Mosaic3D: Foundation Dataset and Model for Open-Vocabulary 3D Segmentation

Supplementary Material

Contents

A. Dataset Details

Below we report the data statistics of our Mosaic3D-5.6M dataset, detail the data preprocessing steps, pipeline configurations used for each dataset in our experiments, and additional data pipeline experiments that utilize 3D instance mask predictions in caption generation process.

A.1. Data Statistics

In Tab. A1, we report the statistics of our generated dataset, including the number of scenes, RGB-D frames, generated captions, and total tokens in captions for each source dataset. Our dataset contains over 30K scenes, and 5.6M captions with a total of 30M tokens across both real and synthetic indoor environments.

Dataset	# Scenes	# Frames	# Captions	# Tokens	Category
ScanNet [24]	1,513	2.5M	1.3M	7.2M	Real
Matterport3D [14]	2,194	0.2M	0.7M	3.8M	Real
ARKitScenes [7]	5,045	4.0M	2.4M	12.6M	Real
ScanNet++ [99]	380	0.2M	0.2M	0.9M	Real
Structured3D [107]	20,065	0.2M	1.0M	5.4M	Synthetic
Total	29,197	7.1M	5.6M	29.9M	

Table A1. **Statistics of our generated dataset.** We report the number of scenes, RGB-D frames, generated captions, and total tokens in captions for each source dataset.

In Tab. A2, we evaluate caption and 3D mask quality across datasets using three metrics. The *unique normalized nouns count* measures the total number of unique normalized nouns in captions, with higher count indicating richer and more diverse captions. *Mask coverage* (%) calculates the mean percentage of 3D points with associated captions per scene, where higher coverage enables more effective training. *Mask entropy* (bits) measures mask quality for datasets with partial masks generated from multi-view images (*i.e.* OV3D, RegionPLC, and Mosaic3D-5.6M) without using GT. It calculates Shannon entropy of GT instance ID distributions within each mask–higher entropy indicates that a mask contains multiple GT instances, suggesting less accurate mask boundaries. Mosaic3D-5.6M demonstrates superior caption diversity and mask quality compared to both existing large-scale 3D-text datasets and previous open-vocabulary 3D segmentation datasets, validating its value as a new dataset.

Train dataset	Used ScanNet GT	Mask-Caption Quality # Nouns	Coverage	Entropy	ScanNet20 f-mIoU	ScanNet200 f-mIoU	Head	Com.	Tail
Datasets using only ScanNet as source									
OV3D	✗	2.5K	70.6	72.8	45.6	7.0	18.6	2.1	0.1
RegionPLC	✗	1.4K	77.3	81.0	50.4	8.5	21.1	3.6	0.7
Mosaic3D-SN[2]	✗	9.0K	92.6	**60.7**	65.0	13.0	30.2	6.9	1.4
Datasets using multiple sources									
LEO	✓	2.6K	66.2	-	65.9	14.8	**34.3**	8.3	1.4
SceneVerse	✓	8.8K	60.0	-	67.3	13.6	32.4	7.3	0.8
EmbodiedScan	✓	0.3K	14.0	-	44.8	6.7	16.1	3.6	0.2
MMScan	✓	6.0K	48.0	-	64.1	11.7	26.1	7.9	0.7
Mosaic3D-5.6M	✗	**29.9K**	**93.7**	-	**68.1**	**15.7**	32.9	**10.8**	**2.7**

Table A2. **Dataset comparison.** We analyze mask-caption quality metrics and annotation-free 3D semantic segmentation performance of different training datasets, while keeping the same model architecture (SpUNet-34C), CLIP model (Recap-CLIP), and loss function (Contrastive).

A.2. Data Preprocessing

- **ScanNet** [24] To optimize computational efficiency while maintaining adequate spatial coverage, we process every 20th RGB-D frame from each scene. Prior to processing, we resize all RGB-D frames to 640×480 resolution.
- **ScanNet++** [99] From the official dataset, we utilize the *"DSLR"* image collection. Following repository guidelines, we generate synthetic depth images using the reconstructed mesh and camera parameters. After correcting for distortion in both RGB and depth images and adjusting camera intrinsics, we process every 10th frame through our annotation pipeline. Point clouds are generated via surface sampling on the reconstructed meshes.
- **ARKitScenes** [7] We leverage the *"3D Object Detection (3DOD)"* subset, utilizing its RGB-D frames and reconstructed meshes. We use every 10th frame at low resolution (256×192), and apply surface point sampling on

[2] A subset of Mosaic3D-5.6M using only ScanNet as source dataset.

mesh for point clouds.

- **Matterport3D** [14] We use preprocesed RGB-D frames and point clouds provided by the author of OpenScene [68].
- **Structured3D** [107] We utilize RGB-D frames from both perspective and panoramic camera. We utilize preprocessed point clouds from the *Pointcept* [23] library, which fuses multi-view depth unprojection with voxel downsampling to get point clouds.

A.3. Pipeline Configurations

Our data generation pipeline leverages multiple Visual Foundation Models to automate the data annotation process. Below we detail the configuration of each model in our pipeline.

- **RAM++ [41]**: we utilize the official pretrained checkpoint `ram_plus_swin_large_14m` available at https://huggingface.co/xinyu1205/recognize-anything-plus-model.
- **Grounded-SAM [76]**: We employ the official checkpoint of Grounding-DINO [59] `IDEA-Research/grounding-dino-tiny` accessed through HuggingFace at https://huggingface.co/IDEA-Research/grounding-dino-tiny, together with SAM2 [75] with checkpoint `sam2_hiera_l`, available at https://huggingface.co/facebook/sam2-hiera-large. For the postprocessing, we process the output bounding boxes from Grounding-DINO using a box score threshold of 0.25 and a text score threshold of 0.2. We then apply non-maximum suppression (NMS) with an IoU threshold of 0.5 to remove redundancy. To ensure meaningful region proposals, we filter out excessively large boxes that occupy more than 95% of the image area. These refined bounding boxes are then passed to SAM2 for mask prediction.
- **Osprey [103]**: We utilize the official pretrained `sunshine-lwt/Osprey-Chat-7b` checkpoint, available at https://huggingface.co/sunshine-lwt/Osprey-Chat-7b. The generation parameters are set with a temperature of 1.0, top_p of 1.0, beam search size of 1, and the maximum number of new tokens to 512.

A.4. Additional Pipeline Experiments

We explore two additional data pipeline configurations that use Segment3D [40] masks for segmentation while maintaining Osprey [103] for captioning:

- **Segment3D**: We utilize complete Segment3D masks and obtain captions by aggregating descriptions from multiple projected views of each mask. This approach maintains mask completeness but may result in multiple captions being assigned to a single mask from different viewpoints.

```
System: A chat between a curious human
and an artificial intelligence
assistant.  The assistant gives
helpful, detailed, and polite answers
to the human's questions.
User: <image> This provides an overview
of the picture.  Please give me a
short description of <mask><pos>,
using a short phrase.
```

Table A3. **Osprey region caption prompt**. Osprey [103] utilizes this prompt along with segmentation masks generated by Grounded-SAM to produce descriptive captions for each region.

- **Segment3D - Mosaic**: We use partial Segment3D masks as seen from individual views and generate captions based on these view-specific projections. While masks are partial, each mask-caption pair is aligned since it represents the exact visible region from a specific viewpoint.

The results in Tab. A4 demonstrate that Segment3D - Mosaic outperforms the baseline Segment3D approach, highlighting the importance of precise mask-text pair alignment. However, both Segment3D variants are outperformed by our Mosaic3D pipeline, which suggests that our combination of RAM++ [41], Grounded-SAM [76], and SEEM [110] provides superior segmentation quality.

Pipeline	ScanNet20 [24]		ScanNet200 [78]	
	f-mIoU	f-mAcc	f-mIoU	f-mAcc
Segment3D [40]	50.6	76.6	8.3	19.1
Segment3D [40] - Mosaic	57.3	79.6	10.6	22.8
Mosaic3D	**65.0**	**82.5**	**13.0**	**24.5**

Table A4. **Segment3D data pipeline evaluation results.**

B. OV3D [48] Implementation Details

Since there is no publicly available code and data for OV3D [48], we utilized our re-implemented version of OV3D for data visualization (Fig. 2) and statistics (Fig. 4) in the main manuscript. In this section, we provide detailed explanations of our re-implementation results.

B.1. Caption Generation

OV3D [48] obtains entity-level text descriptions of an image through multi-round conversations with LLaVA-1.5 [57]:

1. In the first round, LLaVA-1.5 is prompted to generate an image caption describing the overall scene.
2. In the second round, LLaVA-1.5 is prompted to extract entity names from the generated image caption.
3. In the final round, LLaVA-1.5 is prompted to generate detailed entity descriptions for each extracted entity name.

During our implementation, we encountered inconsistencies in LLaVA-1.5's response formats. To ensure structured and consistent entity-level text descriptions, we modified the final prompt to request responses in JSON format, as shown in Table A5, while maintaining the original prompts for the first two rounds.

User: Please describe each of the above things that appear in the image with three different nouns or phrases. Format your response as a JSON object with the object names as keys and the list of three nouns or phrases as values. For example: {"entity name A": ["description A1", "description A2", "description A3"], "entity name B": ["description B1", "description B2", "description B3"],..}
Assistant: Here is the dictionary of the concrete objects and background classes in the image:

Table A5. **Modified OV3D entity description prompt.** We modified the original OV3D [48] prompt to request JSON responses for consistent entity descriptions. For brevity, we omit the previous conversation history that is included in the actual prompt.

In addition, our experimental results in Table A7 revealed that LLaVA-1.5's performance in entity name detection was suboptimal, which significantly impacts OV3D's overall effectiveness. To overcome this limitation, we introduce OV3D++, an enhanced version that uses RAM++ [41]'s robust tagging capabilities for entity detection while preserving the original entity description process, as shown in Table A6.

B.2. Training Objectives

We experiment with three different training objectives to reproduce OV3D [48]'s performance:

- `DenseAlign`: The original dense alignment loss proposed in OV3D, which maximizes the similarity between text embeddings and point-wise visual features.
- `Align`: A simplified version of dense alignment that computes similarity between text embeddings and pooled visual features within the mask region.
- `Contrastive`: A contrastive learning objective proposed in RegionPLC [98] that pulls matching text-visual pairs closer while pushing non-matching pairs apart in the embedding space.

For fair comparison, we use SparseUNet34C [21] as the backbone network architecture across all experiments, which is the same architecture used in Mosaic3D, and maintain identical training configurations with the only variations being in the training objectives and data generation pipelines.

User: This is a list of entities, including concrete objects and background classes, in the image: <tag>. Based on your description and the given list of entities, please describe each entity with three different nouns or phrases. Format your response as a JSON object with the object names as keys and the list of three nouns or phrases as values. For example: {"entity name A": ["description A1", "description A2", "description A3"], "entity name B": ["description B1", "description B2", "description B3"],..}
Assistant: Here is the dictionary of the concrete objects and background classes in the image:

Table A6. **OV3D++ entity description prompt with tags.** We use RAM++ [41]'s image tagging output results as the placeholder <tag> to leverage its robust entity detection capabilities. For brevity, we omit the previous conversation history that is included in the actual prompt.

B.3. Results

As shown in Table A7, our direct re-implementation (OV3D-rep) is unable to fully reproduce the performance reported in the original OV3D paper [48]. However, our improved version (OV3D++) with RAM++ [41] tagging achieves better results than the original paper in most metrics when using `Contrastive` loss, except for f-mIoU on ScanNet20 [24]. Notably, `Contrastive` loss consistently outperforms other loss functions across all settings, which motivates our choice to use `Contrastive` loss in Mosaic3D as well. While OV3D++ shows significant improvements over the baseline, it is ultimately surpassed by Mosaic3D, demonstrating the effectiveness of Mosaic3D data engine in generating more fine-grained and comprehensive captions.

C. Experimental Analysis

C.1. Model Scaling

Model capacity. Building on the data scaling analysis, we additionally examine how model scales impact performance. We systematically increase the model sizes of 3D encoders while keeping other components fixed. We vary the size of Sparse ConvUNet by changing the model depth and widths following literature [36], where the smallest model, SPUNet14A, has 11.1M trainable parameters, whereas the largest variants, SPUNet101C, has 256M parameters. For these experiments, we fix the training dataset to include ScanNet, ARKitScenes, and ScanNet++. As shown in Fig. A1, increasing model capacity generally leads to better perfor-

Method	Loss	ScanNet20 [24]		ScanNet200 [78]	
		f-mIoU	f-mAcc	f-mIoU	f-mAcc
OV3D [48]	`DenseAlign`	64.0	76.3	8.7	-
OV3D-rep	`DenseAlign`	34.7	54.9	4.6	8.3
OV3D-rep	`Align`	20.0	34.0	2.4	5.4
OV3D-rep	`Contrastive`	45.6	69.8	6.9	14.3
OV3D++	`DenseAlign`	54.3	71.6	7.0	12.0
OV3D++	`Align`	22.5	37.6	3.1	5.6
OV3D++	`Contrastive`	58.4	76.7	9.2	16.7
Mosaic3D	`Contrastive`	**65.0**	**82.5**	**13.0**	**24.5**

Table A7. **Re-implementation and improvement of OV3D [48].** We present our re-implementation results of OV3D with three different training objectives: `DenseAlign`, `Align`, and `Contrastive`. OV3D-rep denotes our re-implementation, while OV3D++ is our improved version using RAM++ [41] tagging.

mance, with diminishing returns after 100M parameters.

Multi-dataset synergistic learning with PPT [95]. Since our Mosaic3D dataset combines multiple datasets with different capture settings and environments, there potentially exists domain gaps between each subset that could hinder effective joint training. Recent work by Wu *et al*. [95] demonstrates that adapting dataset-specific learnable prompts in normalization layers can reduce negative transfer effects when training on multiple point cloud datasets. Building on this insight, we adopt their Point Prompt Training (PPT) approach to ehance our joint training process. As shown in Fig A1, models using PPT demonstrate better scaling compared to standard joint training, confirming PPT's effectiveness in harmonizing multi-source training on our dataset.

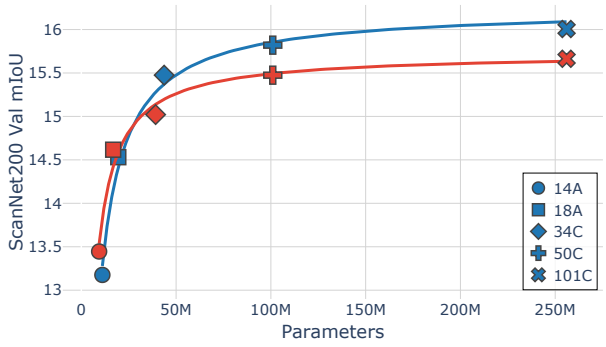


Figure A1. **Model performance scales with model size.** We observe consistent improvements in open-vocabulary semantic segmentation on ScanNet200 [78] as we increase the amount of training data. This shows the value of our large-scale data generation pipeline in improving open-vocabulary 3D scene understanding.

C.2. Impact of Text Encoders

To analyze how different text encoders affect open-vocabulary 3D segmentation performance, we evaluate various CLIP text encoders while keeping the 3D encoder architecture (SPUNet34C) and other components fixed. Table A8 presents the zero-shot performance on ScanNet20 and ScanNet200 benchmarks. We compare standard CLIP text encoders including CLIP/B32, CLIP/B16, and CLIP/L14@336px [71], as well as recently proposed variants like Recap-CLIP [55] and SigLIP [104].

Among all variants, Recap-CLIP achieves the best overall performance with 68.1% f-mIoU on ScanNet20 and 15.7% f-mIoU on ScanNet200. This represents a +0.3% and +0.9% improvement over the base CLIP/B16 model respectively. The superior performance of Recap-CLIP aligns with its enhanced text-image alignment ability demonstrated in 2D vision tasks. Based on these comprehensive experiments, we select Recap-CLIP as our default text encoder for all subsequent experiments. To ensure fair comparisons with previous work, we maintain consistency by using the same text encoder configuration when reproducing baseline results, as shown in Tables 1, 3, and A7. This standardization enables direct performance comparisons and validates the improvements achieved by our proposed approach.

CLIP Model	ScanNet20 [24]		ScanNet200 [78]	
	f-mIoU	f-mAcc	f-mIoU	f-mAcc
CLIP/B16 [71]	67.1	83.8	14.4	27.7
CLIP/B32 [71]	<u>67.8</u>	<u>84.5</u>	14.8	26.5
CLIP/L14@336px [71]	64.2	81.9	14.9	27.7
SigLIP [104]	66.3	**84.6**	<u>15.3</u>	**29.0**
Recap-CLIP [55]	**68.1**	84.4	**15.7**	<u>28.3</u>

Table A8. **Impact of CLIP text encoders on open-vocabulary 3D semantic segmentation.** We train our SPUNet34C architecture on the full Mosaic3D-5.6M dataset (5 subsets) with different CLIP text encoders while keeping other components fixed. Recap-CLIP [55] achieves the best overall performance across both ScanNet20 and ScanNet200 benchmarks, demonstrating the importance of text encoder selection for zero-shot generalization.

C.3. Annotation-free 3D Referring Segmentation

To quantitatively analyze the attention between free-form text queries and point features shown in Fig. 7, we leverage the 3D referring segmentation annotations from ScanRefer [15]. This allows us to evaluate how well our model's attention aligns with human-annotated referring expressions in 3D scenes. Specifically, we evaluate our model's zero-shot performance on the ScanRefer validation set without any fine-tuning on the 3D referring segmentation task. For each referring expression in ScanRefer, we use it as a text query to obtain attention maps between the query and point features. We then threshold the cosine similarity scores to obtain binary segmentation masks, where points with positive similarity scores (greater than 0) are considered as the predicted region. The predicted masks are compared against ground truth annotations using standard IoU metrics.

As shown in Table A9, our method outperforms both OpenScene-3D [68] and RegionPLC [98], demonstrating its superior ability to highlight relevant regions for free-form text queries. These results demonstrate that our model not only excels at semantic segmentation with simple class names but also achieves superior zero-shot performance on more complex free-form referring expressions, quantitatively validating its effectiveness as a general-purpose 3D vision-language foundation model.

Method	OpenScene-3D† [68]	RegionPLC♭ [98]	Mosaic3D
mIoU	3.1	3.7	5.3

Table A9. **Annotation-free 3D referring segmentation on ScanRefer [15].** † and ♭ denote official checkpoints and our reproductions, respectively.

D. Additional Results

D.1. Quantitative Results

In Tab. A10, We conduct a comprehensive evaluation of our model's performance across different category frequencies in ScanNet200. Following standard practice [78], we categorize labels into head, common, and tail groups based on their occurrence frequency in the dataset. As shown in Tab. A10, our approach achieves consistent improvements across all category groups compared to previous methods. Notably, we observe the relative gain is more substantial on common and tail categories as we incorporate more training datasets, highlighting the effectiveness of our multi-dataset training strategy in learning robust features across varying scene distributions.

Method	ScanNet200 val mIoU (%) Head	Common	Tail
OpenScene-3D†	16.4	2.6	0.2
RegionPLC♭	24.2	2.7	0.4
Mosaic3D			
- SN [24]	30.2	6.9	1.4
- SN [24] + AR [7]	32.4	9.3	2.0
- SN [24] + AR [7] + SN2 [99]	33.3	10.0	2.6
- SN [24] + AR [7] + SN2 [99] + M [14]	32.9	10.0	2.5
- SN [24] + AR [7] + SN2 [99] + M [14] + S3D [107]	32.9	10.8	2.7

Table A10. **Category-wise performance analysis on ScanNet200 [78].** † and ♭ denote official checkpoints and our reproductions, respectively.

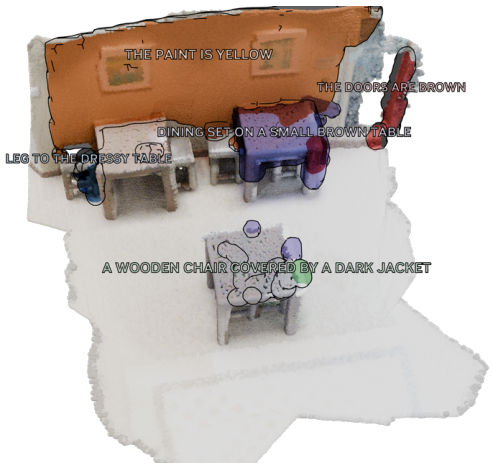
D.2. Qualitative Results

In Fig. A2, we present additional qualitative visualizations of our generated 3D mask-text pair datasets, where we carefully selected mask-text pairs to effectively demonstrate the diversity and quality of our generated data. Furthermore, in Fig. A3, we showcase attention maps for diverse text queries across various scenes, which demonstrates that our model can effectively attend to relevant regions in response to different types of queries, ranging from object-centric descriptions to more abstract concepts like affordances. In Fig. A4, we present qualitative results of annotation-free 3D semantic segmentation on ScanNet200 [78]. Our model shows promising results, particularly in the first scene where it demonstrates an interesting behavior - while the ground truth annotates an integrated chair-desk unit entirely as a chair, Mosaic3D distinctly separates and predicts the desk and chair components. This showcases a potential advantage of our annotation-free approach to training 3D foundation models, where the model can learn more nuanced semantic distinctions that might be overlooked in manual annotations.

(a) scene0055_02 (ScanNet)

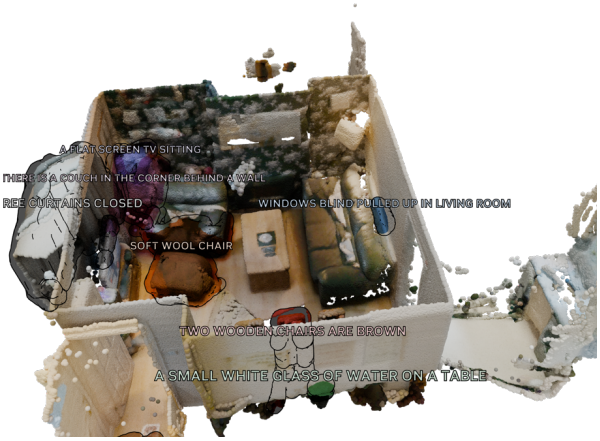
(b) scene0128_00 (ScanNet)

(c) scene0211_01 (ScanNet)



(d) scene0324_00 (ScanNet)

(e) 47333055 (ARKitScenes)



(f) 42898477 (ARKitScenes)

Figure A2. **More visualization of the 3D mask-text pairs in our Mosaic3D-5.6M dataset.** A subset of mask-text pairs has been chosen for better visualization.

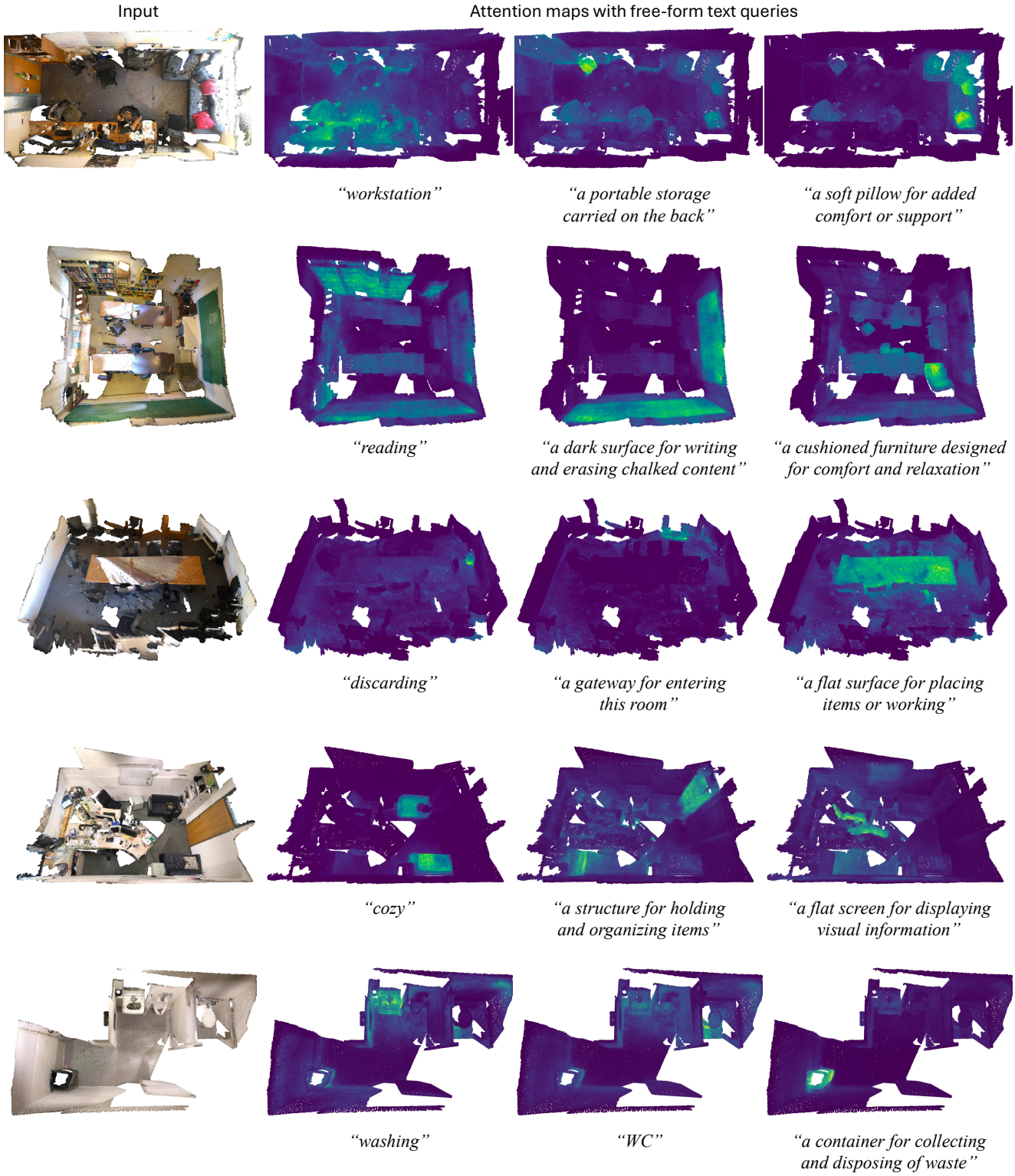


Figure A3. **Attention visualization of Mosaic3D as a 3D foundational model.** We observe that our model can highlight relevant regions even without explicitly mentioning ScanNet [24, 78] class names in queries. The model also effectively attends to regions related to abstract concepts like affordances (*e.g.*, reading, discarding, washing).

Figure A4. **Qualitative results of annotation-free 3D semantic segmentation on ScanNet200 [78].** Despite being trained without ground truth annotations, Mosaic3D shows competitive results on ScanNet200 [78].