Supplementary Materials for
OA-CNNs: Omni-Adaptive Sparse CNNs for 3D Semantic Segmentation

Bohao Peng[1] Xiaoyang Wu[2] Li Jiang[3] Yukang Chen[1]
Hengshuang Zhao[2] Zhuotao Tian[4] Jiaya Jia[1]
[1]CUHK [2]HKU [3]CUHK, Shenzhen [4]HIT, Shenzhen

Appendix

1. Implementation Details

In this section, we present further details and configurations utilized in our experiments.

1.1. Environment

Experimental environment.

- PyTorch version: 1.10.1
- CUDA version: 11.1
- cuDNN version: 1.10.1
- GPU: Nvidia RTX 3090 × 4

1.2. Data Propocessing

Data preprocessing and augmentation. This work maintains consistency in data preprocessing and augmentation with PTv1 and Ptv2 [9, 10] for the ScanNet series and S3DIS datasets [1, 3, 8]. The specific data augmentation strategies employed during training are outlined in Tab. 1.

	Drop	Rotate	Scale	Flip	Jitter	Disort	Chromatic
ScanNet v2 [3]	✓	✓	✓	✓	✓	✓	✓
ScanNet200 [8]	✓	✓	✓	✓	✓	✓	✓
S3DIS [1]			✓	✓	✓		✓

Table 1. Data augmentation strategies on various datasets.

Voxelization.

- voxel size: 0.02m
- hash type: Fowler-Noll-Vo (FNV)

1.3. Training Setting

This subsection offers additional details on our training settings for the three standard benchmarks, including optimizer and learning configurations. More details are listed in Tab. 2.

2. Experimental Results

2.1. Test Benchmarks

In this section, we present detailed results for each category on the ScanNet v2 and ScanNet200 test set. For more detailed information refer to the official benchmarks [3, 8].

	Epoch	LR	Weight Decay	Scheduler	Optimizer	Batch Size
ScanNet v2 [3]	600	1e-3	0.02	Cosine	AdamW	16
ScanNet200 [8]	900	1e-3	0.02	Cosine	AdamW	12
S3DIS [1]	3000	1e-3	0.05	MultiStep	AdamW	16

Table 2. Training settings on various datasets.

ScanNet v2 contains over $1,513$ RGB-D indoor scans of various environments, including apartments, offices, and public spaces. The dataset includes high-quality 3D point clouds with per-point semantic annotations. On the other hand, the ScanNet200 benchmark extends the class categories to 200, an order of magnitude more than the previous, significantly increasing the difficulty and generalizability requirements. Moreover, ScanNet200 partitioned the 200 categories into three distinct subsets based on the labeled surface points' frequency in the train set: head, common, and tail, comprising 66, 68, and 66 categories, respectively, for a more granular understanding of the segmentation performance. As for the evaluation, we follow the standard protocol using the mean class-wise intersection over union (mIoU) for both ScanNet v2 and ScanNet200.

Specifically, Tab. 3 presents comprehensive results on the ScanNet v2, offering a detailed breakdown of the performance for each semantic class. Similarly, Tab. 4 provides the results of the head, common, and tail subsets on the ScanNet200 benchmark, offering a more nuanced understanding of the performance across different levels of class imbalance. Furthermore, Fig. 1 visually represents the segmentation performance for each specific class in the ScanNet200 benchmark.

Category	**AVG**	bathtub	bed	bookshelf	cabinet	chair	counter	curtain	desk	door	shower curtain
mIoU (%)	**75.6**	78.3	82.6	85.8	77.6	83.7	54.8	89.6	64.9	67.5	80.2
Category	picture	floor	refrigerator	sink	sofa	table	toilet	wall	window	otherfurniture	
mIoU (%)	33.5	96.2	77.1	77.0	78.7	69.1	93.6	88.0	76.1	58.6	

Table 3. Results for each category on the ScanNet v2 test benchmark.

Set	Head	Common	Tail	**All**
mIoU (%)	55.8	26.9	12.4	**33.3**

Table 4. Results for various sets on the ScanNet200 test benchmark.

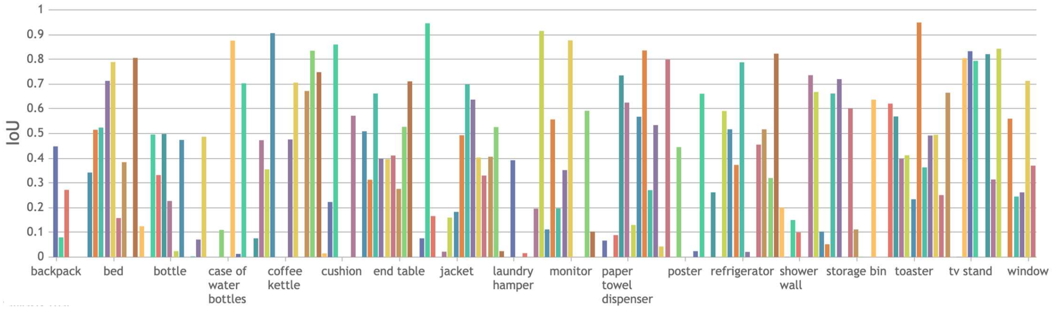


Figure 1. Results for each category on the ScanNet200 test benchmark.

2.2. Raw Points and Structural Voxels

The Point Transformer methods, building upon the fundamental principles of the PointNet series [6, 7], emphasize the advantages of operating directly on raw point data to capture finer-grained local features and preserve the underlying geometric structure of the data. In contrast, traditional CNN-based methods typically require voxelization preprocessing, which involves partitioning the 3D space into a regular grid of equally-sized cubic volumes (voxels). This mapping allows the points' positions to be transformed into discrete indices [2, 4], which can be used for convolutional and index retrieval operations.

However, voxelization may result in losing fine-grained geometric details and potential aliasing effects. To test the influence of voxelization on performance, we conducted an experiment where we input the discretized voxels into the Point Transformer with normalized indices instead of the original positional information while keeping all other configurations the same. The voxelization used in this experiment was the same as for our OA-CNNs' input. The results are shown in Tab. 5, and we observed that the degradation in performance due to discretization was acceptable with appropriate granularity.

Method	Input	mIoU (%)	Input	Size	Hash	mIoU (%)
PointTransformer v2 [9]	Point	**75.6**	Voxel	0.02m	FNV	**75.5**

Table 5. Comparison between point and voxel inputs.

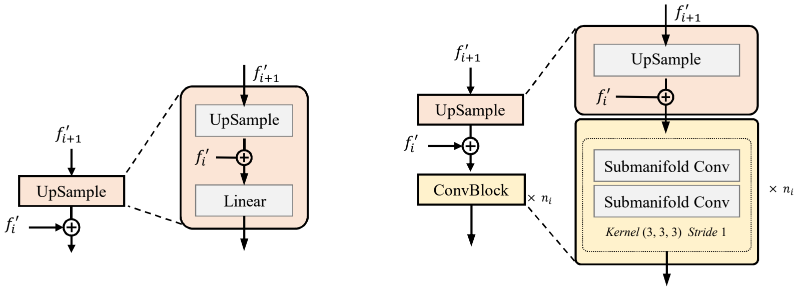
2.3. Decoder Design

Typically, U-Net architectures are adopted by 3D semantic segmentation models, which split the entire process into feature encoding and decoding. The encoder processes the input point cloud features and generates downsampled pyramid features using multi-scale and multi-revolution techniques, while the decoder integrates all the cues. Previous 3D semantic models have constructed decoder blocks using the same components, replacing the downsample sparse modules with upsample modules. In this study, we have constructed our decoder blocks with only essential upsample modules and a single MLP layer, resulting in an extremely lightweight and simple design. Additionally, we have transferred the main components to the encoder section, ensuring the lightweight decoder's effectiveness.

To be specific, our initial model construction adhered to the typical pipeline, which involves constructing the decoder in a manner similar to the encoder, while replacing the downsample modules with upsample modules for the basic blocks. Subsequently, we designed the decoder block to comprise only a single upsample and MLP layer. The experimental results are shown in Tab. 6 and more detailed architectural comparison is displayed in Fig. 2.

Method	Encoder Blocks	Decoder Blocks	mIoU (%)
Basic Blocks (upsample)	[2, 2, 6, 6]	[2, 2, 2, 2]	75.0
MLP	[3, 3, 9, 8]	-	76.1

Table 6. Performance comparison between different decoder designs.



(a) Our Decoder Block

(b) Typical Decoder Block

Figure 2. Comparison between our and typical decoder blocks.

2.4. Decoder Design

Typically, U-Net architectures are adopted by 3D semantic segmentation models, which split the entire process into feature encoding and decoding. The encoder processes the input point cloud features and generates downsampled pyramid features using multi-scale and multi-revolution techniques, while the decoder integrates all the cues. Previous 3D semantic models have constructed decoder blocks using the same components, replacing the downsample sparse modules with upsample modules. In this study, we have constructed our decoder blocks with only essential upsample modules and a single MLP layer, resulting in an extremely lightweight and simple design. Additionally, we have transferred the main components to the encoder section, ensuring the lightweight decoder's effectiveness.

To be specific, our initial model construction adhered to the typical pipeline, which involves constructing the decoder in a manner similar to the encoder, while replacing the downsample modules with upsample modules for the basic blocks. Subsequently, we designed the decoder block to comprise only a single upsample and MLP layer. The experimental results are shown in Tab. 6 and more detailed architectural comparison is displayed in Fig. 2.

3. Impact of the grid size.

To examine the impact of grid size, we supplement ablation experiments by adjusting the grid size to *0.5x, 0.67x, 0.75x, and 1.25x* times compared to the original setting. The experimental results are shown in Fig. 3, which shows that: (1) Significantly reducing the grid size leads to notable performance degradation, attributed to insufficient receptive range. (2) Continuing to expand the grid size does not yield improvements and may even cause minor negative impacts. This could be because fine-grained local details are overwhelmed by the surrounding context, especially for small objects. The time consumption generally remains consistent across different grid sizes, which shows the robustness of our method.

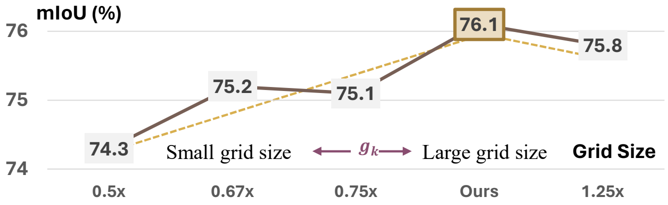


Figure 3. Comparative analysis of the impact of various grid sizes.

4. Visualization Studies

4.1. Receptive Fields Comparison

In this subsection, we present the Effective Receptive Field (ERF) [5] visualization for the feature of interest in the first stage, denoted by red and yellow stars representing the table and wall, respectively. Effective Receptive Field (ERF) is used to measure the ability of a deep neural network to capture the contextual information of an input image or feature map. The ERF of a neuron in a deep network is defined as the area in the input space that influences the neuron's activation, which helps to explain the network's behavior and performance. We conducted ablation experiments to assess the effectiveness of our proposed ARConv and adaptive aggregator on distinct 3D scene parts with different spatial structures and appearances. The visualization results are shown in Fig. 4.

The experimental results demonstrate that our proposed ARConv can significantly expand the receptive range compared to the baseline. Moreover, the adaptive aggregator can dynamically adjust the receptive fields based on the specific geometric and appearance features, allocating a larger receptive field for the wall and a smaller one for the table. These findings suggest that our proposed methods can effectively capture the key features of different parts of the 3D scene and improve the model's overall performance on 3D point cloud tasks.

4.2. Prediction Visualization

In this subsection, we provide additional visualizations of our proposed model's predictions on the ScanNet dataset. Fig. 5 showcases a diverse set of indoor scenes to demonstrate our model's performance across different environments. The visualizations demonstrate that our model performs remarkably well in various indoor scenes, regardless of complexity and structural variations. Specifically, the model accurately segments different indoor objects such as furniture, walls, and floors, and effectively captures their fine details and shapes. Furthermore, the model generates consistent and coherent predictions even in complex indoor environments, where objects are densely packed and occluded.

These visualizations provide compelling evidence of the effectiveness of our proposed approach in achieving accurate and robust 3D semantic segmentation results on the ScanNet dataset.

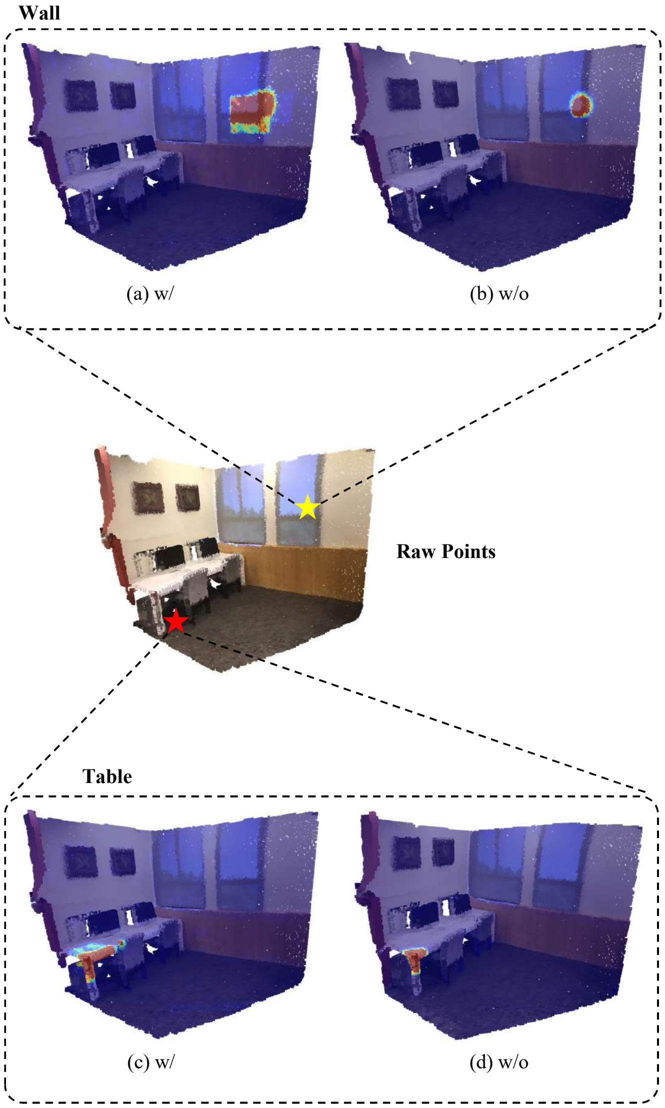


Figure 4. Visualization comparison of the receptive fields on various 3D scene parts.

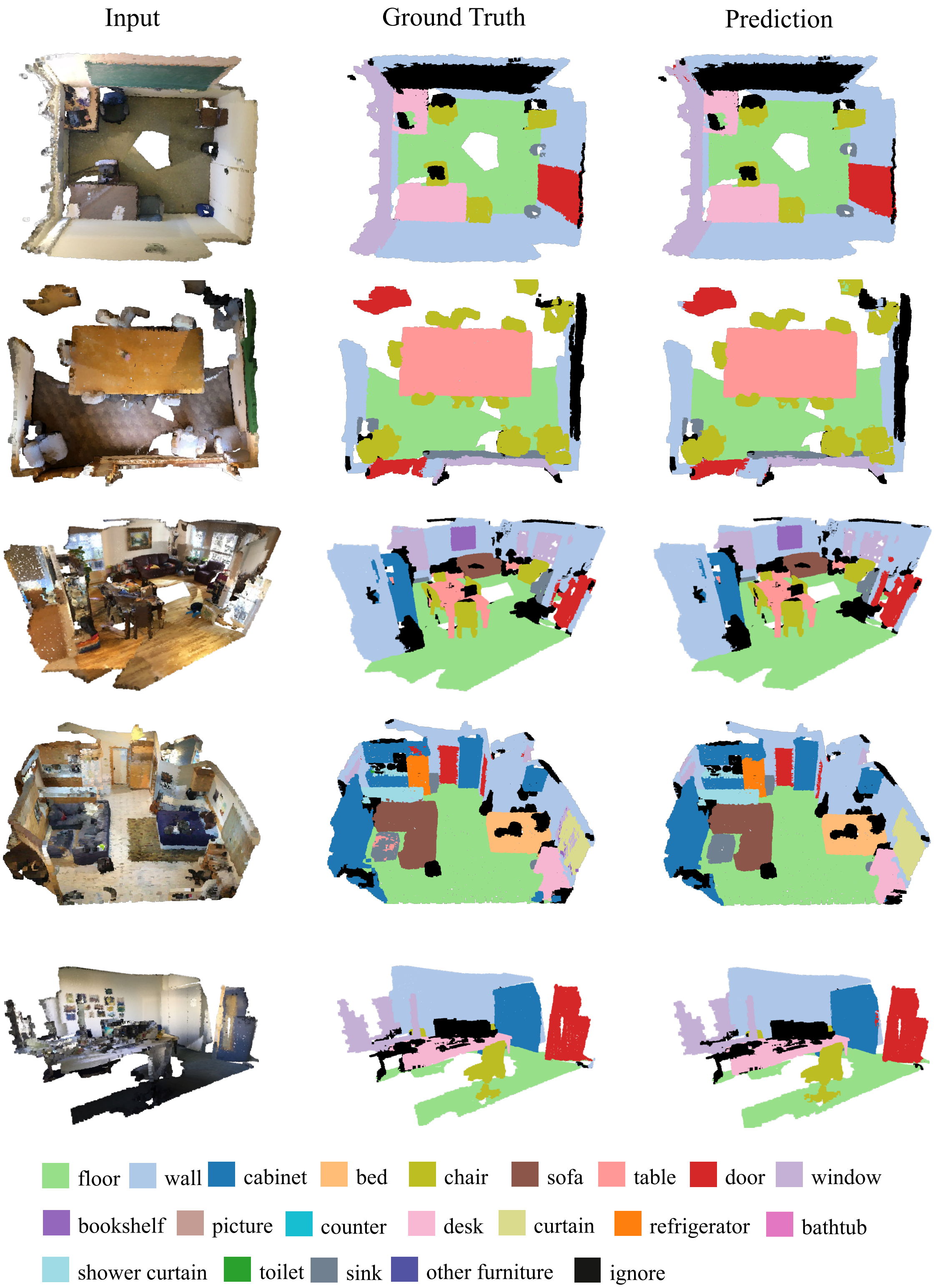


Figure 5. Visualization results of the raw point cloud, ground truth, and our model's prediction.

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1534–1543, 2016. 1, 2

[2] 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

[3] 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, 2

[4] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9224–9232, 2018. 3

[5] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems*, 29, 2016. 4

[6] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 3

[7] 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. 3

[8] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 1, 2

[9] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *arXiv preprint arXiv:2210.05666*, 2022. 1, 3

[10] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 1