ODIN: A Single Model for 2D and 3D Segmentation Supplementary Materials

Ayush Jain[1], Pushkal Katara[1], Nikolaos Gkanatsios[1], Adam W. Harley[2], Gabriel Sarch[1], Kriti Aggarwal[3], Vishrav Chaudhary[3], Katerina Fragkiadaki[1]
[1]Carnegie Mellon University, [2]Stanford University, [3] Microsoft
{ayushj2, pkatara ngkanats,gsarch,kfragki2}@andrew.cmu.edu
aharley@cs.stanford.edu, {kragga,vchaudhary}@microsoft.com

1. Experiments

1.1. Evaluations on ScanNet and ScanNet200 Hidden Test Sets

We submit ODIN to official test benchmarks of ScanNet [5] and ScanNet200 [18]. Following prior works, we train ODIN on a combination of train and validation scenes. Unlike some approaches that employ additional tricks like DBSCAN [20], ensembling models [13], additional specialized augmentations [24], additional pre-training on other datasets [25], finer grid sizes [23] and multiple forward passes through points belonging to the same voxel, our method avoid any such bells and whistles.

The results are shown in Tab. 1. All conclusions from results on the validation set of these datasets as discussed in the main paper are applicable here. On the ScanNet benchmark, ODIN achieves close to SOTA performance on semantic segmentation and mAP25 metric of Instance Segmentation while being significantly worse on mAP metric due to misalignments between sensor and mesh sampled point clouds. On ScanNet200 benchmark, ODIN sets a new SOTA on semantic segmentation and mAP50/mAP25 metric of Instance Segmentation, while achieving close to SOTA performance on mAP metric. Notably ODIN is the first method that operates over sensor RGB-D data for instance segmentation and achieves competitive performance to models operating over mesh-sampled point clouds.

1.2. Evaluation on S3DIS and Matterport3D

We also benchmark ODIN on Matterport3D [2] and S3DIS [1] datasets.

Matterport: Matterport3D comprises 90 building-scale scenes, further divided into individual rooms, with 1554 training rooms and 234 validation rooms. The dataset provides a mapping from each room to the camera IDs that captured images for that room. After discarding 158 training rooms and 18 validation rooms without a valid camera mapping, we are left with 1396 training rooms and 158 validation rooms. For instance segmentation results, we train the state-of-the-art Mask3D [20] model on the same data (reduced set after discarding invalid rooms). For semantic segmentation, we conduct training and testing on the reduced set, while baseline numbers are taken from the OpenScene [16] paper, trained and tested on the original data. Given the small size of the discarded data, we do not anticipate significant performance differences. The official benchmark of Matterport3D tests on 21 classes; however, OpenScene also evaluates on 160 classes to compare with state-of-the-art models on long-tail distributions. We follow them and report results in both settings.

S3DIS: S3DIS comprises 6 building-scale scenes, typically divided into 5 for training and 1 for testing. The dataset provides raw RGB-D images, captured panorama images, and images rendered from the mesh obtained after reconstructing the original sensor data. Unlike Matterport3D, S3DIS do not provide undistorted raw images; thus, we use the provided rendered RGB-D images. Some rooms in S3DIS have major misalignments between RGB-D images and point clouds, which we partially address by incorporating fixes from DeepViewAgg [17] and introducing our own adjustments. Despite these fixes, certain scenes still exhibit significantly low overlap between RGB-D images and the provided mesh-sampled point cloud. To mitigate this, we query images from other rooms and verify their overlap with the provided point cloud for a room. This partially helps in addressing the low overlap issue.

The official S3DIS benchmark evaluates 13 classes. Due to the dataset's small size, some models pre-train on additional datasets like ScanNet, as seen in SoftGroup [22], and on Structured3D datasets [27], consisting of 21,835 rooms, as done by Swin3D-L [25]. Similar to Mask3D [20], we report results in both settings of training from scratch and starting from weights trained on ScanNet.

Like ScanNet and ScanNet200, both S3DIS and Matter-

Table 1. **Evaluation on Test Set of Established 3D Benchmarks.**

(a) **Comparison on ScanNet for Instance Segmentation Task.**

Input	Model	mAP	mAP50	mAP25
Sensor RGBD Point Cloud	ODIN-Swin-B (Ours)	**47.7**	**71.2**	**86.2**
Mesh Sampled Point Cloud	SoftGroup [22]	50.4	76.1	86.5
	PBNet [26]	57.3	74.7	82.5
	Mask3D [20]	56.6	78.0	87.0
	QueryFormer [15]	58.3	**78.7**	**87.4**
	MAFT [14]	**59.6**	78.6	86.0

(b) **Comparison on ScanNet for Semantic Segmentation Task.**

Input	Model	mIoU
Sensor RGBD Point Cloud	MVPNet [9]	64.1
	BPNet [7]	**74.9**
	DeepViewAgg [17]	-
	ODIN-Swin-B (Ours)	74.4
Rendered RGBD Point Cloud	VMVF [12]	**74.6**
Mesh Sampled Point Cloud	Point Transformer v2 [24]	75.2
	Stratified Transformer [13]	74.7
	OctFormer [23]	76.6
	Swin3D-L [25]	**77.9**
Zero-Shot	OpenScene [16]	-

(c) **Comparison on ScanNet200 for Instance Segmentation Task.**

	Model	mAP	mAP50	mAP25
Sensor RGBD Point Cloud	ODIN-Swin-B (Ours)	**27.2**	**39.4**	**47.5**
Mesh Sampled Point Cloud	Mask3D [20]	**27.8**	**38.8**	**44.5**
	QueryFormer [15]	-	-	-
	MAFT [14]	-	-	-
Zero-Shot	OpenMask3D [21]	-	-	-

(d) **Comparison on ScanNet200 for Semantic Segmentation Task.**

Input	Model	mIoU
Sensor RGBD Point Cloud	ODIN-Swin-B (Ours)	**36.8**
Mesh Sampled Point Cloud	LGround [18]	27.2
	CeCo [28]	**34.0**
	Octformer [23]	32.6

port3D undergo post-processing of collected RGB-D data to construct a mesh, from which a point cloud is sampled and labeled. Hence, we train both Mask3D [20] and our model using RGB-D sensor point cloud data and evaluate on the benchmark-provided point cloud. Additionally, we explore model variants by training and testing them on the mesh-sampled point cloud for comparative analysis.

The results are shown in Tab. 2. We draw the following conclusions:

ODIN outperforms SOTA 3D models on Matterport3D Instance Segmentation Benchmark across all settings (Tab. 2a)

ODIN sets a new state-of-the-art on Matterport3D Semantic Segmentation Benchmark (Tab. 2b): Our model achieves superior performance in both the 21 and 160 class settings. It also largely outperforms OpenScene [16] on both settings. OpenScene is a zero-shot method while ODIN is supervised in-domain, making this comparison unfair. However, OpenScene notes that their zero-shot model outperforms fully-supervised models in 160 class setup as their model is robust to rare classes while the supervised models can severely suffer in segmenting long-tail. ConceptFusion [10], another open-vocabulary 3D segmentation model, also draws a similar conclusion. With this result, we point to a possibility of supervising in 3D while also being robust to long-tail by simply utilizing the strong 2D pre-trained weight initialization.

On S3DIS Instance Segmentation Benchmark (Tab. 2c), in the setup where baseline Mask3D start from ScanNet pre-trained checkpoint, our model outperforms them in the RGBD point cloud setup but obtains lower performance compared to mesh sampled point cloud methods and when compared on the setup where all models train from scratch.

On S3DIS Semantic Segmentation Benchmark (Tab. 2d, ODIN trained with ScanNet weight initialization outperforms all RGBD point cloud based methods, while achieving competitive performance on mesh sampled point cloud. When trained from scratch, it is much worse than other baselines. Given the limited dataset size of S3DIS with only 200 training scenes, we observe severe overfitting.

1.3. ScanNet200 Detailed Results

ScanNet200 [18] categorizes its 200 object classes into three groups—*Head*, *Common*, and *Tail*—each comprising 66, 68, and 66 categories, respectively. In Tab. 3, we provide a detailed breakdown of the ScanNet200 results across these splits. We observe that in comparison to SOTA Mask3D model trained on mesh-sampled point cloud, ODIN achieves lower performance on *Head* classes, while significantly better performance on *Common* and *Tail*

Table 2. **Evaluation on Matterport3D [2] and S3DIS [1] datasets.**

(a) **Comparison on Matterport3D for Instance Segmentation Task.**

Input	Model	21		160	
		mAP	mAP25	mAP	mAP25
Sensor RGBD Point Cloud	Mask3D [20]	7.2	16.8	2.5	10.9
	ODIN-ResNet50 (Ours)	22.5	56.4	11.5	27.6
	ODIN-Swin-B (Ours)	**24.7**	**63.8**	**14.5**	**36.8**
Mesh Sampled Point Cloud	Mask3D [20]	**22.9**	**55.9**	**11.3**	**23.9**

(b) **Comparison on Matterport3D for Semantic Segmentation Task.**

Input	Model	21		160	
		mIoU	mAcc	mIoU	mAcc
Sensor RGBD Point Cloud	ODIN-ResNet50 (Ours)	54.5	65.8	22.4	28.5
	ODIN-Swin-B (Ours)	**57.3**	**69.4**	**28.6**	**38.2**
Mesh Sampled Point Cloud	TextureNet [8]	-	63.0	-	-
	DCM-Net [19]	-	**67.2**	-	-
	MinkowskiNet [4]	**54.2**	64.6	-	**18.4**
Zero-Shot	OpenScene [16]	**42.6**	**59.2**	-	**23.1**

(c) **Comparison on S3DIS Area5 for Instance Segmentation Task.** († = uses additional data)

	Model	mAP	mAP50	mAP25
RGBD Point Cloud	Mask3D [20]	40.7	54.6	64.2
	Mask3D [20] †	41.3	55.9	66.1
	ODIN-ResNet50 (Ours)	36.3	48.0	61.2
	ODIN-ResNet50 † (Ours)	**44.7**	**57.7**	67.5
	ODIN-Swin-B † (Ours)	43.0	56.4	**70.0**
Mesh Sampled Point Cloud	SoftGroup [22] †	51.6	66.1	-
	Mask3D [20]	56.6	68.4	75.2
	Mask3D [20] †	**57.8**	**71.9**	**77.2**
	QueryFormer [15]	57.7	69.9	-
	MAFT [14]	-	69.1	75.7

(d) **Comparison on S3DIS for Semantic Segmentation Task.** († = uses additional data)

Input	Model	mIoU
RGBD Point Cloud	MVPNet [9]	62.4
	VMVF [12]	65.4
	DeepViewAgg [17]	67.2
	ODIN-ResNet50 (Ours)	59.7
	ODIN-ResNet50 † (Ours)	66.8
	ODIN-Swin-B † (Ours)	**68.6**
Mesh Sampled Point Cloud	Point Transformer v2 [24]	71.6
	Stratified Transformer [13]	72.0
	Swin3D-L [25] †	**74.5**

Table 3. **Detailed ScanNet200 results for Instance Segmentation** (§ = trained by us using official codebase)

Input	Model	All			Head			Common			Tail		
		mAP	mAP50	mAP25	mAP	mAP50	mAP25	mAP	mAP50	mAP25	mAP	mAP50	mAP25
Sensor RGBD point cloud	Mask3D § [20]	15.5	21.4	24.3	21.9	31.4	37.1	13.0	17.2	18.9	7.9	10.3	11.5
	ODIN-ResNet50 (Ours)	25.6	36.9	43.8	34.8	51.1	63.9	23.4	33.4	37.9	17.8	24.9	28.1
	ODIN-Swin-B (Ours)	**31.5**	**45.3**	**53.1**	37.5	54.2	**66.1**	**31.6**	**43.9**	**50.2**	**24.1**	**36.6**	**41.2**
Mesh Sampled point cloud	Mask3D [20]	27.4	37.0	42.3	**40.3**	**55.0**	62.2	22.4	30.6	35.4	18.2	23.2	27.0

classes. This highlights the contribution of effectively utilizing 2D pre-trained features, particularly in detecting a long tail of class distribution where limited 3D data is available.

1.4. Variation of Performance with Number of Views

We examine the influence of the number of views on segmentation performance using the AI2THOR dataset, specifically focusing on the 2D mAP performance metric. The evaluation is conducted by varying the number of *context* images surrounding a given *query* RGB image. Starting from a single-view without any context (N=0), we increment N to 5, 10, 20, 40, 60, and finally consider all images in the scene as context. ODIN takes these $N + 1$ RGB-D images as input, predicts per-pixel instance segmentation for each image, and assesses the 2D mAP performance on the *query* image. The results, depicted in Fig. 1, show a continuous increase in 2D mAP with the growing number of views. This observation underscores the advantage of utilizing multiview RGB-D images over single-view RGB images whenever feasible.

1.5. Inference Time

We assess the inference time of Mask3D and ODIN by averaging the forward pass time of each model across the entire validation set, utilizing a 40 GB VRAM A100. When fed the mesh-sampled point cloud directly, Mask3D achieves an inference time of 228ms. When provided with the sensor point cloud as input, the inference time increases to 864 ms. Mask3D with sensor point cloud is slower than with mesh point cloud because at the same voxel size (0.02m), more voxels are occupied in sensor point cloud (110k on avg.) compared to mesh point clouds (64k on avg.) as mesh-cleaning sometimes discards large portion of the scene. The transfer of features from the sensor point cloud to the mesh point cloud adds an extra 7 ms. ODIN-SwinB, which operates over the sensor point cloud, has an inference time of

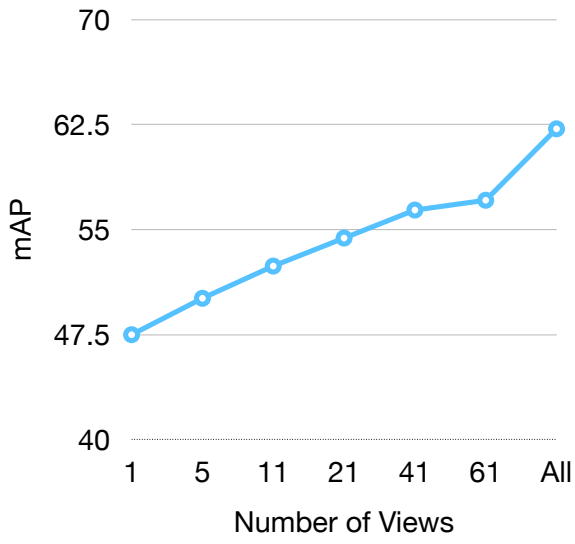

Figure 1. **2D mAP Performance Variation with increasing number of context views used**

960ms.

2. Additional Implementation Details

The detailed components of our architecture and their descriptions are presented in Fig. 2.

More implementation details are presented below:

Augmentations: For RGB image augmentation, we implement the Large Scale Jittering Augmentation method from Mask2Former [3], resizing images to a scale between 0.1 and 2.0. We adjust intrinsics accordingly post-augmentation and apply color jittering to RGB images. Training involves a consecutive set of N images, typically set to 25. With a 50% probability, we randomly sample k images from the range $[1, N]$ instead of using all N images. Additionally, instead of consistently sampling N consecutive images, we randomly skip k images in between, where k ranges from 1 to 4.

For 3D augmentations, we adopt the Mask3D [20] approach, applying random 3D rotation, scaling, and jitter noise to the unprojected XYZs. Elastic distortion and random flipping augmentations from Mask3D are omitted due to a slight drop in performance observed in our initial experiments.

Image Resolutions We use a resolution of 256×256 for ScanNet, 512×512 for ScanNet200, and AI2THOR. In our AI2THOR experiments, we discovered that employing higher image resolutions enhances the detection of smaller objects, with no noticeable impact on the detection of larger ScanNet-like objects. This observation was confirmed in ScanNet, where we experimented with 512×512 image resolutions and did not observe any discernible benefit.

Interpolation Throughout our model, interpolations are employed in various instances, such as when upsampling the feature map from $1/8$th resolution to $1/4$th. In cases involving depth, we unproject feature maps to 3D and perform trilinear interpolation, as opposed to directly applying bilinear interpolation on the 2D feature maps. For upsampling/downsampling the depth maps, we use the nearest interpolation. Trilinear interpolation proves crucial for obtaining accurate feature maps, particularly at 2D object boundaries like table and floor edges. This is because nearest depth interpolation may capture depth from either the table or the floor. Utilizing trilinear upsampling of feature maps ensures that if the upsampled depth is derived from the floor, it interpolates features from floor points rather than table points.

Use of Segments: Some datasets, such as ScanNet and ScanNet200, provide supervoxelization of the point cloud, commonly referred to as *segments*. Rather than directly segmenting all input points, many 3D methods predict outputs over these segments. Specifically, Mask3D [20] featurizes the input points and then conducts mean pooling over the features of points belonging to a segment, resulting in one feature per segment. Following prior work, we also leverage segments in a similar manner. We observe that utilizing segments is crucial for achieving good mAP performance, while it has no discernible impact on mAP25 performance. We suspect that this phenomenon may arise from the annotation process of these datasets. Humans were tasked with labelling segments rather than individual points, ensuring that all points within a segment share the same label. Utilizing segments with our models guarantees that the entire segment is labelled with the same class. It's worth noting that in AI2THOR, our method and the baselines do not utilize these segments, as they are not available.

Post-hoc output transfer vs feature transfer: ODIN takes the sensor point cloud as input and generates segmentation output on the benchmark-provided point cloud. In this process, we featurize the sensor point cloud and transfer these features from the sensor point cloud to the benchmark-provided point cloud. Subsequently, we predict segmentation outputs on this benchmark-provided feature cloud and supervise the model with the labels provided in the dataset. An alternative approach involves segmenting and supervising the sensor RGB-D point cloud and later transferring the segmentation output to the benchmark point cloud for evaluation. We experimented with both strategies and found them to yield similar results. However, as many datasets provide segmentation outputs only on the point cloud, transferring labels to RGB-D images for the latter strategy requires careful consideration. This is due to the sparser nature of the provided point cloud compared to the RGB-D sensor point cloud, and factors such as depth

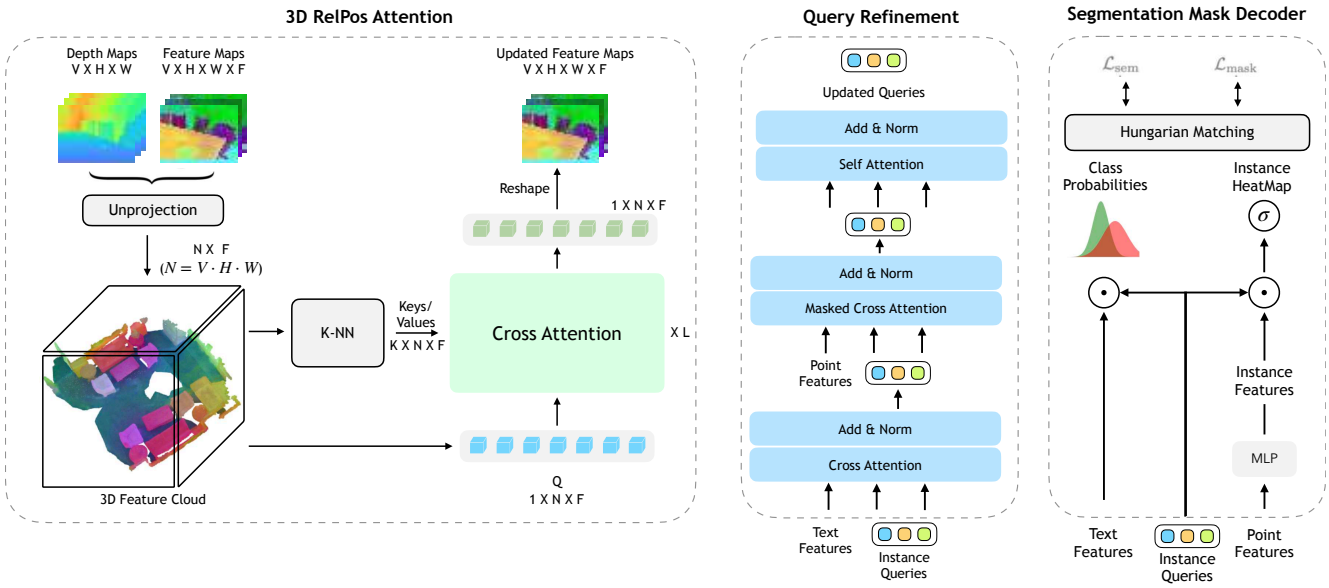

Figure 2. **Detailed ODIN Architecture Components**: On the **Left** is the 3D RelPos Attention module which takes as input the depth, camera parameters and feature maps from all views, lifts the features to 3D to get 3D tokens. Each 3D token serves as a query. The K-Nearest Neighbors of each 3D token become the corresponding keys and values. The 3D tokens attend to their neighbours for L layers and update themselves. Finally, the 3D tokens are mapped back to the 2D feature map by simply reshaping the 3D feature cloud to 2D multi-view feature maps. On the **Middle** is the query refinement block where queries first attend to the text tokens, then to the visual tokens and finally undergo self-attention. The text features are optional and are only used in the open-vocabulary decoder setup. On the **Right** is the segmentation mask decoder head where the queries simply perform a dot-product with visual tokens to decode the segmentation heatmap, which can be thresholded to obtain the segmentation mask. In the Open-Vocabulary decoding setup, the queries also perform a dot-product with text tokens to decode a distribution over individual words. In a closed vocabulary decoding setup, queries simply pass through an MLP to predict a distribution over classes.

noise and misalignments can contribute to low-quality label transfer. Consequently, we opt for the former strategy in all our experiments.

Depth Hole-Infilling: The sensor-collected depth maps usually have holes around object boundaries and shiny/transparent surfaces. We perform simple OpenCV depth inpainting to fill these holes. We tried using neural-based depth completion methods and NERF depth-inpainting but did not observe significant benefits.

AI2THOR Data Collection: AI2THOR [11] is an embodied simulator where an agent can navigate within a house, execute actions, and capture RGB-D images of the scene. We load the structurally generated houses from ProcTHOR [6] into the AI2THOR simulator, and place an agent randomly at a navigable point provided by the simulator. The agent performs a single random rotation around its initial location and captures an RGB-D frame. This process is repeated, with the agent spawning at another random location, until either all navigable points are exhausted or a maximum of $N = 120$ frames is collected. While ProcTHOR offers 10,000 scenes, we randomly select only 1,500 scenes to match the size of ScanNet. Additionally, we retain scenes with fewer than 100 objects, as our model utilizes a maximum of 100 object queries.

3. Qualitative Results

Fig. 3 shows qualitative visualizations of ODIN for various 3D and 2D datasets.

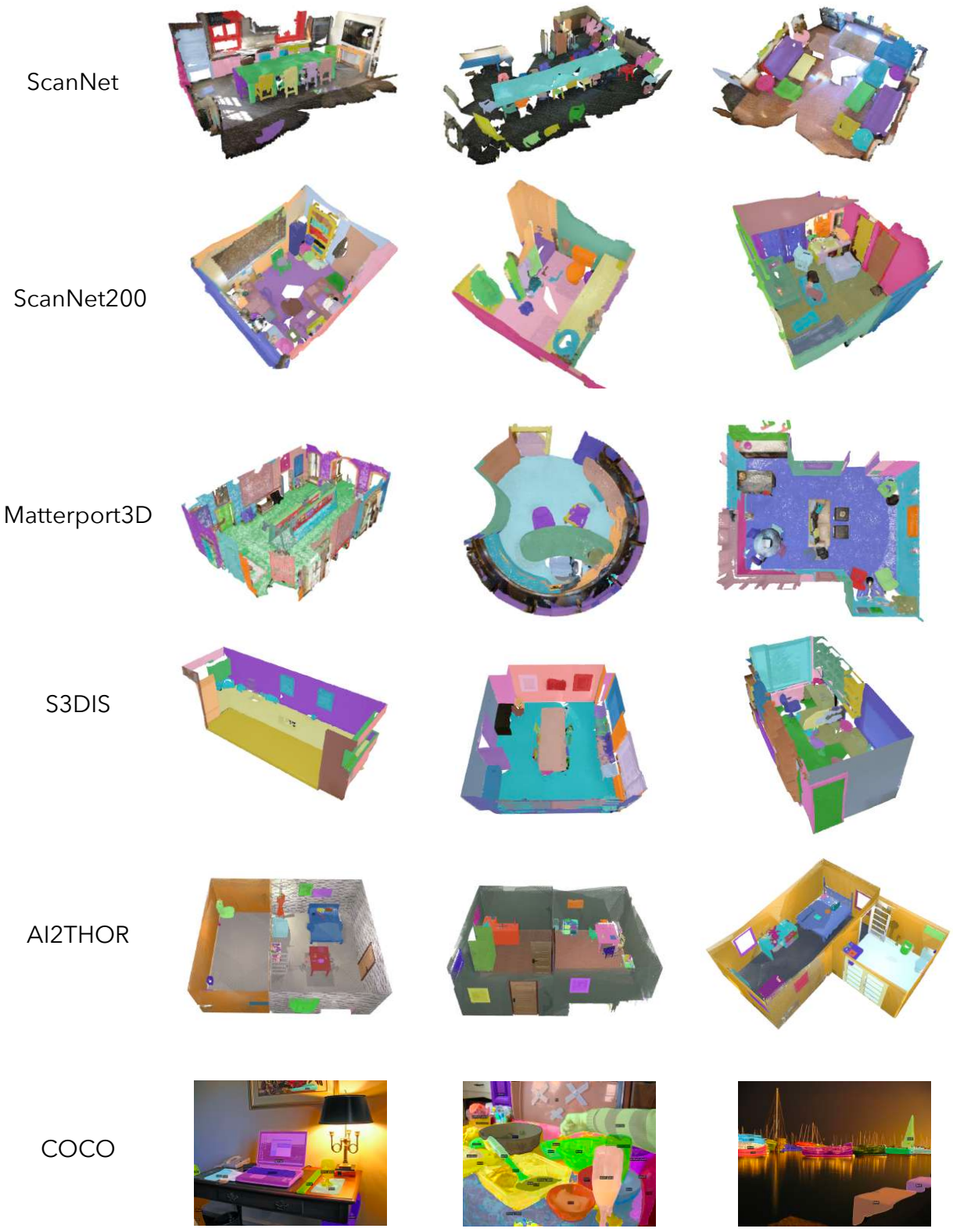

Figure 3. Qualitative Results on various 3D and 2D datasets

References

[1] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 1, 3

[2] 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. *arXiv preprint arXiv:1709.06158*, 2017. 1, 3

[3] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. 2022. 4

[4] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3075–3084, 2019. 3

[5] 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. 1

[6] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Procthor: Large-scale embodied ai using procedural generation. *Advances in Neural Information Processing Systems*, 35:5982–5994, 2022. 5

[7] Wenbo Hu, Hengshuang Zhao, Li Jiang, Jiaya Jia, and Tien-Tsin Wong. Bidirectional projection network for cross dimension scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14373–14382, 2021. 2

[8] Jingwei Huang, Haotian Zhang, Li Yi, Thomas Funkhouser, Matthias Nießner, and Leonidas J Guibas. Texturenet: Consistent local parametrizations for learning from high-resolution signals on meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4440–4449, 2019. 3

[9] Maximilian Jaritz, Jiayuan Gu, and Hao Su. Multi-view pointnet for 3d scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 2, 3

[10] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241*, 2023. 2

[11] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017. 5

[12] Abhijit Kundu, Xiaoqi Yin, Alireza Fathi, David Ross, Brian Brewington, Thomas Funkhouser, and Caroline Pantofaru. Virtual multi-view fusion for 3d semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 518–535. Springer, 2020. 2, 3

[13] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8500–8509, 2022. 1, 2, 3

[14] Xin Lai, Yuhui Yuan, Ruihang Chu, Yukang Chen, Han Hu, and Jiaya Jia. Mask-attention-free transformer for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3693–3703, 2023. 2, 3

[15] Jiahao Lu, Jiacheng Deng, Chuxin Wang, Jianfeng He, and Tianzhu Zhang. Query refinement transformer for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18516–18526, 2023. 2, 3

[16] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 815–824, 2023. 1, 2, 3

[17] Damien Robert, Bruno Vallet, and Loic Landrieu. Learning multi-view aggregation in the wild for large-scale 3d semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5575–5584, 2022. 1, 2, 3

[18] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *European Conference on Computer Vision*, pages 125–141. Springer, 2022. 1, 2

[19] Jonas Schult, Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. Dualconvmesh-net: Joint geodesic and euclidean convolutions on 3d meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8612–8622, 2020. 3

[20] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8216–8223. IEEE, 2023. 1, 2, 3, 4

[21] Ayça Takmaz, Elisabetta Fedele, Robert W Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. Openmask3d: Open-vocabulary 3d instance segmentation. *arXiv preprint arXiv:2306.13631*, 2023. 2

[22] Thang Vu, Kookhoi Kim, Tung M Luu, Thanh Nguyen, and Chang D Yoo. Softgroup for 3d instance segmentation on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2708–2717, 2022. 1, 2, 3

[23] Peng-Shuai Wang. Octformer: Octree-based transformers for 3d point clouds. *arXiv preprint arXiv:2305.03045*, 2023. 1, 2

[24] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems*, 35:33330–33342, 2022. 1, 2, 3

[25] Yu-Qi Yang, Yu-Xiao Guo, Jian-Yu Xiong, Yang Liu, Hao Pan, Peng-Shuai Wang, Xin Tong, and Baining Guo. Swin3d: A pretrained transformer backbone for 3d indoor scene understanding. *arXiv preprint arXiv:2304.06906*, 2023. 1, 2, 3

[26] Weiguang Zhao, Yuyao Yan, Chaolong Yang, Jianan Ye, Xi Yang, and Kaizhu Huang. Divide and conquer: 3d point cloud instance segmentation with point-wise binarization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 562–571, 2023. 2

[27] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 519–535. Springer, 2020. 1

[28] Zhisheng Zhong, Jiequan Cui, Yibo Yang, Xiaoyang Wu, Xiaojuan Qi, Xiangyu Zhang, and Jiaya Jia. Understanding imbalanced semantic segmentation through neural collapse. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19550–19560, 2023. 2