SeRP: Self-Supervised Representation Learning Using Perturbed Point Clouds

Siddhant Garg
University of Massachusetts Amherst
siddhantgarg@umass.edu

Mudit Chaudhary
University of Massachusetts Amherst
mchaudhary@umass.edu

Abstract

*We present SeRP, a framework for Self-Supervised Learning of 3D point clouds. SeRP consists of encoder-decoder architecture that takes perturbed or corrupted point clouds as its inputs and aims to reconstruct the original point cloud without corruption. The encoder learns the high-level latent representations of the points clouds in a low-dimensional subspace and recover the original structure. In this work, we have used Transformers and PointNet based Autoencoders. The proposed framework also addresses some of the limitations of Transformers based Masked Autoencoders [25] which are prone to leakage of location information and uneven information density [14]. We pretrained our models on the complete ShapeNet dataset [2] and evaluated them on ModelNet40 [22] as a downstream classification task. We have shown that the pretrained models achieved $0.5 - 1\%$ higher classification accuracies than the networks trained from scratch. Furthermore, we also proposed VASP: **V**ector-Quantized **A**utoencoder for **S**elf-supervised Representation Learning for **P**oint Clouds that employs Vector-Quantization [19] for discrete representation learning for Transformer based autoencoders.*

1. Introduction

Self-Supervised learning methods have come a long way today. They have been extremely successful in the domains of Natural Language Processing [5, 4] and 2D Computer Vision [9, 8] where large unlabeled datasets are leveraged to introduce necessary inductive biases in the models that can be transferred on the downstream tasks with limited annotated datasets. Recently, many self-supervised learning methods are also being proposed for 3D deep learning [23, 10, 3, 12], specifically for representation learning of the raw point clouds data. The annotation of point cloud is a difficult process and there are lesser annotated datasets in 3D deep learning relative to NLP and 2D computer vision. With the proliferation of modern 3D scanners, that can generate large unlabeled point clouds, 3D self-supervised learning methods is a promising direction to improve results on small annotated 3D datasets.

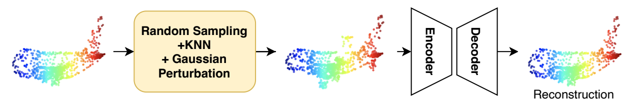

Figure 1: Overview of SeRP. First, an input point cloud is perturbed or corrupted and then it is processed by an autoencoder to learn latent representations using reconstruction process.

In this work we are proposing Auto-Encoders for learning high level latent representations of points clouds. As shown in Figure 1, the encoder takes a perturbed or noisy point cloud and the decoder aims to reconstruct the original point cloud from the latent representation. We used PointNet [16] and Point Cloud Transformers to design our Auto-Encoder framework. Our approach also overcomes the issue of information leakage associated with masked auto-encoding [14, 25], where the position embeddings of the masks are also provided to the model [14, 25].

Our contributions can be summarized as follows:

- Proposed a perturbation technique to create noisy point clouds by selecting small patches using random samplint and adding a Gaussian noise to the patch of point clouds.
- Proposed PointNet and Point Cloud Transformer based Auto-Encoders to learn latent representations of the point clouds.
- Evaluated the pretrained encoder representations on downsteam classification tasks where we showed improvement of $0.5 - 1\%$ on ModelNet40 dataset and ShapeNet55 classification datasets.
- Also proposed Vector-Quantization for Transformer based auto-encoding.

2. Related Works

Self-supervised learning methods leverage the unlabeled data to learn robust representations. This can be done by

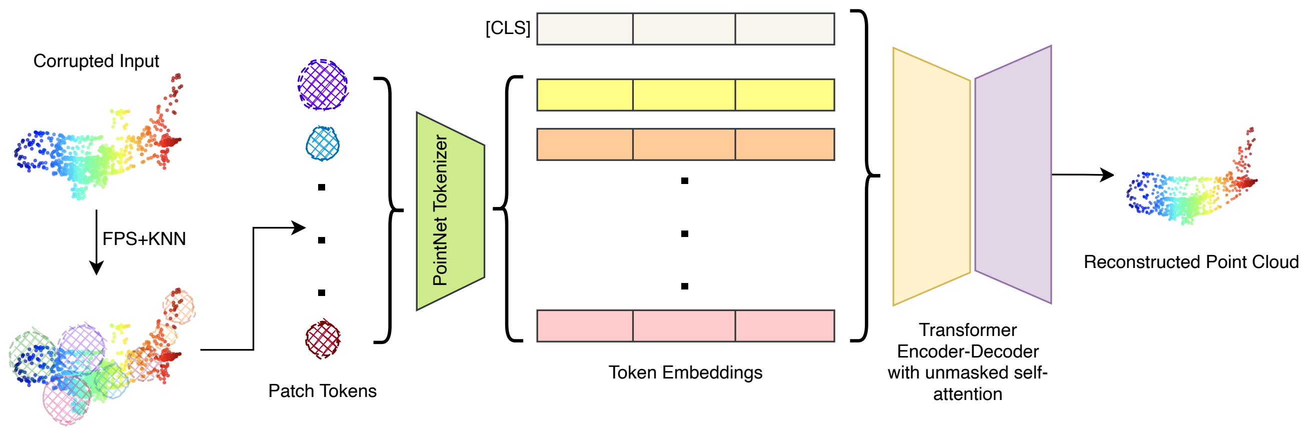

Figure 2: Overview of SeRP-Transformer model. The corrupted point cloud is divided into patches using FPS and KNN to create patch tokens. The token embeddings are learned using a lightweight PointNet. The embeddings are passed through a Transformer Encoder-Decoder with unmasked self-attention. The trained encoder is then used for downstream tasks.

developing pretext tasks like learning reconstruction spaces [18], orientation estimation [15], and occlusion completion [21] for point clouds. Some other techniques fall under the umbrella of contrastive [23, 26] and non-contrastive [3, 10] learning methods that learn by making the representations of augmented views of same object, closer and augmented views of different objects farther from each other.

Recently, a lot of work has been done towards 3D self-supervised representation learning from the raw point cloud data. Point Constrast [23] is a contrastive learning method that uses registration objective as a pretext task, DGCLR [7] used knowledge distillation with constrastive learning, and recently, Point-BERT [25], which is transformer [5] based network used BERT-style pre-training by masking patches of point clouds and predict the masked parts with dVAE [17]. Many auto-encoding methods like, Point-MAE [14], and Implicit Autoencoders [24] are also proposed that aim to reconstruct point clouds and surfaces respectively. Some of the recent works also try to leverage the spatio-temporal dimension of the input point clouds [10, 3, 13]. There also has been a lot of work to learn from multiple modalities [12, 1, 11], like meshes and multi-view images, where the aim is to leverage information from all the modalities to encode the point clouds. Our work is based on auto-encoding [14] and reconstruction of point clouds [25] but existing works like Point-MAE [14], and Point-BERT [25] used masking as a pretext task for transformer based encoders that are trained to reconstruct the masked point cloud patch. In contrast to the masking approach, we are corrupting the point cloud randomly and train our model to reconstruct the original point cloud.

3. Methods

3.1. Point Cloud Perturbation

To introduce robustness in the latent representations of the point clouds, we first corrupt the input point clouds before the decoder is trained to reconstruct the original point cloud. An input point cloud consists of a set of 1024 points representing a shape. To prevent the loss of the structure as well as introduce sufficient corruption, we selected 20 centers using random sampling and apply nearest neighbors to select 20 nearest points for each of the centers to form a patch. For each patch, random Gaussian noise with zero mean and 0.03 standard deviation is applied to corrupt the point clouds. Therefore, for each point cloud, we perturb approximately 400 points out of 1024 points.

3.2. SeRP-PointNet

We modify the architecture of PointNet [16] auto-encoder to form SeRP-PointNet. We use PointNet architecture to learn a global 1024-dimensional vector representation of the perturbed point cloud. Similar to PointNet's segmentation network, the global vector representation is concatenated with the local features followed by 4 fully-connected layers to give per-point 128-dimensional features. To pre-train the SeRP-PointNet auto-encoder we use two tasks:

- **Classification:** Classify which points are perturbed. It is performed by a classification head consisting of a fully-connected layer on top of the per-point features. We use a Cross-entropy loss function.

- **Reconstruction:** Reconstructing the perturbed points. We attach a fully-connected layer over the top of per-point features to form the reconstruction head. We provide two methods to learn reconstruction:

1. δ-learning: The reconstruction head predicts δ, i.e. the 3-coordinate differences between the ground truth point cloud and the perturbed point cloud. To calculate the loss, we use a Mean Squared Loss function.
2. $cd\ell_2$-learning: The reconstruction head predicts the 3-d coordinates directly from the per-point features. For this method, we use the Chamfer Distance ℓ_2 loss described in Section 3.3.5.

The total loss is calculated as:

$$\mathcal{L} = \alpha_{cls} * l_{cls} + \alpha_{rec} * l_{rec} \tag{1}$$

where, l_{cls} and l_{rec} are classification and reconstruction losses respectively. In our experiments, we set $\alpha_{cls} = 0.001$ and $\alpha_{rec} = 1.5$.

3.3. SeRP-Transformer

Transformer models [5] employ self-attention [20] mechanism to model long-range dependencies between different input tokens. In this section, we will detail our Autoencoder with Transformer backbone and how it is used for processing point cloud data.

3.3.1 Point Tokenization

Since the input set of points is large, it becomes computationally infeasible to use transformers based architectures because of the quadratic scaling of self-attention modules. To overcome this, we divide the corrupted point cloud into irregular patches via Farthest Point Sampling (FPS) and K-Nearest Neighbors (KNN) algorithm. Formally, given an input point cloud with p points called $X \in \mathbb{R}^{p\times 3}$, we sample a set of c centers called C, and then KNN is applied on those centers to form patches called the patch tokens, T containing n points in each patch.

$$C = FPS(X),\ C \in \mathbb{R}^{c\times 3} \tag{2}$$

$$T = KNN(C),\ T \in \mathbb{R}^{c\times n\times 3} \tag{3}$$

Furthermore, each of the patch token is normalized with respect to the mean and standard deviation of the points present in each patch to improve convergence.

3.3.2 Embedding

To learn the patch token embeddings, T_e, we use a lightweight PointNet architecture. Since the patches are center normalized, the position embeddings of the center coordinates are also appended to T_e. The position embeddings, P_e are learned using a small MLP networks. Both T_e and P_e are of same dimensions, d and T_e and P_e are concatenated to form the inputs embeddings T_i

$$T_e = \text{PointNet}(T),\ T_e \in \mathbb{R}^{c\times d} \tag{4}$$

$$P_e = \text{MLP}(C),\ P_e \in \mathbb{R}^{c\times d} \tag{5}$$

$$T_i = \text{Concatenate}[T_e; P_e] \in \mathbb{R}^{c\times 2d} \tag{6}$$

Following standard transformers [5], we also append a learnable [CLS] token and its learnable position embedding with the patch tokens T_i. The [CLS] token representations learns the overall point cloud structure and it is used in the downstream evaluation tasks. Therefore, the input, I, to the auto-encoder is the set of $c + 1$ vectors each of dimensions $2d$.

$$I = ([CLS]; T_i),\ I \in \mathbb{R}^{(c+1)\times 2d} \tag{7}$$

3.3.3 Encoder-Decoder

Our Autoencoder is based on standard Transformers with asymmetric encoder-decoder design as in [14]. Our transformer encoder takes the inputs and passes it through the self-attention layers to form low-dimensional representations of size ℓ, E. The decoder, then takes those representations and projects them back in the original subspace to give D.

$$E = \text{Encoder}(I),\ E \in \mathbb{R}^{(c+1)\times \ell} \tag{8}$$

$$D = \text{Decoder}(E),\ D \in \mathbb{R}^{(c+1)\times d} \tag{9}$$

3.3.4 Reconstruction Head

The reconstruction is a fully-connected layer, FC, that outputs the point reconstructions. The decoder output, D, is fed to FC to output $\hat{T} \in \mathbb{R}^{c\times 3n}$ where $\hat{T}$ consists of c vectors corresponding to each input patch token, and each reconstructed patch consists of n generated points, each with 3 coordinates.

$$\hat{T} = \text{FC}(D),\ \hat{T} \in \mathbb{R}^{c\times 3n} \tag{10}$$

We use Chamfer Distance ℓ_2 [6] reconstruction loss as described in Section 3.3.5.

The complete overview of the architecture is described in figure 2.

3.3.5 Loss function

Chamfer Distance ℓ_2 Loss: We use reconstruction loss given by ℓ_2 Chamfer Distance [6], where given an input

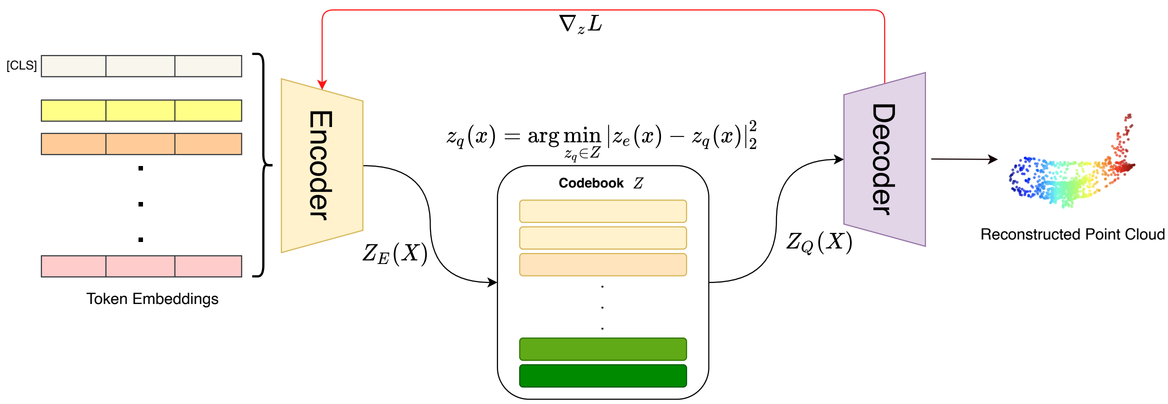

Figure 3: Overview of VASP. Vector Quantization of latent representations is done by maintaining a codebook $\mathcal{Z}$ of learnable discreet latent vectors. The output of the encoder $z_e(x)$ is passed through the codebook that returns the vector $z_q(x)$ closest to $z_e(x)$. As this operation is not differentiable, the gradients of the decoder are copied to the encoder during backpropagation.

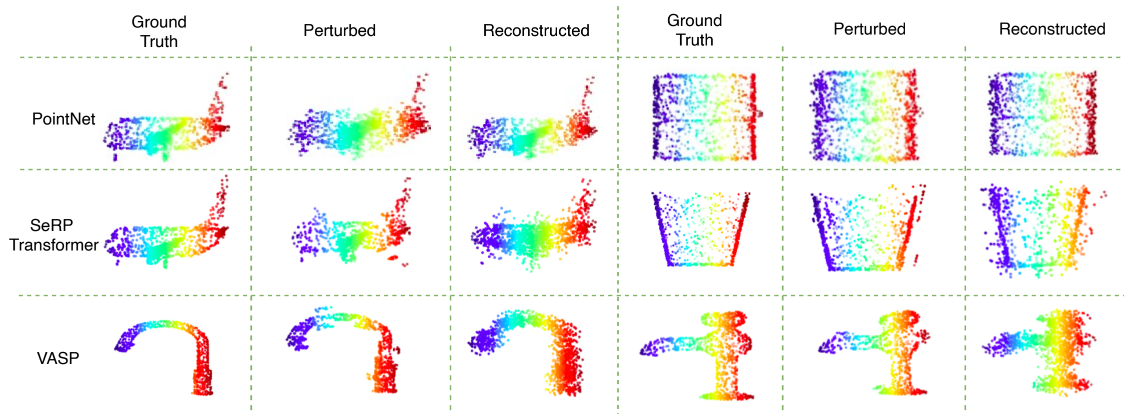

Figure 4: Reconstructions of the ShapeNet validation set: Above figure shows some examples of the original point cloud, corrupted point cloud and the reconstructed point by all the three autoencoders that we proposed. We can see that all the models are able to recover the shape from the latent representations.

point cloud, P and reconstructed point cloud $\hat{P}$, the loss is given by

$$\mathcal{L}(P, \hat{P}) = \frac{1}{|\hat{P}|} \sum_{x \in \hat{P}} \min_{y \in P} \|x - y\|_2^2 + \frac{1}{|P|} \sum_{x \in P} \min_{y \in \hat{P}} \|x - y\|_2^2 \tag{11}$$

3.4. VASP: Vector-Quantized Autoencoder for Self-supervised Representation Learning for point Clouds

Generally, lots of real-world data can be modeled by discrete representations like language, human speech as well as images. Furthermore, architectures like Transformers are designed to work on discrete datasets. Therefore, in order to extend the SeRP-Transformer autoencoder for doing discrete variational inference, we introduce a codebook, $\mathcal{Z}$ that contains a set of K discrete latent vectors, $e_1, \ldots, e_K$, of dimension d which is same the dimensions of encoder output. As shown in Figure 3, the output of the encoder, $z_e(x)$, are passed through this codebook and the discrete latent variable, $z_q(x)$, is computed by the nearest neighbor look-up in the embedding space.

$$z_q(x) = e_k \quad \text{where } k = \arg\min_j ||z_e(x) - e_j||^2 \tag{12}$$

The latent vector, is then passed through the decoder for reconstruction.

Learning: Note that, selecting the latent vector from the codebook, is not a differentiable operation. Therefore, during backpropagation, the gradients of the decoder are copied directly to the encoder. Since the outputs of the encoder and the inputs of the decoder are in the same representational space, the gradients contain usefull information about how the encoder weights should be updated for reconstruction [19]. The full objective is given by equation 13

$$\mathcal{L}_{VQ} = \log p(x|z_q(x)) + \alpha ||sg[z_e(x)] - e||_2^2 + \beta ||z_e(x) - sg[e]||_2^2 \tag{13}$$

where $sg[.]$ denotes the stop gradient operation. The first term is the reconstruction loss given by L2 Chamfer distance given equation 11. The second term of the equation 13 makes the embeddings closer to the encoder output and thus, induces learning of the discrete latent space without updating $z_e(x)$. The third term is called the commitment loss function because it makes the output of the encoder commit to an embedding and the embedding is not updated in this term because of the stop-gradient operation. The commitment loss is added because the output space of the encoder is not constrained and thus, the encoder outputs can keep growing if the discrete embeddings are not learned as fast the encoder representations.

4. Experiments and Results

For SeRP-Transformer, we use $c = 64$ centers and each center consists of $n = 32$ nearest neighbors. The value of d was set to 384, therefore, the input T_i was 786 dimensions and latent representations size was set to $\ell = 384$.

4.1. Pretraining Setup

We use ShapeNet [2] dataset for pre-training our autoencoders. ShapeNet dataset is a large point cloud dataset containing approximately 50,000 models spanning across 55 categories. The dataset contains dense points clouds and for training we sample 1024 points using Farthest Point Sampling (FPS) as the input point cloud on which we apply perturbation before feeding it to the auto-encoder. Training was done using AdamW optimizer with initial learning rate of 0.001 with cosine annealing with a batch size of 128. SeRP-PointNet was trained for 100 epochs, whereas SeRP-Transformer was trained for 300 epochs.

The reconstruction results are shown in Figure 4 for all the three autoencoders - SeRP-PointNet, SeRP-Transformer and VASP Transformer. In the figure, we can see that the autoencoders are able to recover the shape of the original point cloud approximately, and therefore, indicating learning of informative latent representations.

4.2. Downstream Evaluation

We evaluate the quality of the learned latent representations by taking the pre-trained encoder and finetuning it on another dataset for classification. We finetune our pre-trained encoder on ModelNet40 dataset which contains CAD models of 40 categories.

Dataset / Model	ModelNet40		ShapeNet	
	Accuracy	Gain	Accuracy	Gain
Scratch	88.48	-	87.43	-
SeRP-Net	**89.1**	0.62 ↑	**87.97**	0.54 ↑
VASP	87.85	-0.63 ↓	86.54	-0.89 ↓

Table 1: Downstream Task Evaluation for SeRP-Transformer variants. *Scratch = Non-pre-trained model.*

Dataset / Model	ModelNet40		ShapeNet	
	Accuracy	Gain	Accuracy	Gain
Scratch	82.97	-	84.24	-
δ **learn**	**84.1**	1.13 ↑	**84.43**	0.19 ↑
$cd\ell_2$ **learn**	84.06	1.09 ↑	84.39	0.15 ↑

Table 2: Downstream Task Evaluation for SeRP-PointNet variants. *Scratch = Non-pre-trained model.*

4.2.1 Downstream Evaluation Results

We present the results of our downstream evaluation in Tables 1 and 2. From the results, we observe that the encoders of SeRP-PointNet and SeRP Transformer outperform the non-pre-trained models i.e., the models trained from scratch on ModelNet40.

From the table, we can also see that the VASP Transformer shows worse results than the non-pre-trained encoder, indicating that the encoder, trained using discrete codes, is not able to learn robust representations and therefore, necessitating more work towards discrete representation learning of point clouds.

5. Conclusion and Future Work

In this work, we presented a self-supervised learning framework to learn representations of point cloud data. We trained autoencoder models that aim to reconstruct the original point cloud using perturbed point clouds as inputs and thus, learning low-dimensional latent space in the process. We showed that pre-trained models perform better than the encoders trained from scratch on a downstream classification task.

For the future work, we can propose a more challenging strategy to perturb point clouds by sampling centers using FPS instead of random sampling, that would encourage corruption at the edges than inside the volume. We can also compare the existing approach with traditional variational inference and discrete variational inference methods.

References

[1] Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. *arXiv preprint arXiv:2203.00680*, 2022. 2

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1, 5

[3] Yujin Chen, Matthias Nießner, and Angela Dai. 4dcontrast: Contrastive learning with dynamic correspondences for 3d scene understanding. *arXiv preprint arXiv:2112.02990*, 2021. 1, 2

[4] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020. 1

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1, 2, 3

[6] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 3

[7] Kexue Fu, Peng Gao, Renrui Zhang, Hongsheng Li, Yu Qiao, and Manning Wang. Distillation with contrast is all you need for self-supervised point cloud representation learning. *arXiv preprint arXiv:2202.04241*, 2022. 2

[8] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020. 1

[9] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 1

[10] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6535–6545, 2021. 1, 2

[11] Longlong Jing, Yucheng Chen, Ling Zhang, Mingyi He, and Yingli Tian. Self-supervised modal and view invariant feature learning. *arXiv preprint arXiv:2005.14169*, 2020. 2

[12] Longlong Jing, Zhimin Chen, Bing Li, and Yingli Tian. Self-supervised modality-invariant and modality-specific feature learning for 3d objects. 2021. 1, 2

[13] Benedikt Mersch, Xieyuanli Chen, Jens Behley, and Cyrill Stachniss. Self-supervised point cloud prediction using 3d spatio-temporal convolutional networks. In *Conference on Robot Learning*, pages 1444–1454. PMLR, 2022. 2

[14] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. *arXiv preprint arXiv:2203.06604*, 2022. 1, 2, 3

[15] Omid Poursaeed, Tianxing Jiang, Han Qiao, Nayun Xu, and Vladimir G Kim. Self-supervised learning of point clouds via orientation estimation. In *2020 International Conference on 3D Vision (3DV)*, pages 1018–1028. IEEE, 2020. 2

[16] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1, 2

[17] Jason Tyler Rolfe. Discrete variational autoencoders. *arXiv preprint arXiv:1609.02200*, 2016. 2

[18] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[19] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 1, 5

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[21] Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J Kusner. Unsupervised point cloud pre-training via occlusion completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9782–9792, 2021. 2

[22] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 1

[23] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pretraining for 3d point cloud understanding. In *European conference on computer vision*, pages 574–591. Springer, 2020. 1, 2

[24] Siming Yan, Zhenpei Yang, Haoxiang Li, Li Guan, Hao Kang, Gang Hua, and Qixing Huang. Implicit autoencoder for point cloud self-supervised representation learning. *arXiv preprint arXiv:2201.00785*, 2022. 2

[25] 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. *arXiv preprint arXiv:2111.14819*, 2021. 1, 2

[26] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10252–10263, 2021. 2