Masked Point-Entity Contrast for Open-Vocabulary 3D Scene Understanding

Supplementary Material

A. Implementation Details

We provide implementation and training details of our proposed *Masked Point-Entity Contrast* (MPEC) model.

Model Architecture For open-vocabulary 3D semantic segmentation and zero-shot visual grounding experiments, we ensure a fair comparison with existing methods by employing SparseUNet16 and SparseUNet32 [6] as the 3D encoder and the frozen CLIP [11] as the text encoder. The vision-language adapter consists of a two-layer MLP. For downstream fine-tuning tasks, we train a MinkUNet34C [1] re-implemented with *SpConv* [3] following *Pointcept* [2] using the same point-entity contrastive supervision.

For fine-tuning on low-level perception tasks, we build our experiments using the point cloud perception codebase *Pointcept* [2]. Specifically, for all the semantic segmentation tasks, including closed-set experiments on ScanNet [4], ScanNet200 [12] , we add a single linear layer as the classification head and utilize the cross-entropy loss for supervision. For instance segmentation tasks, we follow previous works [7, 13–15] to adopt PointGroup [9] as the instance segmentation head.

For fine-tuning high-level reasoning tasks, we select PQ3D [18], a state-of-the-art framework for reasoning tasks for indoor scenes as the baseline. Specifically, we replace the voxel encoder of PQ3D with our MinkUNet34C while maintaining other configurations, and fine-tune on the reasoning datasets.

View Generation and Masking Strategy We follow the view generation pipeline and mask strategy in MSC [14]. For a given 3D point cloud, we first create two copies and apply separate random augmentation sequences to each, generating two distinct views of the same scene. The augmentation sequence, detailed in Tab. A.1, consists of three main components: spatial augmentations , photometric augmentations and sampling augmentations .

For the masking strategy, we adopt the approach proposed in MSC [14], setting the grid size to 0.1m to partition the original coordinates into evenly spaced, non-overlapping grids. For each view, 40% of the grids are selected and the features inside are masked and replaced with learnable tokens. Importantly, the selected grids for the two views are mutually exclusive, ensuring no overlap.

Table A.1. **View Generation Pipeline.**

Augmentation	Value
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
grid sample	grid size=0.02
random crop	ratio=0.6
center shift	n/a
color normalize	n/a

Table A.2. **Fine-Tuning Setting on Low-Level Perception Tasks.**

Config	Value
optimizer	SGD
scheduler	cosine decay
weight decay	1e-4
optimizer momentum	0.9
batch size	12
warmup epochs	40
epochs	800

Training For the point-entity contrastive learning, we utilize the AdamW optimizer with a learning rate of 1×10^{-3} for 500 epochs with a cosine warm-up period of 200 steps. During training, we set a batch size of 4 scenes for each GPU and sample 64 text descriptions for each scene. To balance the scale of cross-entropy loss and binary cross-entropy loss in $\mathcal{L}_{\text{e2l}}$, we empirically set α and β to 1.0 and 6.0, respectively. All the contrastive learning experiments are performed on 4 NVIDIA-A100 GPUs with the longest training taking less than 4 days.

For the downstream fine-tuning experiments, we conduct all the low-level perception tasks on *Pointcept* [2] and all the high-level reasoning tasks on PQ3D [18].

The general fine-tuning setting for low-level perception tasks is shown in Tab. A.2. We adjust the learning rate based on the task. Specifically, for full-set semantic and instance segmentation fine-tuning experiments on ScanNet and ScanNet200, the learning rate is set to 0.2.

For training PQ3D on high-level reasoning tasks, we train the model on multiple 3D vision-language tasks including visual grounding, question answering, and dense captioning for 50 epochs. The model architecture uses a hidden dimen-

Table A.3. **Partial Per-Category Performance on ScanNet** [4]. We compare the IoU (%) and accuracy (%) with previous SOTA RegionPLC [16] of each category.

	chair		bookshelf		counter		toilet		sink		shower curtain		curtain	
	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc
RegionPLC	75.4	82.9	72.9	**96.3**	49.0	64.6	64.2	**98.6**	38.1	84.2	43.3	**90.7**	46.5	51.4
MPEC	**83.8**	**85.5**	**80.3**	92.3	**56.4**	**65.2**	**85.6**	98.2	**48.5**	**85.6**	**61.6**	87.1	**66.1**	**73.1**

sion of 768 and 4 query decoder layers. Optimization is performed using the AdamW optimizer with a learning rate of 1×10^{-4}, a batch size of 16, and momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.98$. The loss balancing weights are set to $\lambda_{\text{gen}} = 1$ and $\lambda_{\text{grd}} = 10$.

B. Additional Per-category Performance Analyses on ScanNet

We provide part of the per-category performance on ScanNet in Tab. A.3. As can be seen, though RegionPLC [16] and MPEC achieve similar accuracies on many categories, MPEC continuously outperforms RegionPLC on the IoU metric by a large margin, indicating fewer false positives and better geometric understanding. Superior results on *shower curtain* and *curtain* further highlight MPEC's strong spatial reasoning and semantic understanding ability.

C. Additional Experiment Results for *Zero-shot* 3D Visual Grounding

In Fig. 3 of the main paper, we observe that MPEC still faces challenges when dealing with complicated grounding texts qualitatively. We attribute this phenomenon to the limitations of the fixed CLIP [11] text encoder. This section provides quantitative analysis on *zero-shot* 3D visual grounding experiments on SceneVerse [8] to support this hypothesis.

Experiment Settings Following the SceneVerse-val *zero-shot* setting in [8], we remove MultiScan data during training for fair comparisons. We provide the model ground-truth object proposals and use the pooled feature $\boldsymbol{F}_{\text{VL}}$ for each object to match with the grounding text for predictions. We test the grounding accuracy of different text encoders, *i.e.*, frozen CLIP [11] and trainable BERT [5].

Results & Analyses We present experiments for *zero-shot* 3D visual grounding on SceneVerse-val [8] in Tab. A.4. MPEC with the frozen CLIP text encoder achieves a better overall grounding accuracy of 17% compared with existing available open-vocabulary 3D understanding models like OpenScene [10] and RegionPLC [16]. However, compared with task-specific models for 3D visual grounding, *i.e.*, 3D-VisTA [17] and GPS [8], MPEC with the frozen (❄) CLIP text encoder is considerably lower by more than 35% (17% *vs.* 52+%). After replacing the frozen CLIP text encoder with a trainable (🔥) BERT, the overall accuracy significantly improves from 17% to 42.6%. This underscores the limitation of the frozen LCIP text encoder, which struggles to handle long and detailed descriptions, particularly when grounding specific 3D objects in complex 3D scenes.

Table A.4. ***Zero-Shot* 3D Visual Grounding on SceneVerse-val** [8]. We report accuracy (%) on SceneVerse-val [8] and evaluate models using GT object proposals. 🔥 and ❄ indicates trainable and frozen text encoder, respectively.

Method	Text Encoder	Overall	Easy	Hard
3D-VisTA [17]	BERT🔥	52.9	59.6	35.4
GPS [8]	BERT🔥	**59.2**	**69.4**	**44.0**
OpenScene [10]	CLIP❄	13.3	15.5	10.1
RegionPLC [16]	CLIP❄	10.6	11.8	8.9
MPEC	CLIP❄	17.0	23.8	6.7
MPEC	BERT🔥	42.6	56.2	22.2

D. Additional Experiment Results for Data-efficiency Fine-tuning

In this section, we provide additional fine-tuning experiment results on the ScanNet Data-Efficiency benchmark [7].

Experiment Settings We compare our method with previous methods on ScanNet-LR (Limited Scene Reconstruction) and ScanNet-LA (Limited Annotation) test splits. For ScanNet-LR, we use the $\{1\%, 5\%, 10\%, 20\%\}$ sampled scenes provided in ScanNet-LR and use the annotations within each scene to fine-tune our pre-trained representation $\boldsymbol{F}_P$ for semantic segmentation. Similarly, For ScanNet-LA, we follow [7] and provide $\{20, 50, 100, 200\}$ labeled points per scene for fine-tuning our learned representation. Notably, we train MPEC by removing ScanNet data under this setting and report the mIoU for semantic segmentation on both splits as the evaluation metric.

Results & Analyses As shown in Tab. A.5 and Tab. A.6, our method consistently outperforms previous methods by a large margin, particularly in scenarios with extremely limited

Table A.5. **ScanNet Limited Scene Resconstruction.** We report the mIoU (%) results on ScanNet [4] data efficient semantic segmentation benchmark with limited scene reconstruction setting.

LR	Semantic Segmentation (mIoU)				
Pct.	SC	CSC [7]	MSC [14]	GC [13]	Ours
1%	26.1	28.9	29.2	30.7	**40.8**
5%	47.8	49.8	50.7	52.9	**58.5**
10%	56.7	59.4	61.0	62.0	**64.0**
20%	62.9	64.6	64.9	**66.5**	66.3

Table A.6. **ScanNet Limited Annotation.** We report the mIoU (%) results on ScanNet [4] data efficient semantic segmentation benchmark with limited point annotation setting.

LA	Semantic Segmentation (mIoU)				
Pts.	SC	CSC [7]	MSC [14]	GC [13]	Ours
20	41.9	55.5	61.2	61.2	**62.9**
50	53.9	60.5	66.8	67.3	**69.2**
100	62.2	65.9	69.7	70.3	**72.0**
200	65.5	68.2	70.7	71.8	**73.1**

reconstructions (∼10% improvement for 1% trained scenes). This highlights the ability of MPEC to retain language-aligned 3D feature extraction on unseen scenes and the fast adaptability of the learned representations to downstream tasks under data-scarce scenarios.

E. More Quantitative Results

We provide more qualitative results in Fig. A.1 and Fig. A.2.

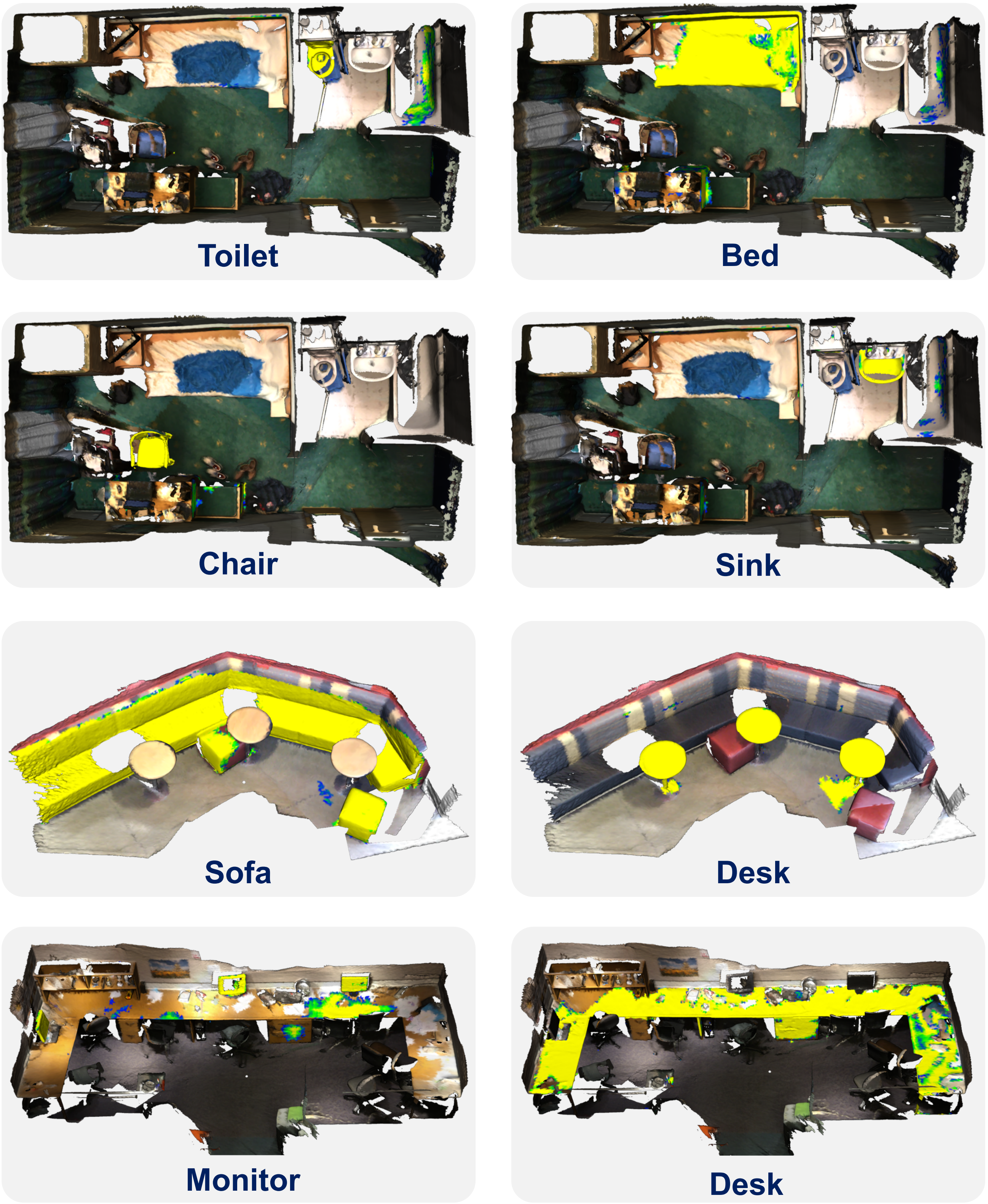

Figure A.1. **More Qualitative Results on ScanNet** [4].

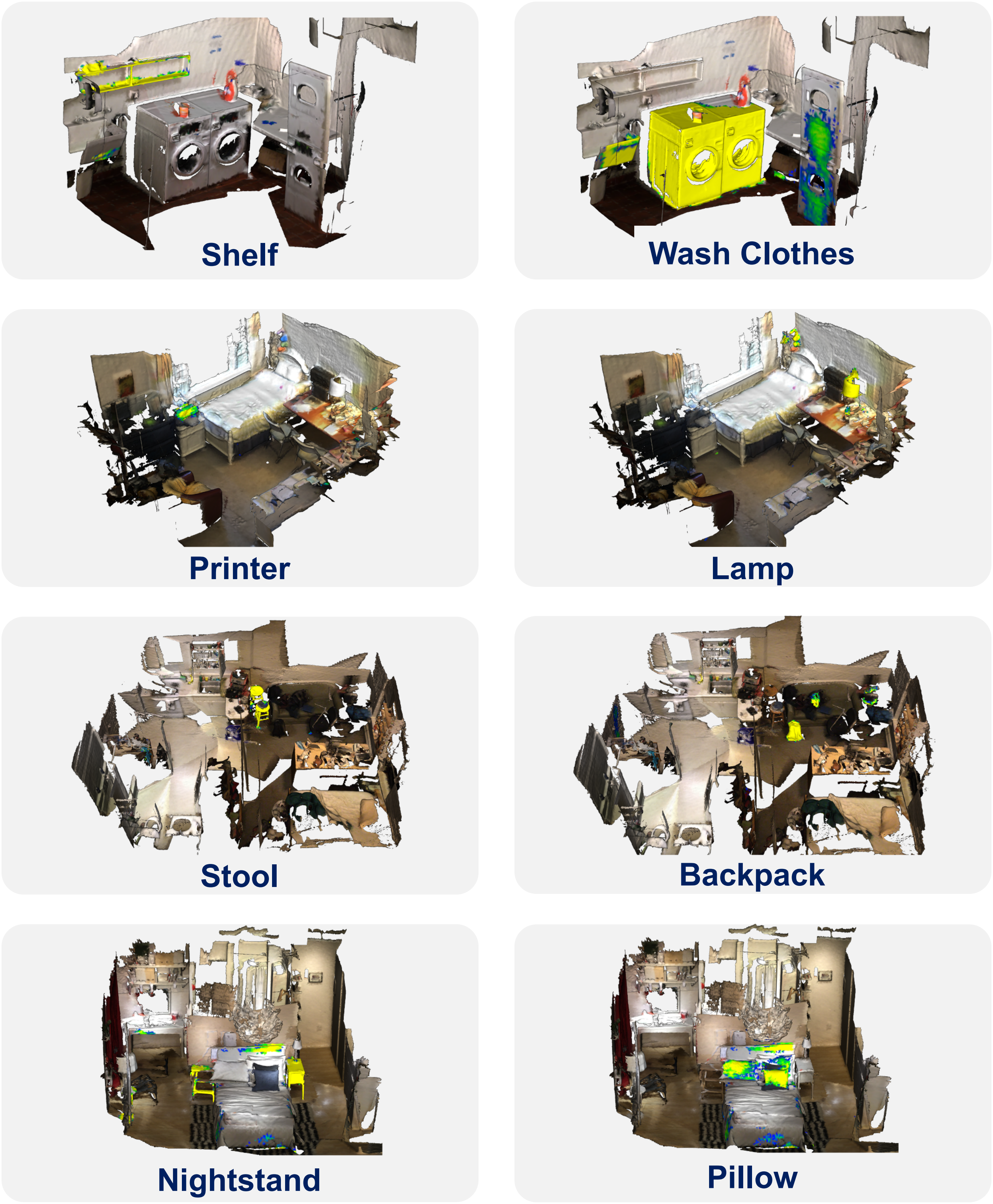

Figure A.2. **More Qualitative Results on ScanNet** [4].

References

[1] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, pages 3075–3084, 2019. 1

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

[3] Spconv Contributors. Spconv: Spatially sparse convolution library. https://github.com/traveller59/spconv, 2022. 1

[4] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017. 1, 2, 3, 4, 5

[5] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2

[6] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, pages 9224–9232, 2018. 1

[7] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *CVPR*, pages 15587–15597, 2021. 1, 2, 3

[8] Baoxiong Jia, Yixin Chen, Huangyue Yu, Yan Wang, Xuesong Niu, Tengyu Liu, Qing Li, and Siyuan Huang. Sceneverse: Scaling 3d vision-language learning for grounded scene understanding. *arXiv preprint arXiv:2401.09340*, 2024. 2

[9] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *CVPR*, pages 4867–4876, 2020. 1

[10] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *CVPR*, pages 815–824, 2023. 2

[11] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 1, 2

[12] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *ECCV*, pages 125–141. Springer, 2022. 1

[13] Chengyao Wang, Li Jiang, Xiaoyang Wu, Zhuotao Tian, Bohao Peng, Hengshuang Zhao, and Jiaya Jia. Groupcontrast: Semantic-aware self-supervised representation learning for 3d understanding. In *CVPR*, pages 4917–4928, 2024. 1, 3

[14] Xiaoyang Wu, Xin Wen, Xihui Liu, and Hengshuang Zhao. Masked scene contrast: A scalable framework for unsupervised 3d representation learning. In *CVPR*, pages 9415–9424, 2023. 1, 3

[15] 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*, pages 574–591. Springer, 2020. 1

[16] Jihan Yang, Runyu Ding, Weipeng Deng, Zhe Wang, and Xiaojuan Qi. Regionplc: Regional point-language contrastive learning for open-world 3d scene understanding. In *CVPR*, 2024. 2

[17] Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *ICCV*, pages 2911–2921, 2023. 2

[18] Ziyu Zhu, Zhuofan Zhang, Xiaojian Ma, Xuesong Niu, Yixin Chen, Baoxiong Jia, Zhidong Deng, Siyuan Huang, and Qing Li. Unifying 3d vision-language understanding via promptable queries. *arXiv preprint arXiv:2405.11442*, 2024. 1