GroupContrast: Semantic-aware Self-supervised Representation Learning for 3D Understanding - Supplementary Material

1. Implementation Details

Our implementation is mainly based on Pointcept [2], a codebase focusing on 3D scene understanding and representation learning. The implementation details on pre-training and fine-tuning are listed below.

1.1. Pre-training

Backbone architecture. Following previous self-supervised representation learning approaches [5, 9, 10], we adopt SparseUNet34C [1] as a backbone for ablation studies and result comparisons. The implementation detail of the backbone architecture is the same as in previous approaches.

Pre-training dataset. Following previous work [5, 9, 10], we conduct self-supervised pre-training with GroupContrast on ScanNet v2 [3] point cloud data.

Data augmentation. We follow MSC [9] to set our data augmentation pipeline for all experiments, which include Spatial augmentations, photometric augmentations and sampling augmentations. The data augmentation pipeline is illustrated in Table 1.

Pre-training setting. For ablation studies experiments, the number of default pre-training epochs is 600. For transfer learning results comparison, the number of pre-training epochs is 1200. Please refer to Table 2a for more implementation details at the pre-training stage.

1.2. Fine-tuning

Semantic segmentation. We use a SparseUNet [1] together with a projection layer for semantic segmentation fine-tuning. Experiments are conducted on ScanNet v2 and S3DIS. For ScanNet v2, we fine-tune the model on the training set and report the performance on the validation set. For S3DIS, we report the performance on Area 5 and use other data for fine-tuning. For ScanNet and ScanNet200 semantic segmentation, the model is fine-tuned for 800 epochs with a batch size of 48. For S3DIS semantic segmentation, the model is fine-tuned for 3000 epochs with a batch size of 12. The voxel size is set to 0.02 for ScanNet fine-tuning and 0.05 for S3DIS fine-tuning. Please refer to Table 2b and Table 2c for more details on semantic segmentation fine-tuning. For data-efficient semantic segmentation on ScanNet, we follow the same setting as full dataset fine-tuning, as illustrated in Table 2b.

Augmentation	Value
random rotate	angle=[-1, 1], axis=‘z’, p=1
random rotate	angle=[-1/64, 1/64], axis=‘x’, p=1
random rotate	angle=[-1/64, 1/64], axis=‘y’, p=1
random flip	p=0.5
random coord jitter	sigma=0.005, clip=0.02
random color brightness jitter	ratio=0.4, p=0.8
random color contrast jitter	ratio=0.4, p=0.8
random color saturation jitter	ratio=0.2, p=0.8
random color hue jitter	ratio=0.02, p=0.8
random color gaussian jitter	std=0.05, p=0.95
voxelization	voxel size=0.02
random crop	ratio=0.6

Table 1. **Data augmentation pipeline.**

Instance segmentation. We use SparseUNet [1] as the backbone and PointGroup [6] as the segmentation head for instance segmentation fine-tuning. Experiments are conducted on ScanNet v2 and S3DIS. For ScanNet v2, we fine-tune the model on the training set and report the performance on the validation set. For S3DIS, we report the performance on Area 5 and use other data for fine-tuning. For ScanNet and ScanNet200 instance segmentation, the model is fine-tuned for 800 epochs with a batch size of 48. For S3DIS instance segmentation, the model is fine-tuned for 3000 epochs with a batch size of 12. The voxel size is set to 0.02 for ScanNet fine-tuning and 0.05 for S3DIS fine-tuning. Please refer to Table 2d and Table 2e for more details on instance segmentation fine-tuning.

Object detection. We use SparseUNet [1] as the backbone and VoteNet [8] as the detection head for object detection fine-tuning. Experiments are conducted on ScanNet v2 and SUN-RGBD. We fine-tune the model on the training set and report the performance on the validation set. We report the transfer learning results on ScanNet and SUN-RGBD object detection. We fine-tune the model for 360 epochs with a batch size of 64 for both datasets. The voxel size is set to

Config	Value
optimizer	SGD
scheduler	cosine
learning rate	0.1
weight decay	1e-4
optimizer momentum	0.8
batch size	32
warmup epochs	12
epochs	1200

(a) **Self-supervised pre-training on ScanNet**

Config	Value
optimizer	SGD
scheduler	cosine
learning rate	0.05
weight decay	1e-4
optimizer momentum	0.9
batch size	48
warmup epochs	40
epochs	800

(b) **Semantic Segmentation fine-tuning on ScanNet**

Config	Value
optimizer	SGD
scheduler	cosine
learning rate	0.1
weight decay	1e-4
optimizer momentum	0.9
batch size	12
warmup epochs	0
epochs	3000

(c) **Semantic Segmentation fine-tuning on S3DIS**

Config	Value
optimizer	SGD
scheduler	poly
learning rate	0.1
weight decay	1e-4
optimizer momentum	0.9
batch size	48
warmup epochs	0
epochs	800

(d) **Instance Segmentation fine-tuning on ScanNet**

Config	Value
optimizer	SGD
scheduler	poly
learning rate	0.1
weight decay	1e-4
optimizer momentum	0.9
batch size	12
warmup epochs	0
epochs	3000

(e) **Instance Segmentation fine-tuning on S3DIS**

Config	Value
optimizer	SGD
scheduler	step
learning rate	1e-3
weight decay	0
optimizer momentum	0.9
batch size	64
warmup epochs	0
epochs	180

(f) **Object Detection fine-tuning on ScanNet and SUN-RGBD**

Table 2. **Experiment settings.** We list experiment settings for both upstream pre-training and downstream fine-tuning.

0.02. Please refer to Table 2f for more details on object detection fine-tuning.

2. Collaboration with Foundation Models

We further study the potential of collaborating our work with existing visual foundation models, such as Segment Anything Models (SAM) [7]. Recently, there emerge several works that leverage SAM to predict 3D bounding boxes or segmentation masks on point clouds. These segmentation masks can directly replace the GraphCut [4] results in Segment Grouping. To assess this possibility, we substitute the GraphCut results with the segmentation mask of SAM3D [11] and validate its effectiveness on 3D representation learning. As depicted in Figure 1, Segment Grouping successfully clusters both Graph Cut mask and SAM3D mask into proper regions. The mIoU result for ScanNet-v2 semantic segmentation fine-tuning is $\mathbf{75.9}\%$, which is higher than the result that using Graph Cut ($\mathbf{75.7}\%$). Incorporating existing visual foundation models is a promising way to mitigate data scarcity for 3D visual representation learning. We intend to pursue further in our future research.

3. Prototype Visualization and Analysis

We attempt to visualize the regions assigned to each prototype to analyse whether the randomly initialized prototypes can learn semantic meanings. As illustrated in Figure 2, the model successfully discovers semantic meaningful concepts from unlabeled 3D scenes. These concepts include semantic categories such as floor, table, ceiling and wall, as well as object parts like chair backrests and sofa backrests. The visualization results demonstrate that the prototypes have effectively learned and captured semantic meaning.

Since no supervision signals are provided, the results of segment grouping are bound to orthogonal to the semantic labels sometimes. For example, assign points with the same semantic label to different clusters, or group points with different semantic labels into identical clusters. We believe discovering semantic meaningful subcategories is not harmful at the representation learning stage. It can help the model learn a better representation space and benefit downstream fine-tuning.

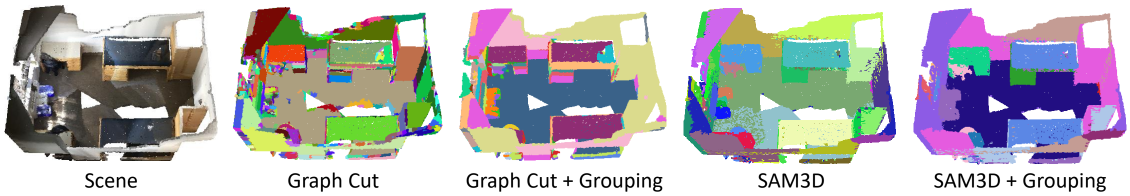

Figure 1. Segment Grouping is capable of aggregating both Graph Cut mask and SAM3D mask into semantic meaningful regions.

Figure 2. **Prototype Visualization.** Each row refers to one prototype, and the group regions are highlighted with a specific color. Our method can discover semantic meaningful concepts from unlabeled 3D scenes.

References

[1] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, 2019. 1

[2] Pointcept Contributors. Pointcept: A codebase for point cloud perception research, 2023. 1

[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

[4] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59:167–181, 2004. 2

[5] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *CVPR*, 2021. 1

[6] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. *CVPR*, 2020. 1

[7] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 2

[8] 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 International Conference on Computer Vision*, 2019. 1

[9] Xiaoyang Wu, Xin Wen, Xihui Liu, and Hengshuang Zhao. Masked scene contrast: A scalable framework for unsupervised 3d representation learning. In *CVPR*, 2023. 1

[10] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *ECCV*, 2020. 1

[11] Yunhan Yang, Xiaoyang Wu, Tong He, Hengshuang Zhao, and Xihui Liu. Sam3d: Segment anything in 3d scenes. *arXiv preprint arXiv:2306.03908*, 2023. 2