Masked Scene Modeling: Narrowing the Gap Between Supervised and Self-Supervised Learning in 3D Scene Understanding

Supplementary Material

A. Additional Qualitative Results

Fig. 1 presents additional feature visualization of our self-supervised model for different 3D scenes. We follow [15] and use PCA to reduce the point features to three dimensions and visualize them as point colors. Results show that semantically similar objects result in similar features for all scenes.

B. Additional Experiments

B.1. Fine-Tuning

Although not the main focus of this work, we also present results where our self-supervised model is used as a weight initialization step for fine-tuning on the downstream task. In Tbl. 1, we present results on the semantic segmentation task on the three datasets used in our main experiments. Our self-supervised model provides a significant improvement over supervised models trained from scratch and outperforms all existing self-supervised models.

		ScN	ScN200	S3DIS
Fine Tuning	SR-UNet	72.2	25.0	68.2
	+ PC [21]	74.1	26.2	70.3
	+ CSC [8]	73.8	26.4	72.2
	+ MSC [19]	75.3	28.8	–
	+ GC [18]	75.7	30.0	72.0
	HUNet	77.0	35.4	71.3
	+ MSC [19]	78.2	34.9	72.1
	+ Ours	**78.5**	**35.7**	**73.2**

Table 1. **Fine-tuning.** Performance of different pre-trained methods after fine-tuning on the semantic segmentation task.

B.2. Object-Centric Self-Supervised Methods

Another important line of research focuses on self-supervised models pre-trained specifically on object-centric datasets. While these models present strong performance in object-centric tasks, such as shape classification or shape segmentation, those models are not well suited for dense predictions usually required in 3D scene understanding, such as semantic segmentation of large indoor scenes. However, due to the nature of these object-centric models, they are usually also evaluated on the 3D scene understanding task of object detection, where models need to predict the bounding box of objects instead of dense per-point instance segmentation maps. Therefore, we use our self-supervised model as the 3D backbone in an object detection framework to compare our model with such methods.

Dataset. In this experiment, we use the ScanNet dataset [5], and we report mean Average Precission (mAP) with Intersection over Union (IoU) thresholds of 0.5 and 0.25. We use our model as the 3D backbone of the 3DETR [12] object detection framework, and we evaluate our self-supervised model with two different protocols. First, we obtain off-the-shelf features by freezing the 3D backbone while we train the remaining components of the 3DETR [12] framework using our general-purpose features as input. In the second protocol, we also fine-tune all the parameters of the 3D backbone using our self-supervised model as weight initialization.

Baselines. We compare our model to several state-of-the-art self-supervised models pre-trained on object-centric datasets and then fine-tuned on the object detection task. These object-centric models use transformer-based architectures trained with different Masked Image Modelling (MIM) objectives. While Point-Bert [23], Point-MAE [13], and MaskPoint [10] use a non-hierarchical architecture,

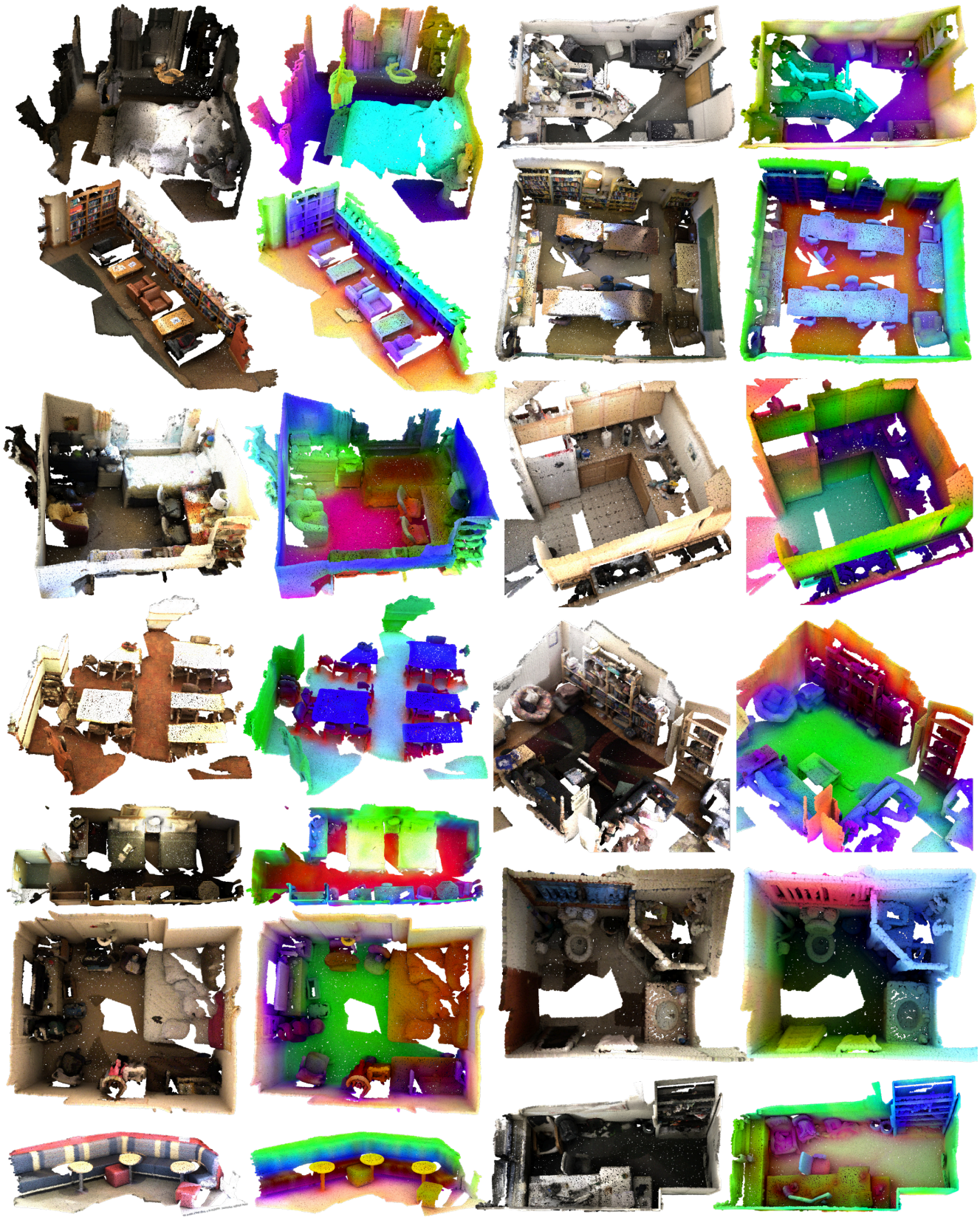

Figure 1. **Qualitative results.** Feature visualization of off-the-shelf features of our method and the baselines. Our learned features align with semantic classes better than existing methods.

	mAP@25	mAP@50
3DETR [12]	62.1	37.9
+ Point-Bert [23]	61.0	38.3
+ Point-MAE [13]	63.4	40.6
+ MaskPoint [10]	63.4	40.6
+ Point-M2AE [25]	66.3	48.3
+ Ours (Lin.)	65.6	40.2
+ Ours (FT)	**71.3**	**52.2**

Table 2. **Object detection.** Comparison of our off-the-shelf features to fine-tuning object-centric self-supervised methods.

	Obj. Det.		Sem. Seg.	
	mAP@25	mAP@50	ScN	S3DIS
Bridge3D [2]	65.3	44.2	73.9	70.2
SAM-MAE [3]	68.2	48.4	75.4	71.8
Ours (Lin.)	65.6	40.2	68.7	59.5
Ours (FT)	**71.3**	**52.2**	**78.5**	**73.2**

Table 3. **2D-3D KD.** Comparison to methods that rely on knowledge distillation from 2D foundation models.

Point-M2AE [25] use a hierarchical model with a bottom-up masking approach. However, all models reconstruct the point coordinates from the last layer in the model.

Results. Tbl. 2 presents the results of our experiments. Our off-the-shelf features, *Lin.* on Tbl. 2, present a competitive performance, outperforming most existing object-centric self-supervised methods. When we further fine-tune our model on the downstream task, *FT* on Tbl. 2, we outperform all models by a large margin. These results are in line with the results presented by Xie *et al*. [21] and highlight the need for scene-centric self-supervised methods.

B.3. 2D-3D Knowledge Distillation Methods

Since general models for 3D scene understanding are not available, recent works have proposed distilling knowledge from 2D foundation models. While Bridge3D [2] combines several 2D foundation models for knowledge distillation into a non-hierarchical 3D transformer architecture, SAM-MAE [3] uses SAM [9] to mask objects in 3D space and a MIM objective to train the same model architecture. We compare our self-supervised model to these models fine-tuned on object detection and semantic segmentation tasks.

Result. Tbl. 3 presents the results of this experiment. While our linear probing setup is not able to achieve the same performance as the baselines, when fine-tuned, our model can outperform them in all experiments.

C. Ablation Studies

In this section, we describe the ablation studies conducted to validate our design choices. For all our experiments, we report linear probing performance on the task of semantic segmentation on ScanNet. Unless otherwise stated, due to the large training times of the self-supervise stage, we perform our ablation studies on a smaller model that takes as input a coarser voxelization of the scene, 4 cm voxels, and we train our models for 800 epochs instead of 1800. For more details of the experimental setup and model used, we refer the reader to Sec. D.

C.1. Masking

In this experiment, we evaluate the importance of our *Masked Scene Modeling* objective. We train a model with our full framework and the same model without our masking strategy. In this version of our framework, the crops given to the student model are not masked, and the full crop is processed by the model. Then, the training objective is the prediction of deep features from the teacher model, which has access to a full view of the scene with different data augmentation. This objective is similar to the self-distillation objective used in MM3D [22]. Tbl. 4 (a) presents the results of this experiment. We can see that the proposed *Masked Scene Modeling* objective is essential for learning semantically relevant features, leading to an improvement of more than **+16** points.

C.2. Hierarchical Supervision

In this experiment, we measure the importance of the hierarchical reconstruction objective. We compare our full framework with a model trained with supervision only on the last layer of the decoder, a common practice in existing self-supervised approaches for 3D scenes [8, 19, 21]. Tbl. 4 (b) shows that supervising only the last layer leads to a gap in performance of more than **+6**. This experiment aligns with the findings of our pilot study and highlights the importance of hierarchical supervision when training hierarchical architectures.

C.3. Masking Strategy

We also compare our bottom-up masking strategy with a traditional top-down approach, similar to the one used in MSC [19]. In this approach, instead of incorporating the masked patches in the decoder, we add them in the encoder with the corresponding learnable token. We can see that in Tbl. 4 (c), even though a top-down approach can lead to relatively good features, our bottom-up approach leads to semantically richer features with more than **+4** points of improvement on the downstream task.

No Mask	50.7
Mask	**66.8**

(a) **Masking.** Patch supervision with vs without masking.

Last	60.5
All	**66.8**

(b) **Supervision.** Layers in the hierarchy used in the loss.

top-down	62.4
bottom-up	**66.8**

(c) **Mask strategy.** Masking hierarchy top-down vs bottom-up.

SparseConv	61.8
MHA	52.3
HUNet	**66.8**

(d) **Model.** Types of model used.

Table 4. **Ablation studies.** Evaluation of the different components of our framework on the task of semantic segmentation on ScanNet.

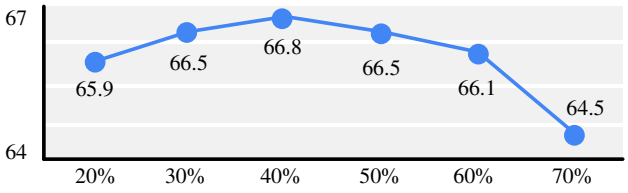

Figure 2. **Masking ratio.** Linear probing performance for different masking ratios.

Layer	Alone	Remove
1	28.3	67.1
2	43.5	67.0
3	54.8	67.0
4	62.8	66.6
5	62.4	64.4
All	68.7	

Table 5. **Layer importance.** Comparison of the performance on the linear probing setup when only one layer is used (*Alone*) or when all layers except one are used (*Remove*).

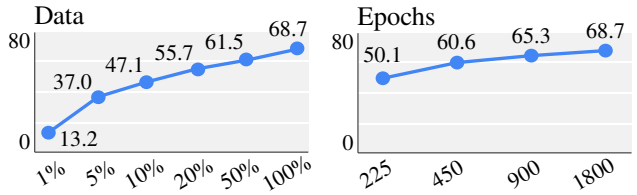

Figure 3. **Scalability experiments.** Evaluate the performance of the model under reduced data used for pre-training and reduced number of epochs.

C.4. Model Architecture

We also evaluate the effect of the model architecture used. We trained two additional models, one only based on Sparse convolutions without Multi-Head Attention (MHA) blocks, and another one with MHA instead of ResNet blocks as in Ptv3 [20]. Tbl. 4 (d) indicates that the model using only sparse convolutions provides lower performance than our hybrid architecture. Moreover, the model with only MHA layers significantly reduces the performance on the downstream task. This is due to the additional constraints of such models, where a lower learning rate is necessary to avoid unstable training. Although we believe that an exhaustive hyperparameter search could lead to an improvement of such models, our hybrid model architecture is robust to higher learning rates and, therefore, easier to train.

C.5. Masking Ratio

Additionally, we measure the influence of the masking ratio on the final performance of the model. We evaluated a range of ratios from 20 % to 70 % with intervals of 10 % and plot the results in Fig. 2. The results show that the framework is relatively robust to the masking ratio used, achieving similar performance for ratios between 30 % and 60 %, with the highest value obtained at 40 %. However, smaller ratios, such as 20 %, or too high, such as 70 %, lead to a significant drop in performance.

C.6. Layer Importance

To expand our pilot study, we further evaluate the importance of the different layers on the performance of our final model. First, we evaluate the linear probing abilities when only one layer is used as input. Then, we evaluate the effect of using all layers except one for the same linear probing setup. Tbl. 5 present the results of this experiment. Results show that, for all layers, using the output of one layer alone (*Alone* in Tbl. 5) leads to a lower performance than using a concatenation of all of them. Moreover, results also show that using all layers except one (*Remove* in Tbl. 5) also leads to a degradation in performance in all cases. This experiment shows the importance of all layers, indicating that each layer provides complementary information.

Additionally, we also evaluate different methods of combining such features. We compare the concatenation of features used in all of our experiments (68.7 mIoU), to a setup where the features are aggregated with a sum operator (68.7 mIoU) and to a setup where the features are aggregated with a learned weighted sum (68.8 mIoU). Our results show that there is no significant difference between these methods.

C.7. Scaling Properties

Moreover, we evaluate the scaling abilities of our framework w.r.t. the data used for pre-training and the number of epochs. For this setup, we use our full model and configuration as in the main experiments in the paper. Fig. 3 presents the results of these experiments. Results show that more data and longer pre-training yield significant improve-

Config	Value
Voxel size	2 cm
Norm layers	RMSNorm [24]
Downsample	Strided SparseConv
Upsample	Transpose SparseConv
Serialization	Z + TZ + H + TH [20]
Block bias	False
Att. drop	0.1
Drop path	0.4
Activation func.	GELU [7]
FF layer	GEGLU [17]
FF ratio	4
Enc. channels	[32, 64, 128, 256, 384]
Enc. ResNet	[2, 2, 2, 2, 2]
Enc. MHA	[0, 0, 0, 2, 2]
Enc. MHA Window	[0, 0, 0, 1024, 1024]
Enc. MHA # Heads	[0, 0, 0, 32, 48]
Dec. channels	[64, 96, 128, 256, 384]
Dec. ResNet	[2, 2, 2, 2, 2]
Dec. MHA	[0, 0, 0, 2, 2]
Dec. MHA Window	[0, 0, 0, 1024, 1024]
Enc. MHA # Heads	[0, 0, 0, 32, 48]

Table 6. **Model configuration.**

Config	Value
Voxel size	4 cm
Norm layers	RMSNorm [24]
Downsample	Strided SparseConv
Upsample	Transpose SparseConv
Serialization	Z + TZ + H + TH [20]
Block bias	False
Att. drop	0.1
Drop path	0.4
Activation func.	GELU [7]
FF layer	GEGLU [17]
FF Ratio	4
Enc. channels	[64, 128, 256, 384]
Enc. ResNet	[2, 2, 2, 2]
Enc. MHA	[0, 0, 2, 2]
Enc. MHA Window	[0, 0, 1024, 1024]
Enc. MHA # Heads	[0, 0, 32, 48]
Dec. channels	[96, 128, 256, 384]
Dec. ResNet	[2, 2, 2, 2]
Dec. MHA	[0, 0, 2, 2]
Dec. MHA Window	[0, 0, 1024, 1024]
Enc. MHA # Heads	[0, 0, 32, 48]

Table 7. **Model configuration for ablation studies.**

ments for linear probing on semantic segmentation. This highlights the importance of additional data and training in self-supervised objectives and paves the road for future improvements of our method.

C.8. NN Robustness

Lastly, we evaluate the robustness of the NN evaluation protocol w.r.t. the distance metric used to compare features. We compare the L2 distance used in all our experiments (65.7 mIoU), to the L1 distance (66.4 mIoU) and to the cosine distance (66.0 mIoU). Although other distance metrics yield slightly better performance, the experiment indicates that the evaluation protocol is robust to the distance metric chosen for evaluation.

D. Detailed experimental setup

D.1. Model architecture

We designed a Hybrid UNet architecture (HUnet) combining standard ResNet blocks [6] with serialization transformer layers as in PTv3 [20]. However, contrary to PTv3 [20], we use sliding-window attention as in LongFormer [1] since this eliminates the need for padding and makes the receptive field adaptive. Moreover, we do not include xCPE [20] in such layers since the ResNet blocks can act as conditional positional encoding. Furthermore, following the design of Stable Diffusion [16], we only included the MHA layers in the lowest resolution levels of the model, making the model faster and more stable to different learning rates. Tbl. 6 presents a detailed description of the different components of our architecture, such as channels per level, number of layers per level, or activation function used. We also provide the configuration of the model used for the ablation studies in Tbl. 7. For these experiments, we used a smaller model with one level less in the encoder and decoder, which takes bigger voxels of 4 cm as input.

D.2. Experiment hyperparameters

Self-supervised training. We build our self-supervised framework on top of the codebase Pointcept [4]. The hyperparameters used for training our self-supervised model are described in Tbl. 8. As data augmentation, we use the default augmentations for indoor semantic segmentation of PTv3 [20]. We only increase the number of points per crop as described in Tbl. 8.

Linear probing - Semantic and Instance segmentation. We use the codebase Pointcept [4] for our linear probing experiments in the downstream tasks of semantic and instance segmentation. The hyperparameters used in these experiments are described in Tbl. 9 and Tbl. 10. For data augmentation, we use the default configuration of PTv3 [20].

Config	Value
Optimizer	AdamW [11]
Betas	(0.9, 0.95)
Weight decay	0.05
Learning rate	0.0015
LR Scheduler	Cosine
Batch size	12
Epochs	1800
Warmup epochs	60
Crop size	240000
Mask Ratio	0.4
Teacher mom.	$0.996 \rightarrow 1.0$

Table 8. **Self-supervised training configuration.**

Config	Value		
	ScanNet	ScanNet200	S3DIS
Optimizer	AdamW [11]		
Betas	(0.9, 0.95)		
Weight decay	0.01		
Learning rate	0.01		
LR Scheduler	Cosine		
Batch size	8		
Epochs	200	200	100
Warmup epochs	2	2	1
Crop size	120000	120000	200000

Table 9. **Linear probing config. for semantic segmentation.**

Config	Value		
	ScanNet	ScanNet200	S3DIS
Optimizer	SGD		
Momentum	0.9		
Weight decay	0.0001		
Learning rate	0.1		
LR Scheduler	PolyR		
Batch size	12	12	8
Epochs	200	200	100
Crop size	–	–	200000

Table 10. **Linear probing config. for instance segmentation.**

Coss-Attention - Visual grounding. Given a 3D point cloud with associated features, 3D ground truth bounding boxes of objects, and a text description, the model is tasked to select the object that matches the text description. We encode the text with the CLIP text encoder [14] and use the attention head of Zhand *et al*. [26] composed of self- and cross-attention layers. The cross-attention layers combine the text CLIP embeddings and object features (obtained from aggregating point features inside object bounding boxes). The output of the model is a probability per object. Then, we train the model using cross-entropy loss, since the task can be formulated as a classification problem where the object matching the text description should have the highest probability. We use the codebase of Multi3DRefer [26] and the hyperparameters used in these experiments are described in Tbl. 11. For data augmentation, we use the default configuration of PTv3 [20] for the task of instance segmentation.

Config	Value
Optimizer	AdamW [11]
Betas	(0.9, 0.95)
Weight decay	0.00001
Learning rate	0.0005
LR Scheduler	Cosine
Batch size	12
Epochs	10
Warmup epochs	1

Table 11. **Visual grounding configuration.**

Config	Value
Optimizer	AdamW [11]
Betas	(0.9, 0.95)
Weight decay	0.1
Learning rate	1e-6
LR Scheduler	Cosine
Batch size	24
Epochs	1080
Warmup epochs	9
Clip gradients	0.1
# queries	256
# points	2048

Table 12. **Object detection configuration.**

Object detection. In these experiments, we use the object detection framework 3DETR [12]. For the linear probing and fine-tuning experiments, we use the same configuration described in Tbl. 12. For data augmentation, we use the default configuration of 3DETR [12].

Fine-tuning - Semantic segmentation. For fine-tuning on the task of semantic segmentation, we use a different configuration than the one used in our linear probing experiments. The hyperparameters of these experiments are described in Tbl. 13.

Config	Value		
	ScanNet	ScanNet200	S3DIS
Optimizer	AdamW [11]		
Betas	(0.9, 0.95)		
Weight decay	0.05		
Learning rate	0.001		
LR Scheduler	Cosine		
Batch size	48	48	32
Epochs	200	200	500
Warmup epochs	2	2	20
Crop size	120000	120000	100000

Table 13. **Fine-tuning config. for semantic segmentation.**

Masked Scene Context. For training our model with the baseline MSC [19], we use different hyperparameters than the ones recommended by the authors. Our model trained with the default parameters leads to subpar performance, obtaining less than 20 mIoU on the task of linear probing for semantic segmentation on ScanNet. Therefore, we modified the number of training epochs to 1800 instead of 600 and the optimizer from SGD to AdamW [11]. These small changes lead to an increase in performance, as reported in the main experiments of this paper.

References

[1] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 5

[2] Zhimin Chen and Bing Li. Bridging the domain gap: Self-supervised 3d scene understanding with foundation models. *Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 3

[3] Zhimin Chen, Liang Yang, Yingwei Li, Longlong Jing, and Bing Li. SAM-guided masked token prediction for 3d scene understanding. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024. 3

[4] Pointcept Contributors. Pointcept: A codebase for point cloud perception research. https://github.com/Pointcept/Pointcept, 2023. 5

[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] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016. 5

[7] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 5

[8] Ji Hou, Benjamin Graham, Matthias Niesner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *IEEE / CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021. 1, 3

[9] 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 (ICCV)*, 2023. 3

[10] Haotian Liu, Mu Cai, and Yong Jae Lee. Masked discrimination for self-supervised learning on point clouds. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 1, 3

[11] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. 6, 7

[12] Ishan Misra, Rohit Girdhar, and Armand Joulin. An End-to-End Transformer Model for 3D Object Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1, 3, 6

[13] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *European conference on computer vision*, pages 604–621. Springer, 2022. 1, 3

[14] 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 *International conference on machine learning (ICML)*, 2021. 6

[15] Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Am-radio: Agglomerative vision foundation model reduce all domains into one. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12490–12500, 2024. 1

[16] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 5

[17] Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020. 5

[18] 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 *IEEE / CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1

[19] Xiaoyang Wu, Xin Wen, Xihui Liu, and Hengshuang Zhao. Masked scene contrast: A scalable framework for unsupervised 3d representation learning. In *IEEE / CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 3, 7

[20] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 4, 5, 6

[21] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 574–591. Springer, 2020. 1, 3

[22] Mingye Xu, Mutian Xu, Tong He, Wanli Ouyang, Yali Wang, Xiaoguang Han, and Yu Qiao. Mm-3dscene: 3d scene understanding by customizing masked modeling with informative-preserved reconstruction and self-distilled consistency. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[23] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19313–19322, 2022. 1, 3

[24] Biao Zhang and Rico Sennrich. Root Mean Square Layer Normalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 5

[25] Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-m2ae: Multi-scale masked autoencoders for hierarchical point cloud pre-training. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3

[26] Yiming Zhang, ZeMing Gong, and Angel X Chang. Multi3drefer: Grounding text description to multiple 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 6