormer (SUN RGB-D dataset). None is without the Positional Encoding.

Model	Frames/s	mAP
VoteNet* [13]	26.1	59.7
imVoteNet* [12]	13.1	64.5
3DETR [11]	6.6	59.1
BRNet [2]	21.9	61.1
RBGNet [19]	3.1	64.1
Ours(VoteNet*)	19.2	62.2
Ours(imVoteNet*)	10.0	65.8

Table 7. Comparison on realistic inference speed on SUN RGB-D validation set with mAP@0.25.

C. More Experimental Results

C.1. Per-category results.

We evaluate per-category on ScanNet V2 [4] under different IoU thresholds. Table 8 and Table 9 report the results on 18 classes of ScanNet V2 with 0.25 and 0.5 box IoU thresholds respectively. Taking VoteNet [13] as the baseline, our method achieves remarkable 4.5% and 6.6% improvements at mAP@0.25 and mAP@0.5, respectively. our AShapeFormer applied to VoteNet* [13] achieves 2.8% and 3.6% improvements at mAP@0.25 and mAP@0.5, respectively. Applying AShapeFormer to the more recent Transformer method GroupFree3D [9] also has a significant improvement with 1.3% and 0.6%. These improvements are achieved by using AShapeFormer to better encode object-level shape feature.

C.2. Visualization of SGM

Figure 1 visualizes the positive impact of the SGM module on seed point sampling and voting during testing. The second row of Fig. 1 is the seed points sampled in the test phase of vanilla votenet. The following two rows show the seed points selected by our SGM module and the generated voting points. The second row shows that VoteNet's FPS sampling covers the entire scene but picks a large number of background points. These background points will be forced

Model	mAP	cab	bed	chair	sofa	table	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn
3D-SIS [7]	40.2	19.7	69.7	66.1	71.8	36.1	30.6	10.9	27.3	0.0	10.0	46.9	14.1	53.8	35.9	87.6	42.9	84.2	16.2
HGNet [1]	61.3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
VoteNet [17]	58.6	36.2	87.9	88.7	89.6	58.7	47.3	38.1	44.6	7.8	56.1	71.6	47.2	45.3	57.1	94.9	54.7	92.1	37.2
VoteNet∗ [17]	63.8	51.1	86.5	89.1	88.2	65.7	54.6	45.1	57.8	16.3	60.6	72.2	50.6	50.6	72.1	98.9	64.5	92.2	50.8
MLCVNet [20]	64.4	42.4	88.4	88.9	87.4	63.5	56.9	46.9	56.9	11.9	63.9	76.0	56.7	60.8	65.9	98.3	59.1	87.2	47.8
3DETR [11]	65.0	49.4	83.6	90.9	89.8	67.6	52.4	39.6	56.4	15.2	55.9	79.2	58.3	57.6	67.6	97.2	70.6	92.2	53.0
GroupFree3D [9]	69.1	52.1	91.9	93.6	88.0	70.7	60.7	53.7	62.4	16.1	58.5	80.9	67.9	47.0	76.3	99.6	72.0	95.3	56.4
RBGNet [19]	70.2	52.6	91.3	93.1	89.7	73.5	60.1	51.9	53.5	20.0	72.6	82.5	63.5	59.8	76.0	99.2	74.7	92.6	55.8
Ours(VoteNet)	**63.1(+4.5)**	46.6	89.5	89.5	88.3	65.4	53.2	43.4	54.7	14.4	49.8	67.3	53.3	49.0	73.2	96.6	64.4	92.5	45.0
Ours(VoteNet∗)	**66.6(+2.8)**	50.0	90.6	91.3	91.7	68.1	58.8	49.9	52.5	19.5	58.0	76.6	49.6	50.4	75.2	99.0	71.7	92.4	53.6
Ours(GroupFree3D)	**70.4(+1.3)**	52.1	82.2	91.5	90.1	75.8	60.9	50.3	62.6	14.5	65.7	80.6	73.7	60.3	81.2	100.0	72.1	96.8	56.2
Ours(RBGNet)	**71.1(+0.9)**	54.6	89.0	93.9	91.9	75.3	64.5	61.3	59.2	25.4	63.1	80.0	57.8	66.4	75.7	99.3	72.4	93.2	57.8

Table 8. 3D object detection results on ScanNet V2 validation set with mAP@0.25. ∗ denotes that the model is implemented on MMDetection3D [3]. Ours ($\mathcal{M}$) denotes that $\mathcal{M}$ is enhanced with our AShapeFormer.

Model	mAP	cab	bed	chair	sofa	table	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn
3D-SIS [7]	22.5	5.7	50.2	52.5	55.4	21.9	10.8	0.0	13.1	0.0	0.0	23.6	2.6	24.5	0.8	71.7	8.9	56.4	6.8
HGNet [1]	34.4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
VoteNet [17]	33.5	8.0	76.0	67.2	68.8	42.3	15.3	6.4	28.0	1.25	9.5	37.5	11.5	27.8	9.9	86.5	16.7	78.8	11.6
VoteNet∗ [17]	44.2	23.1	77.7	76.7	70.6	46.9	30.4	15.7	45.7	4.6	27.4	49.8	30.0	36.9	20.9	90.7	32.5	83.3	28.4
MLCVNet [20]	42.1	16.6	83.3	78.1	74.7	55.1	28.1	17.0	51.7	3.7	13.9	47.7	28.6	36.3	13.4	70.9	25.6	85.7	27.5
3DETR [11]	47.0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
GroupFree3D [9]	52.8	26.0	81.3	82.9	70.7	62.2	41.7	26.5	55.8	7.8	34.7	67.2	43.9	44.3	44.1	92.8	37.4	89.7	40.6
RBGNet [19]	54.2	30.6	80.9	86.5	84.8	66.4	40.3	29.5	48.6	7.9	44.7	59.1	40.8	44.8	39.7	92.9	45.3	90.9	41.5
Ours(VoteNet)	**41.6(+8.1)**	20.1	80.8	76.1	70.2	53.6	31.5	14.7	30.4	5.4	25.5	33.1	27.3	35.5	13.6	89.1	30.5	89.7	22.2
Ours(VoteNet∗)	**47.8(+3.6)**	25.3	81.1	80.8	71.3	56.8	33.3	20.9	53.8	6.1	33.8	56.9	29.2	38.0	33.4	86.5	40.1	80.3	30.9
Ours(GroupFree3D)	**53.4(+0.6)**	30.3	82.5	82.5	74.2	64.6	39.8	26.7	56.2	6.8	34.2	69.9	47.3	41.6	43.2	89.9	39.9	91.1	40.8
Ours(RBGNet)	**56.6(+1.4)**	31.0	82.2	86.9	87.7	67.6	43.1	35.4	57.2	13.2	34.8	59.8	37.2	52.8	50.0	97.6	43.7	91.0	46.4

Table 9. 3D object detection results on ScanNet V2 validation set with mAP@0.5. ∗ denotes that the model is implemented on MMDetection3D [3]. Ours ($\mathcal{M}$) denotes that $\mathcal{M}$ is enhanced with our AShapeFormer.

to predict the displacement relative to the center of the object in the voting process. Therefore, the voting points are of low quality, not only not close to the center of the object, but also somewhat distributed outside the bounding box. The fourth row of the Fig. 1 is the seed point sampling result guided by the semantic information. It can be seen that most of our seed points are foreground points. Based on sampling that is more biased towards the foreground points, we get better voting results, as shown in the 5th row of the Fig. 1, our vote points are mostly close to the center of the object and are also very close to each other, which is very important for the subsequent shape encoding and 3D bounding box prediction.

C.3. More Qualitative Results

We provide more qualitative comparisons between our method and the baseline methods on ScanNet V2 and SUN RGB-D datasets. Figure 2 visualizes the detection results on the Scannet V2 dataset. We compare our AShapeFormer with VoteNet. As stated in the main text, our method has a stronger ability to eliminate false positives. For example, the table in the first row, the cabinet in the fourth row and the chair in the fifth row. The detection results of VoteNet have a large number of false positives, our AShapeForemer eliminates these false positives and obtains more reliable and accurate results.

Our method utilizes the complete object-level shape features, so it can classify objects more accurately. For example, the cabinet enclosed by the pink box in the upper left corner of the third row, votenet wrongly predicts it as door, and the garbagebin in the lower left corner, votenet thinks it is a cabinet. Our AShapeFormer predicts more correctly for these difficult scenarios.

We also show some typical failure cases in Fig. 2. As shown in row 2, When an object consists of multiple clearly demarcated parts, our AShapeFormer cannot avoid the existence of false positive prediction bounding boxes. As shown in the third and fourth rows of the Fig. 2, when the point cloud is too sparse or incomplete, our AShapeFormer will miss object.Tackling these missed detections when the

points are too sparse and incomplete is an interesting and important future direction of our work.

Figure 3 visualizes the experimental results of imvotenet and our imvotenet-based AShapeFormer.

D. Limitations

Although our method achieves promising performance on multiple datasets, there are still some limitations. Compared with the previous approaches, AShapeFormer achieves a large improvement without adding much computation. However, As shown in Fig. 1 and Fig. 2, despite the guidance of semantic information, it is still not guaranteed to eliminate all outliers. Secondly, due to the sparseness and incompleteness of point clouds, there will be missed detections. In the future, we will explore to incorporate RGB images and point cloud completion methods to encode more complete shape information in our technique.

References

[1] Jintai Chen, Biwen Lei, Qingyu Song, Haochao Ying, Danny Z Chen, and Jian Wu. A hierarchical graph network for 3d object detection on point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 392–401, 2020. 4

[2] Bowen Cheng, Lu Sheng, Shaoshuai Shi, Ming Yang, and Dong Xu. Back-tracing representative points for voting-based 3d object detection in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8963–8972, 2021. 3

[3] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d, 2020. 2, 4

[4] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 2, 3, 8

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 1

[6] Yao Duan, Chenyang Zhu, Yuqing Lan, Renjiao Yi, Xinwang Liu, and Kai Xu. Disarm: Displacement aware relation module for 3d detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16980–16989, 2022. 3

[7] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4421–4430, 2019. 4

[8] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3588–3597, 2018. 1

[9] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2949–2958, 2021. 2, 3, 4

[10] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 2

[11] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2906–2917, 2021. 3, 4

[12] Charles R Qi, Xinlei Chen, Or Litany, and Leonidas J Guibas. Imvotenet: Boosting 3d object detection in point clouds with image votes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4404–4413, 2020. 2, 3, 9

[13] 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. 1, 2, 3, 8

[14] Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018. 2

[15] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 1

[16] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015. 2, 3, 9

[17] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 808–816, 2016. 4

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[19] 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. 2, 3, 4

[20] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Yiming Zhang, Kai Xu, and Jun Wang. Mlcvnet: Multi-level context votenet for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10447–10456, 2020. 4

[21] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings*

of the IEEE/CVF conference on computer vision and pattern recognition, pages 11040–11048, 2020. 2

[22] Yifan Zhang, Qingyong Hu, Guoquan Xu, Yanxin Ma, Jianwei Wan, and Yulan Guo. Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18953–18962, 2022. 2

Figure 1. Semantic Guided Sampling and Voting. Rows 2-4 are VoteNet seed points, VoteNet vote points, our seed points and our vote points. Best viewed on screen.

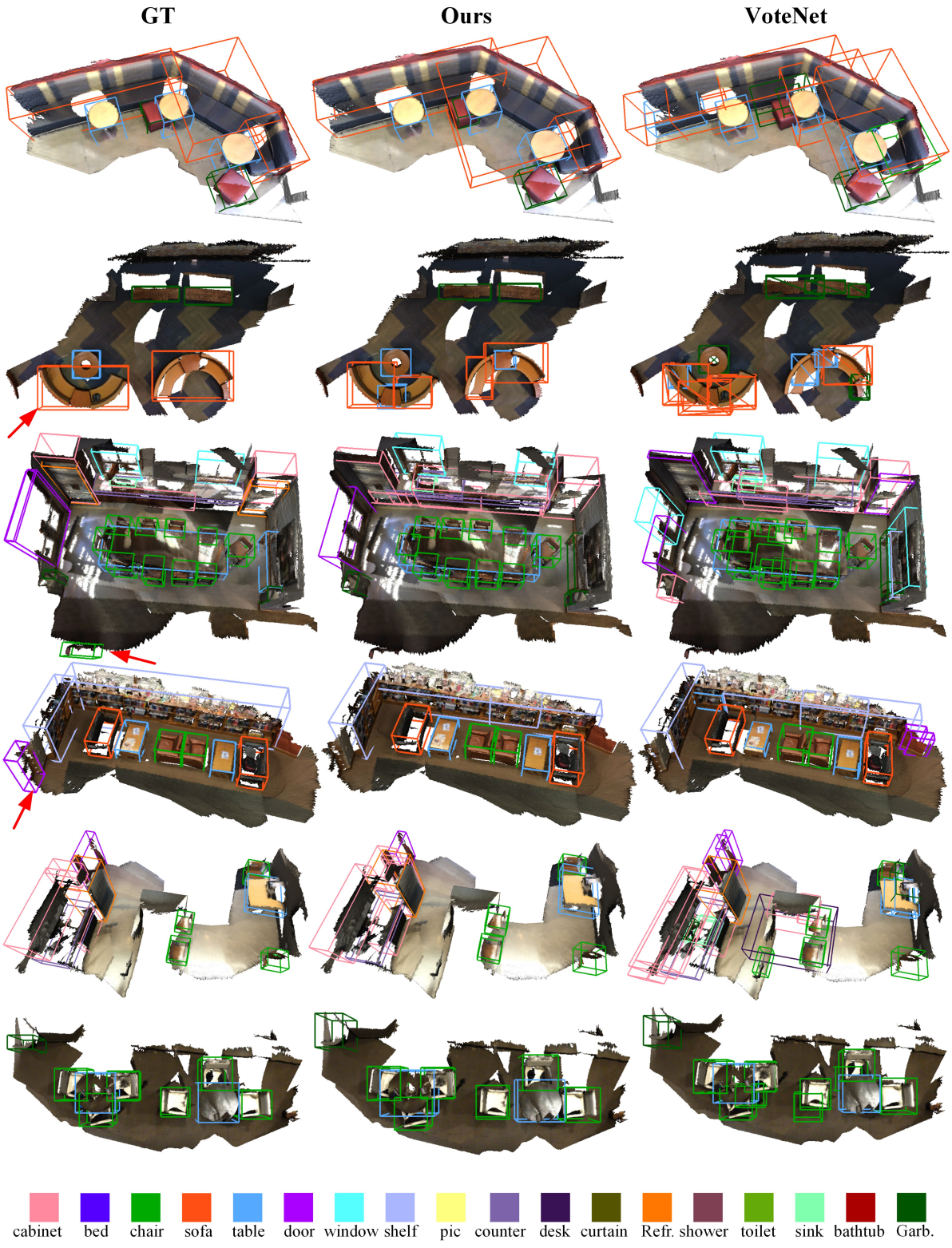

Figure 2. Representative qualitative results on ScanNet V2 dataset [4]. As compared to the baseline, i.e., VoteNet [13], AShapeFormer enhancement not only enables detection of more challenging objects, but also reduces false positive detections. Best viewed on screen.

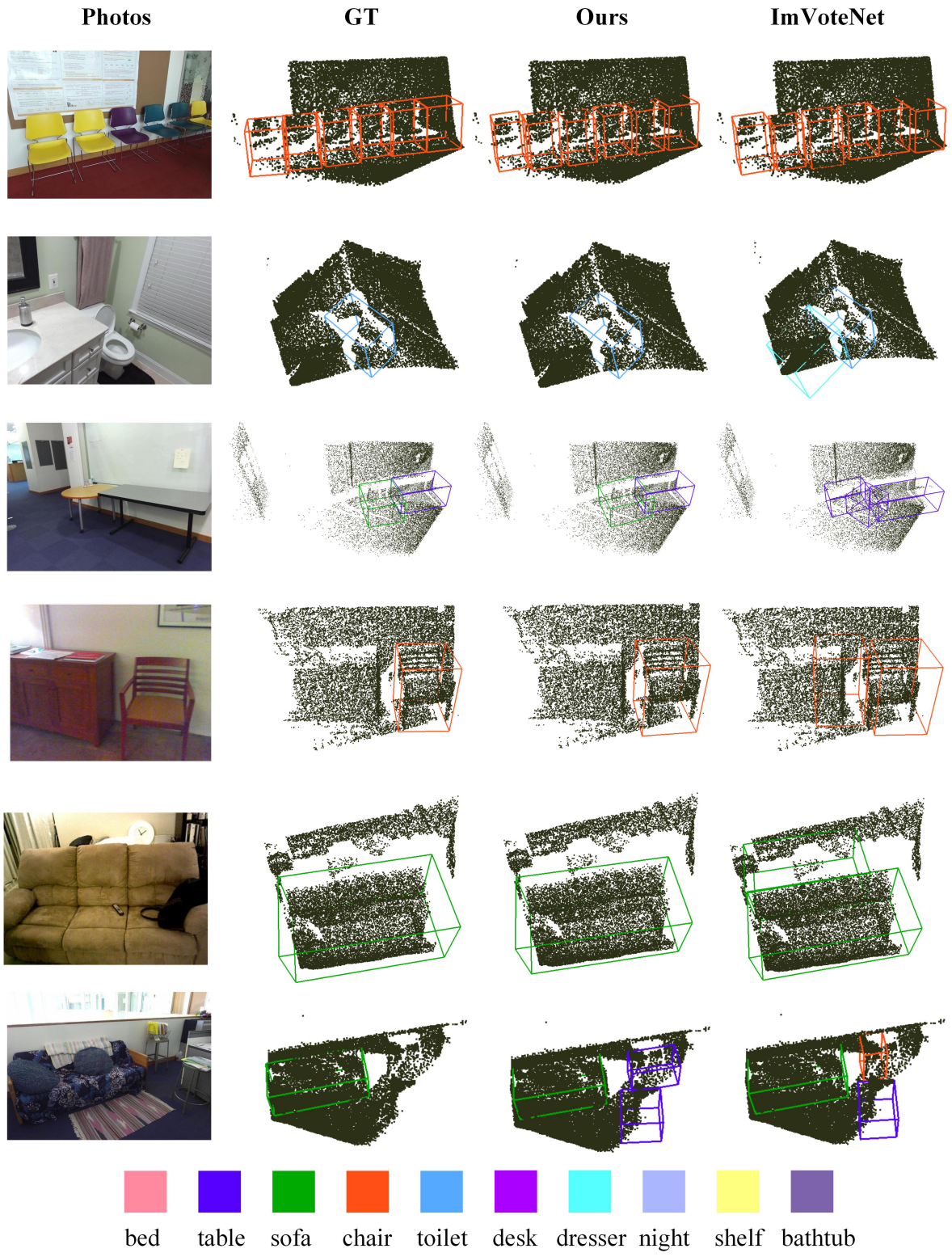

Figure 3. More qualitative results of different 3D object detection methods on SUN RGBD dataset [16]. The baseline methods is imVoteNet [12]. Best viewed on screen. Our method often correctly detects those objects for which ground truth annotation is not provided. This implies that mAP values of our method are under-estimated.