NeRF-MAE: Masked AutoEncoders for Self Supervised 3D Representation Learning for Neural Radiance Fields — Supplementary Material —

Muhammad Zubair Irshad[1,2], Sergey Zakharov[1], Vitor Guizilini[1], Adrien Gaidon[1], Zsolt Kira[2], and Rares Ambrus[1]

[1] Toyota Research Institute {first.last}@tri.global
[2] Georgia Tech {zkira}@gatech.edu

This supplementary material is divided into 5 sections. First, we provide details regarding network architecture and number of parameters in Section A. Second, we provide an overview of implementation details in Section B. This is followed by a discussion on additional experimental analysis in Section C. We provide an overview of pretraining datasets in Section D and discuss compute resources required to pretrain our representation in Section E. Lastly, we discuss limitations and future work in Section F.

A Network Architecture Details

In this section, we provide more details about our architectural design. We use PyTorch [10] for implementation and training our approach. We described in detail the encoder-decoder U-Net style architecture for pretraining in Section 3.4 (in the main paper). We use a constant W, D, H dimension of 160 for all of our datasets, a patch size $p = 4$, and mask out randomly 75% of the patches. We use 4 Swin Transformer blocks and the spatial feature size of the last block is $5 \times 5 \times 5$ with the channel dimension being 768. Consequently, we upsample features with skip connection decoders and use 4 decoders and a final convolution block to restore the original dimension i.e. $B \times 4 \times 160 \times 160 \times 160$. Below we specify the specifications and architectures of our task-specific decoders.

A.1 Task-specific Downstream 3D Heads

Our task-specific downstream heads are attached to the Swin-Transformer encoder. The goal of task-specific heads is to output dedicated meaningful outputs to regress the useful 3D information such as 3D OBBs and semantic labels for each voxel.

3D OBB Prediction: Our 3D OBB prediction head is similar to NeRF-RPN [6] except that for all of our transfer learning experiments and comparisons, we start with warm-started encoder weights via our large-scale self-supervised pretraining, whereas NeRF-RPN weights are started from scratch. Starting from scratch here denotes that the network weights were not pretrained on any other dataset but rather NeRF-RPN initializes the weights of linear layers with truncated normal distribution with a standard deviation of 0.02. We use the author's [6] official implementation for all the comparisons. Specifically, we utilize an anchor-free RPN head from [6] which is based on

Fig. S.1: Data preparation for pretraining: showing multi-view images, trained NeRF and extracted radiance and density grid for Instant-NGP [8] trained NeRFs for Front3D [3], HyperSim [13] and Habitat-Matterport3D [12]. The right column shows multi-view images, rendered RGB/depth images, and extracted radiance and density grid for ScanNet [1] dataset, which is trained using dense depth prior NeRF [14]. Note that we do not pre-train our representation on ScanNet, rather use this dataset for our cross-dataset transfer experiment for the downstream 3D OBB prediction task.

the FCOS detector lifted to the 3D domain via replacing 2D convolutions with their 3D counterparts. We employ a combination of focal loss, IOU loss, and binary cross-entropy loss for downstream transfer learning task similar to [6].

3D Semantic Voxel-Labelling: The task of semantic voxel labeling is described in detail in Section 4.1 (in the main paper). Specifically, we use the same decoders with skip connections i.e. $\mathcal{D}_i$, as described in Section 3.4 (in the main paper) where i ranges from 2 to 4. We replace $\mathcal{D}_1$ by incorporating an additional skip connection with the encoded output from the input grid ($\mathcal{G}$). This is followed by a final convolutional output block to output semantic volume ($\mathcal{G}_S \in \mathbb{R}^{H\times W\times D\times C}$) of spatial dimension W, H, D where $W = H = D = 160$ and $C = 18$ for the Front3D dataset and 20 for the Habitat-Matterport3D dataset. We use a weighted masked cross-entropy loss, balanced by the inverse log propensity of each class, where propensity is defined to be the frequency of each class in the training dataset.

Voxel Super-Resolution: The task of voxel super-resolution is described in detail in Section 4.1 (in the main paper). Specifically, we replace convolution decoders with four 3D convolution layers with instance norm and ReLU activation followed by an upsampling by a factor of 2. This is followed by a final upsampling of 1.6 and 2.4 depending on the output resolution size of 256^3 and 384^3 and a final 3D convolution. Similar to the original reconstruction loss (Section 3.4 (in the main paper)), we employ a masked color reconstruction loss enforced with the ground-truth upsampled grid.

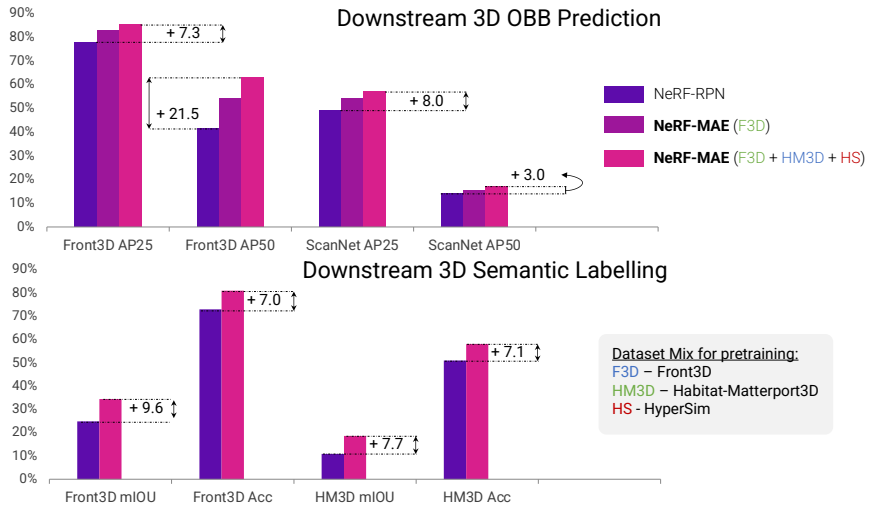

Fig. S.2: Quantitative Comparison: We show further quantitative analysis of results reported in Table 2 and Table 5 in our main paper. Specifically, we highlight improvement numbers for AP25 and AP50 on ScanNet [1] and Front3D [3] and mIOU and Acc for HM3D [12] and Front3D [3] over the state-of-the-art baseline. Furthermore, we also show that increasing unlabelled posed 2D data from different sources improves performance on 3D OBB prediction downstream task.

A.2 Parameters

NeRF-MAE efficiently trains on diverse scenes using a 70 million parameter network, highlighting its capability to handle complex and varied training data. Our Swin-S Transformer encoder architecture as well as the anchor-free RPN head remains the same as NeRF-RPN [6] for a fair comparison on the downstream 3D OBB prediction task. For reconstruction, we employ lightweight decoders (Section 3.4 in the main paper) since for our transfer learning experiments, we eventually discard the decoders and utilize the feature pyramids for downstream 3D tasks as described above.

B Implementation Details

To pre-train NeRF-MAE, we utilize the Swin-S backbone of the SwinTransformer. As detailed in Section 3.4 (in the main paper), we utilize four swin blocks of depth 2, 2, 18, and 2 with the number of heads in each block equalling 3, 6, 12, and 24. We pre-train both encoder and decoder networks for 1200 epochs with the Adam optimizer and a maximum learning rate of $3e^{-4}$, a weight decay of $1e^{-3}$ with a one-cycle learning rate scheduling. We employ a batch size of 32 for our pretraining. We additionally utilize online data augmentation during both pretraining and transfer learning stages. We employ random flip, rotation, and scaling augmentation with a probability of 50%. Data augmentation is shown to improve our pretraining, as described in Section 4. For transfer learning, we train both our network and the baseline NeRF-RPN [6] for 1000 epochs for 3D OBB prediction and 500 epochs for voxel super-resolution with the same learning

	256^3		384^3	
Method	**PSNR↑**	**MSE ↓**	**PSNR↑**	**MSE ↓**
NeRF-RPN* [6]	14.70	0.034	14.65	0.035
NeRF-MAE(Ours)	**15.01**	**0.032**	**15.20**	**0.030**

3D OBB on HM3D [12]		
Method	**AP@25↑**	**Recall@50↑**
NeRF-RPN [6]	0.30	0.31
NeRF-MAE (Ours)	**0.37**	**0.40**

Table S.1: Voxel super-resolution and 3D OBB prediction quantitative results for HM3D [12] test-set showing superior performance of our method vs NeRF-RPN [6]. These results further confirm the findings of Tables 2 and 5 in our main paper that our pretraining helps improve downstream tasks on various datasets.

Pretraining scenes on downstream performance					
Pretraining Scenes		Metrics			
% Scenes	# Scenes	**Recall@25↑**	**Recall@50↑**	**AP@25↑**	**AP@50↑**
10%	151	0.94	0.61	0.74	0.45
25%	378	0.95	0.62	0.76	0.46
50%	757	0.95	0.66	0.77	0.51
100%	1515	**0.96**	**0.67**	**0.79**	**0.54**

Pretraining scenes on reconstruction quality				
Pretraining Scenes			Metrics	
% Scenes	**# Scenes**	**Aug.**	**3D PSNR ↑**	**MSE ↓**
10%	151	✗	14.20	0.038
25%	378	✗	14.98	0.032
50%	757	✗	15.42	0.028
100%	1515	✗	16.00	0.025
100%	1515	✓	**19.09**	**0.012**

Table S.2: Effect of percent of pretraining scenes on downstream 3D object detection (left) and voxel-grid reconstruction (right) showing more unlabeled data helps learn better representation for downstream 3D tasks as well as better reconstruction quality of unmasked patches in the input radiance and density grid

rate and strategy as our pretraining stage since we empirically determined that changing the learning rate and strategy results in decreased performance for both our method and the baseline. We additionally utilize a maximum normal gradient clip value of 0.1 and fine-tune the network until convergence. We determine transfer learning convergence on the hold-out validation set and employ early stopping based on AP50 metrics for 3D OBB prediction, accuracy metric for 3D semantic voxel labeling, and 3D PSNR metric for voxel super-resolution experiments. For transfer learning experiments, we use a batch size of 8.

Pretraining and Fine-tuning Weights: In the spirit of the original 2D MAE work [4], the pretraining weights of the model are general-purpose, containing the scene understanding knowledge such as semantic and spatial structure. Finetuning allows the model weights to be task-specific and aligns the model with the task using lightweight decoders. We will further include a detailed discussion on this topic in our paper.

	Front3D [6]		ScanNet [1]	
NeRF-MAE	**Recall@50↑**	**AP@25↑**	**Recall@50↑**	**AP@25↑**
w/o Skip	0.73	0.83	0.39	0.54
w/ Skip	**0.74**	**0.85**	**0.40**	**0.57**

Masking Ratio Ablation		
Method	**AP@25↑**	**AP@50↑**
65%	0.805	0.519
75%	**0.839**	**0.546**
85%	0.814	0.512

Table S.3: Effect of skip connections in the decoder (Left) and masking ratio ablation (Right): shows NeRF-MAE with skip connection helps learn better representations for downstream tasks as well as 75% masking ratio achieves the best downstream performance.

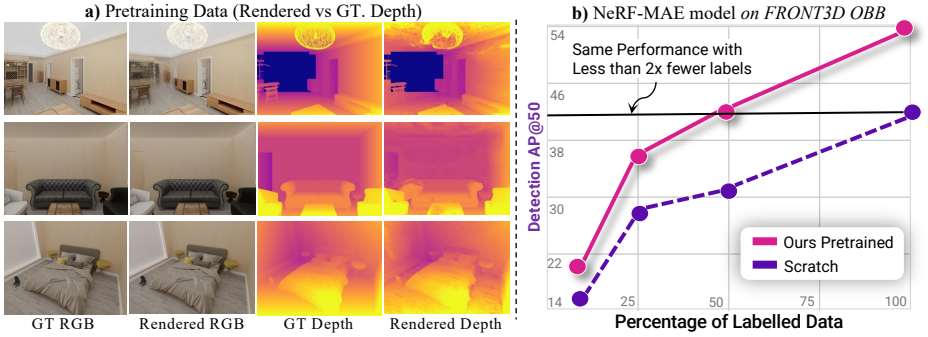

Fig. S.3: Qualitative Rendered Depth (a) and Quantitative Detection Performance with a varying number of labeled scenes (b): We show the qualitative comparison of ground truth vs rendered depth map from NeRFs used for our pretraining scenes showing faithful reconstruction of geometry for our input data. We further show in the chart on the right (b), that our pretraining requires much less data to achieve the same performance as the strong baseline.

C Experiment Analysis

In this section, we present additional experimental analysis. First, we present in detail an additional discussion on the downstream 3D object detection, semantic voxel-labeling, and voxel super-resolution tasks in Section C.1. We present the additional analysis of scaling laws and ablation studies in Section C.2 specifically highlighting the architectural ablation of adding skip connections to the decoder and discussing masking ratio ablation. We present additional insights with comparison to 3D pretraining baselines in Section C.3 and details about per-scene NeRF optimization to prepare our pretraining dataset in Section C.4

C.1 Comparison on Downstream 3D Tasks

We analyze all of the improvement numbers mentioned in Table 2 and Table 5 by highlighting them in Figure S.2. One can clearly see our pretraining consistently improves all metrics across different datasets, especially on ScanNet which is omitted from our pretraining and used as a hold-out dataset for our cross-dataset transfer experiments. The results also clearly show that downstream 3D task numbers for 3D OBB prediction improve with adding unlabelled data from diverse sources, in this case, adding unlabelled HM3D scenes to our pretraining improves ScanNet numbers showing the efficacy of our pretraining strategy. Note that we only utilized posed 2D data for pertaining, and no 3D information was used in our pretraining pipeline.

Next, we present results for the task of voxel grid super-resolution on HM3D dataset hold-out scenes never seen by the model during pretraining or transfer learning. We summarize these results in Table S.1.

One can clearly observe that our technique achieves a higher Recall@50 and AP@50 compared to strong baseline, specifically showing improvement of 9% in Recall@50 and 7% in AP@50, further confirming the efficacy of our pretraining strategy on a challenging dataset and 3D task. The results in Table 1 (Left) further confirm the finding

of Table 5 (Left), achieving consistently higher 3D PSNR and lower MSE metrics for hold-out test scenes on HM3D for the challenging voxel grid super-resolution task.

We highlight additional qualitative and qualitative comparison results in Figure S.3. This figure shows the rendered depth maps against ground truth depth which highlights the input used to pretrain our representation provides good geometric cues to the network. We further show the impact of our pretraining on the amount of labeled data required during finetuning. One can clearly see from Figure S.3 (Right) that our pretraining requires twice as fewer scenes for finetuning compared to the state-of-the-art baseline that starts from scratch to achieve the same performance, especially achieving an AP50 of 0.42 with only 50% of the labeled data for finetuning (i.e. 61 scenes) whereas the strong baseline uses 100% of the labeled data (i.e. 122 scenes) to achieve a similar performance with 2x as much data as required by our pipeline.

C.2 Additional Ablation Analysis

We report complete metrics for our scaling laws experiments shown in Figure 2 of the main paper. We report these metrics in Table S.2. Both results further confirm our findings in the paper that our results consistently improve on reconstruction task as well as during transfer learning by utilizing more unlabelled data during pretraining. We additionally include ablations to our NeRF-MAE pretraining architecture in Table S.3. We show the effect of adding skip connections to the lightweight decoders in Table S.3 (Left). One can clearly see that adding skip connections during reconstruction increases the accuracy of downstream tasks, specifically increasing the AP25 by 2% and 3% on FRONT3D and ScanNet datasets respectively. We also analyze the effect of masking ratio on downstream task performance in Table S.3 (Right). This table clearly shows our design choice of using 75% as the masking ratio for input radiance and density grid. This significantly increases the accuracy of 3D object detection downstream task.

C.3 Comparison to 3D pretraining baselines

We present a detailed comparison to 3D pretraining baselines [5, 9, 20, 21] in Table 1 in our main paper. Here, we share more insights on these results. Despite given NeRF rendered depth maps (from the same NeRF model, NeRF-MAE uses to sample its radiance and density grid from) and ground truth poses, the baselines [5, 21] struggle to improve performance in an extreme data-scarcity setting (i.e. on cross-domain generalization where 90 scenes are available for finetuning), whereas, our approach, NeRF-MAE excels in these challenging scenarios. We conclude that this is due to the local training structure of DepthContrast and other MAE approaches, whereas NeRF-MAE pretraining captures both the local and global structure of the full-scenes with its Transformer-inspired architecture. Despite given NeRF's radiance and density grid as an input to point-based pretraining baselines [9, 20], we see degraded performance. We conclude that these approaches use specialized modules like KNN used in Point-MAE [9], DGCNN [16] used in Point-BERT [19] which are specialized for modeling surface level information and require extra parameter tunings such as group size and

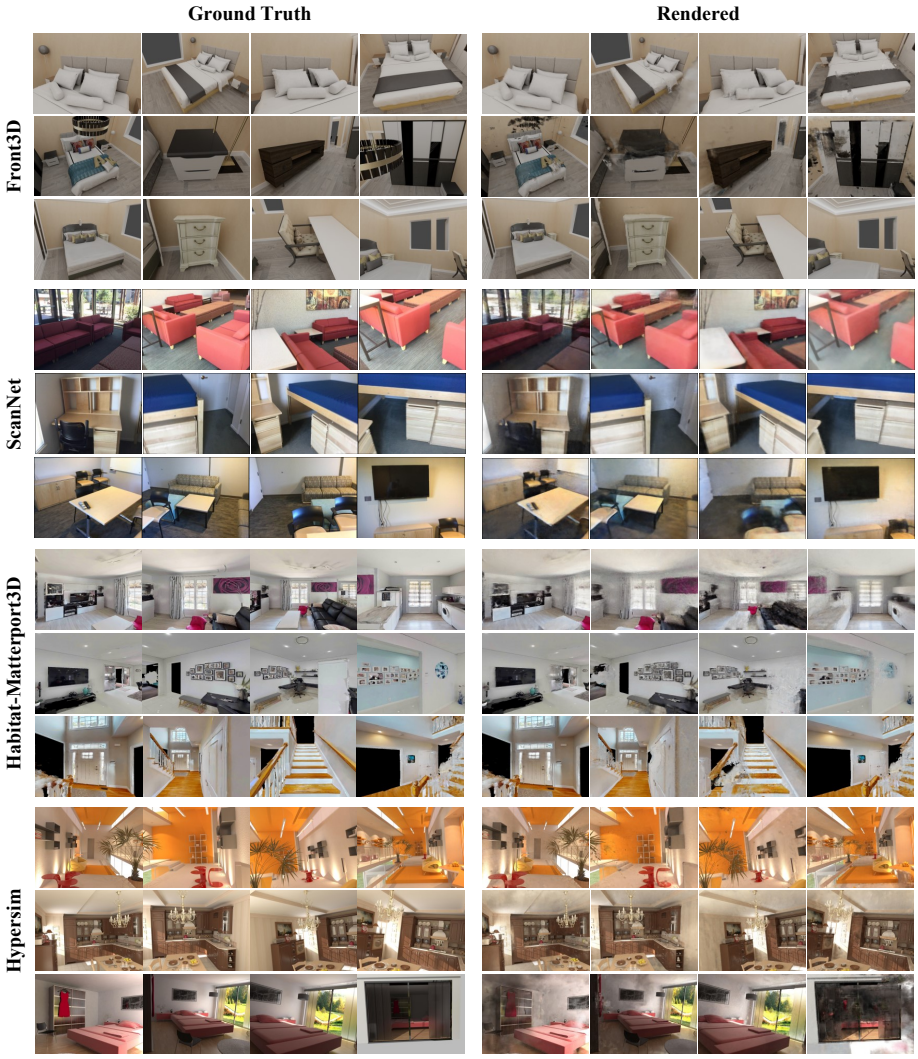

Fig. S.4: Multi-view Rendered vs Ground-truth posed RGB images from four distinct datasets used in this paper i.e. Front3D [3], ScanNet [1], Habitat-Matterport3D [12] and Hypersim [13]. As mentioned in the main paper, we use Instant-NGP to train neural radiance fields for Front3D, Hypersim and HM3D, whereas Scannet scenes are trained using dense depth prior nerf [14], following [6]

number of centers. Even though PointM2AE [20] improved performance over Point-MAE [9] using a hierarchical training, we show that they still struggle for this challenging downstream task where only a few labeled scenes are available for finetuning. On the contrary, our approach utilized standard Transformer modules and shows very effective downstream performance with an opacity and radiance-aware masked reconstruction objective.

	HM3D [12] (NeRF-MAE)		ScanNet [1] test (NeRF-MAE)	
Method	**Memory**↓	**Memory(Rel)** ↓	**Memory**↓	**Memory(Rel)** ↓
Original (RGB)	256GB	± 0.00%	30GB	± 0.00%
Compressed (Weights)	**46GB**	**− 82.03%**	**1.8GB**	**− 94.00%**

Table S.4: Relative Pretraining Dataset Comparison: showing significant memory saving during data preprocessing for NeRF-MAE.

C.4 Per-scene NeRF optimization Analysis

Our per-scene NeRF optimization takes 8 mins on 1 GPU for a total of just over 2 days on 8 GPUs for the entire pre-training dataset. A by-product of this process that we would like to highlight is the compression of the original data (see Table S.4). Our results indicate significant memory savings, and the resulting compressed representation still conveys all the necessary information for downstream tasks.

D Pretraining Datasets

We show our pretraining datasets in detail in Figure S.1 and Figure S.4. First, we show the data processing flow for each of the Front3D [3], Scannet [1], HM3D [12] and Hypersim [13] dataset in Figure S.1 starting with multi-view images to get a NeRF trained for each scene and finally getting an explicit radiance and density grid for all scenes. We train each scene for 100,000 steps to ensure high-quality NeRF are obtained as input to our pretraining stage as further confirmed by our experiments in Figure 2 in the main paper, which quantitatively shows that better NeRF quality results in better pretraining and leads to improved downstream 3D task performance. We further show rendered RGB vs ground-truth RGB for selected scenes from all four datasets in Figure S.4. The figure confirms that our NeRF quality for validation images is faithful to the ground-truth images. Secondly, it shows that our pretraining dataset for training good 3D representations from posed 2D data is very diverse in realism, lightning, shadows, number of objects in the scene as well as the setting of each scene i.e. ranging from bedroom scenes to office and living room scenes. This makes our 3D representation learning more robust to the varying input data and helps it improve performance for data outside the training data-distribution, as further confirmed in Section 4.

E Compute Resources

Our model was trained on 7 Nvidia A100 GPUs for 1200 epochs for 2 days and 17 hours. The cost for a pretraining run is around 19 GPU days with a GPU capacity of 82GB and an average utilization of 60%. To pretrain our representation, each NeRF was trained for 8 minutes on a single GPU for 100,000 steps. The total training time for training 3500 NeRF was 58 hours and finished in just over 2 days on 8 Nvidia A100 GPUs. Note that our pretraining strategy is agnostic to the NeRF technique used and can be replaced with any of the faster NeRF techniques developed. We show that our pretraining strategy equally applies to both instant-NGP [8] trained NeRF and dense depth prior NeRF [14] and doesn't require any modification to our network architecture or training strategy.

F Limitations and Future Work

NeRF-MAE enables large-scale self-supervised pertaining of 3D representations using NeRF's unique radiance and density grid as an input hence employing standard 3D Transformer blocks to significantly improve various downstream 3D tasks. Although powerful, future work is still required to improve the efficiency of training this representation to allow for seamless and efficient input of a large variety of multi-view data from diverse data sources. This could be achieved using fast linear attention blocks [15] or employing a 3D version of state space models [2]. Another avenue for future work is communication between neural rendering and masking. Concurrent works have combined neural rendering with 3D representation learning [7, 22] albeit in a non-masking manner. Other works have used correspondences to mine multi-view image pairs to train MAE [4] and designed the task of cross-view completion [17] for unsupervised 3D representation learning from 2D images. Our work NeRF-MAE bridges the 2D and 3D domains by using Neural Radiance Fields and future work could look into back-propagating the network weights through both our reconstructed grid as well as neural rendered images to further improve performance. Another promising avenue for future work is to explore various downstream 3D tasks that could benefit from dense NeRF-MAE pertaining such as 3D scene reconstruction [11, 18].

References

1. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5828–5839 (2017)
2. Fu, D.Y., Dao, T., Saab, K.K., Thomas, A.W., Rudra, A., Ré, C.: Hungry hungry hippos: Towards language modeling with state space models (2023)
3. Fu, H., Cai, B., Gao, L., Zhang, L.X., Wang, J., Li, C., Zeng, Q., Sun, C., Jia, R., Zhao, B., et al.: 3d-front: 3d furnished rooms with layouts and semantics. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10933–10942 (2021)
4. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. arxiv preprint arxiv: 211106377 (2021)
5. Hou, J., Graham, B., Nießner, M., Xie, S.: Exploring data-efficient 3d scene understanding with contrastive scene contexts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15587–15597 (2021)
6. Hu, B., Huang, J., Liu, Y., Tai, Y.W., Tang, C.K.: Nerf-rpn: A general framework for object detection in nerfs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23528–23538 (2023)
7. Huang, D., Peng, S., He, T., Yang, H., Zhou, X., Ouyang, W.: Ponder: Point cloud pre-training via neural rendering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16089–16098 (2023)
8. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (ToG) **41**(4), 1–15 (2022)
9. Pang, Y., Wang, W., Tay, F.E., Liu, W., Tian, Y., Yuan, L.: Masked autoencoders for point cloud self-supervised learning. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II. pp. 604–621. Springer (2022)

10. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library (2019)
11. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional occupancy networks. In: European Conference on Computer Vision (ECCV) (2020)
12. Ramakrishnan, S.K., Gokaslan, A., Wijmans, E., Maksymets, O., Clegg, A., Turner, J.M., Undersander, E., Galuba, W., Westbury, A., Chang, A.X., Savva, M., Zhao, Y., Batra, D.: Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2021)
13. Roberts, M., Ramapuram, J., Ranjan, A., Kumar, A., Bautista, M.A., Paczan, N., Webb, R., Susskind, J.M.: Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10912–10922 (2021)
14. Roessle, B., Barron, J.T., Mildenhall, B., Srinivasan, P.P., Nießner, M.: Dense depth priors for neural radiance fields from sparse input views. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12892–12901 (2022)
15. Wang, S., Li, B.Z., Khabsa, M., Fang, H., Ma, H.: Linformer: Self-attention with linear complexity (2020)
16. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics (TOG) (2019)
17. Weinzaepfel, Philippe and Leroy, Vincent and Lucas, Thomas and Brégier, Romain and Cabon, Yohann and Arora, Vaibhav and Antsfeld, Leonid and Chidlovskii, Boris and Csurka, Gabriela and Revaud Jérôme: CroCo: Self-Supervised Pre-training for 3D Vision Tasks by Cross-View Completion. In: NeurIPS (2022)
18. Yan, S., Yang, Z., Li, H., Guan, L., Kang, H., Hua, G., Huang, Q.: Implicit autoencoder for point cloud self-supervised representation learning. arXiv preprint arXiv:2201.00785 (2022)
19. Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J., Lu, J.: 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. pp. 19313–19322 (2022)
20. Zhang, R., Guo, Z., Gao, P., Fang, R., Zhao, B., Wang, D., Qiao, Y., Li, H.: Point-m2ae: Multi-scale masked autoencoders for hierarchical point cloud pre-training. arXiv preprint arXiv:2205.14401 (2022)
21. Zhang, Z., Girdhar, R., Joulin, A., Misra, I.: Self-supervised pretraining of 3d features on any point-cloud (2021)
22. Zhu, H., Yang, H., Wu, X., Huang, D., Zhang, S., He, X., He, T., Zhao, H., Shen, C., Qiao, Y., Ouyang, W.: Ponderv2: Pave the way for 3d foundation model with a universal pre-training paradigm. arXiv preprint arXiv:2310.08586 (2023)