Masked Scene Contrast: A Scalable Framework for Unsupervised 3D Representation Learning

Xiaoyang Wu Xin Wen Xihui Liu Hengshuang Zhao*
The University of Hong Kong

https://github.com/Pointcept/Pointcept

Abstract

As a pioneering work, PointContrast conducts unsupervised 3D representation learning via leveraging contrastive learning over raw RGB-D frames and proves its effectiveness on various downstream tasks. However, the trend of large-scale unsupervised learning in 3D has yet to emerge due to two stumbling blocks: the inefficiency of matching RGB-D frames as contrastive views and the annoying mode collapse phenomenon mentioned in previous works. Turning the two stumbling blocks into empirical stepping stones, we first propose an efficient and effective contrastive learning framework, which generates contrastive views directly on scene-level point clouds by a well-curated data augmentation pipeline and a practical view mixing strategy. Second, we introduce reconstructive learning on the contrastive learning framework with an exquisite design of contrastive cross masks, which targets the reconstruction of point color and surfel normal. Our Masked Scene Contrast (MSC) framework is capable of extracting comprehensive 3D representations more efficiently and effectively. It accelerates the pre-training procedure by at least 3× and still achieves an uncompromised performance compared with previous work. Besides, MSC also enables large-scale 3D pre-training across multiple datasets, which further boosts the performance and achieves state-of-the-art fine-tuning results on several downstream tasks, e.g., 75.5% mIoU on ScanNet semantic segmentation validation set.

1. Introduction

Unsupervised visual representation learning aims at learning visual representations from vast amounts of unlabeled data. The learned representations are proved to be beneficial for various downstream tasks like segmentation and detection. It has attracted lots of attention and achieved remarkable progress in 2D image understanding, exceeding the upper bound of human supervision [20, 25].

*Corresponding Author. Email: hszhao@cs.hku.hk

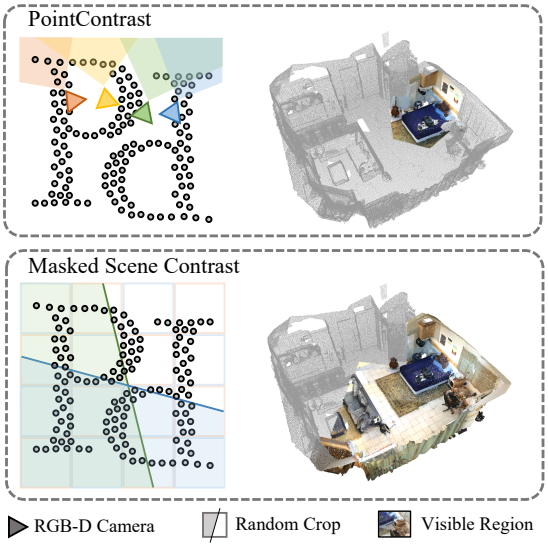


Figure 1. Comparison of unsupervised 3D representation learning. The previous method [56] (top) relies on raw RGB-D frames with restricted views for contrastive learning, resulting in low efficiency and inferior versatility. Our approach (bottom) directly operates on scene-level views with contrastive learning and masked point modeling, leading to high efficiency and superior generality, further enabling large-scale pre-training across multiple datasets.

Despite the impressive success of unsupervised visual representation learning in 2D, it is underexplored in 3D. Modern 3D scene understanding algorithms [10, 52] are focused on supervised learning, where models are trained directly from scratch on targeted datasets and tasks. Well-pre-trained visual representations can undoubtedly boost the performance of these algorithms and are currently in urgent demand. Recent work PointContrast [56] conducts a preliminary exploration in 3D unsupervised learning. However, it is limited to raw RGB-D frames with an inefficient learning paradigm, which is not scalable and applicable to large-scale unsupervised learning. To address this essential and inevitable challenge, we focus on building a scalable framework for large-scale 3D unsupervised learning.

One technical stumbling block towards large-scale pre-training is the inefficient learning strategy introduced by

matching RGB-D frames as contrastive views. PointContrast [56] opens the door to pre-training on real indoor scene datasets and proposes frame matching to generate contrastive views with natural camera views, as in Figure 1 top. However, frame matching is inefficient since duplicated encoding exists for matched frames, resulting in limited scene diversity in batch training and optimization. Meanwhile, not all of the 3D scene data contains raw RGB-D frames, leading to failure deployments of the algorithm. Inspired by the great success of SimCLR [7], we investigate generating strong contrastive views by directly applying a series of well-curated data augmentations to scene-level point clouds, eliminating the dependence on raw RGB-D frames, as in Figure 1 bottom. Combined with an effective mechanism that mixes up query views, our contrastive learning design accelerates the pre-training procedure by 4.4× and achieves superior performance with purely point cloud data, compared to PointContast with raw data. The superior design also enables large-scale pre-training across multiple datasets like ScanNet [16] and ArkitScenes [5].

Another obstacle is the mode collapse phenomenon that occurs when scaling up the optimization iterations. We owe the culprit for this circumstance to the insufficient difficulty of unsupervised learning tasks. To further tackle the mode collapse challenge in unsupervised learning and scale up the optimization iterations, inspired by recent masked autoencoders [23, 57], we construct a masked point modeling paradigm where both point color reconstruction objective and surfel normal reconstruction objective are proposed to recover the masked color and geometric information of the point cloud respectively. We incorporate the mask point modeling strategy into our contrastive learning framework via an exquisite design of contrastive cross masks, leading towards a scalable unsupervised 3D representation learning framework, namely Masked Scene Contrast (MSC).

Our framework is efficient, effective, and scalable. We conduct extensive experimental evaluations to validate its capability. On the popular point cloud dataset ScanNet, our algorithm accelerates the pre-training procedure by more than 3×, and achieves better performance on downstream tasks, when compared to the previous representative PointContrast. Besides, our method also enables large-scale 3D pre-training across multiple datasets, leading to state-of-the-art fine-tuning results on several downstream tasks, e.g. 75.5% mIoU on ScanNet semantic segmentation validation set. In conclusion, our work opens up new possibilities for large-scale unsupervised 3D representation learning.

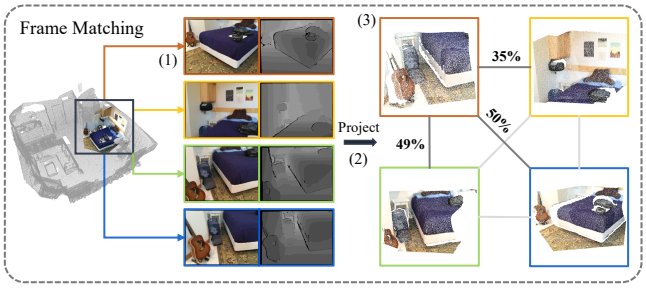
2. Related Work

2D Image contrastive learning. Based on the instance discrimination [17] pretext task, and combined with the contrastive learning [27, 47] paradigm, modern variants of 2D image contrastive representation learning have shown strong abilities in learning transferable visual representations [7, 24, 54]. With the learning objective built on the similarity between randomly augmented image views, this line of work strongly depends on a large batch size [7, 24] and a finely designed data augmentation pipeline [7, 20, 46] to achieve better performance. We find these two points also hold true for 3D contrastive learning.

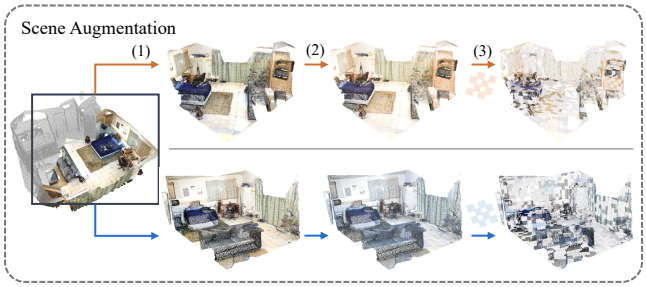
2D Image reconstructive learning. In 2D unsupervised learning, there is also a recent trend of switching the pretext task from instance discrimination [4, 6, 7, 20, 24] to masked image modeling [3, 23, 50, 57, 60]. Based on a denoising autoencoder [48]-style architecture, the task is to reconstruct the RGB value [23, 57], discrete token [3, 60], or feature [50] of masked pixels. This line of work has shown strong potential in learning representations on large-scale datasets and is less prone to model collapsing like instance-discrimination-based methods (e.g., contrastive learning). When combined with contrastive learning [2, 18, 53], the performance can be further boosted, yet with less dependence on data scale [18].

3D Scene understanding. The deep neural architectures for understanding 3D scenes can be roughly categorized into three paradigms according to the way they model point clouds: projection-based, voxel-based, and point-based. Projection-based works project 3D points into various image planes and adopt 2D CNN-based backbones to extract features [9, 30, 31, 43]. On the other hand, the voxel-based stream transforms point clouds into regular voxel representations to operate 3D convolutions [33, 42]. Their efficiency is then improved thanks to sparse convolution [10, 13, 19]. Point-based methods, in contrast, directly operate on the point cloud [12, 37, 38, 44, 58], and see a recent transition towards transformer-based architectures [21, 52, 59]. Following [56], we mainly pre-train on the voxel-based method SparseUNet [10] implemented with SpConv [15].

3D Representation learning. Unlike the 2D counterparts, where large-scale unsupervised pre-training has been a common choice for facilitating downstream tasks [6], 3D representation learning is still not mature, and most works still train from scratch on the target data directly [28]. While earlier works in 3D representation learning simply build on a single object [22, 40, 41, 49], recent works start to train on scene-centric point clouds [28, 56]. However, unlike in 2D that scene-centric representation learning has been well-studied [32, 51, 55], the pre-training on 3D scenes, which relies on raw frame data [28, 56], still faces inefficiency issues and finds it hard to scale up to larger scale datasets. In contrast, we explore directly learning at the scene level, which shows significantly higher efficiency in processing scene data, and opens the possibility for pre-training with larger-scale point clouds, for the first time ever.



(a) **Frame matching [28, 56].** 1. Extract raw RGB-D frames and camera positions from raw data. 2. Project each 2D frame into 3D space produces frame-level point cloud views. 3. Calculate pairwise overlapping rates among each frame of a single view and select pairs with overlapping rates larger than 30% as pairs of contrastive views.



(b) **Scene augmentation (ours).** 1. Apply spatial augmentations containing rotation, flipping, and scaling. 2. Apply photometric augmentations containing brightness, contrast, saturation, hue, and gaussian noise jittering. 3. Generate and apply contrastive cross masks to the two views after sampling augmentations containing cropping and voxelization.

Figure 2. **View generation.** Compared with frame matching (FM), our scene augmentation (SA) is efficient and effective. 1. SA can end-to-end produce contrastive views on the original point cloud with ignorable latency, while FM requires preprocessing devouring enormous storage resources (e.g. additional 1.5 TB storage for ScanNet) in step 2 and pairwise matching is time-consuming. 2. SA produces scene-level views, while FM can only produce frame-level views containing limited information. Benefiting from advanced photometric augmentations, SA has the capacity to simulate the same scene under different lighting.

3. Pilot Study

This section analyzes the two main obstacles towards large-scale pre-training with point clouds. Our proposed design is based on the conclusion of the pilot study.

Is matching RGB-D frames a good choice?

As a seminal work in 3D representation learning, PointContrast [56] first enables pre-training in real-world indoor scenes with matched raw RGB-D frames as contrastive views. A visualization of the frame matching procedure is illustrated in Figure 2a. This protocol seems natural for indoor scenes since point clouds of indoor scenes are usually derived from RGB-D videos [1,5,16], where the raw frames are extracted from. However, this framework has multiple drawbacks that can hinder the scalability of training:

- *Redundant frame encoding.* The pairwise matching strategy that PointContrast adopts allows one frame to be matched multiple times. As a result, each frame can be encoded multiple times in one step, adding to redundancy in training.
- *Low learning efficiency.* In one training step, the frame matching strategy only allows the framework to process several views of a single scene. Therefore the amount of information that PointContrast can process in one step is rather limited, and the overall time for one training cycle is also notably high.
- *Dependency on raw RGB-D frames.* The whole framework is built on the assumption that RGB-D videos are available, yet this is not true for every publicly-available point cloud dataset. Even when available, the storage cost of RGBD frames is also significantly higher than the reconstructed point cloud data.

Consequently, pre-training frameworks [28, 56] based on matching frames as contrastive samples require enormous computing and storage resources. For example, PointContrast sub-samples RGB-D scans from the raw ScanNet videos every 25 frames, consuming ∼30 times the storage of the processed point clouds, and takes 80 GPU hours to process an epoch of 1500 scenes. Even at such cost, our experiments in Table 1a verify that the raw RGB-D frames cannot bring additional information over the processed point clouds to achieve a better representation.

We will explore the possibility of pre-training on the point clouds directly.

What's the revelation behind mode collapse?

Mode collapse, defined as the phenomenon that all features collapse to a single vector, remains an unsolved problem accompanying the development of 3D representation learning [11, 56]. To alleviate this problem, PointContrast introduced InfoNCE loss [35], which has been shown to stabilize training, to replace the hardest-contrastive loss. Yet, the problem of mode collapse can still occur when the amount of training data and the length of the training schedule increase. Given the empirical conclusion of 2D contrastive learning [8], the occurrence of mode collapse is unusual under the premise that a large number of negative samples are already adopted. Interestingly, we notice that the mean negative pair cosine similarity of previous works is mostly close to 0, indicating that the negative samples are mostly easy and thus have little penalty towards the trivial solution. Although the InfoNCE loss alleviates this problem with an alternated optimization objective, we argue that a more desirable solution can be achieved by raising the difficulty of the unsupervised pretext task.

We will further raise the difficulty of the pretext task to solve the mode collapse problem.

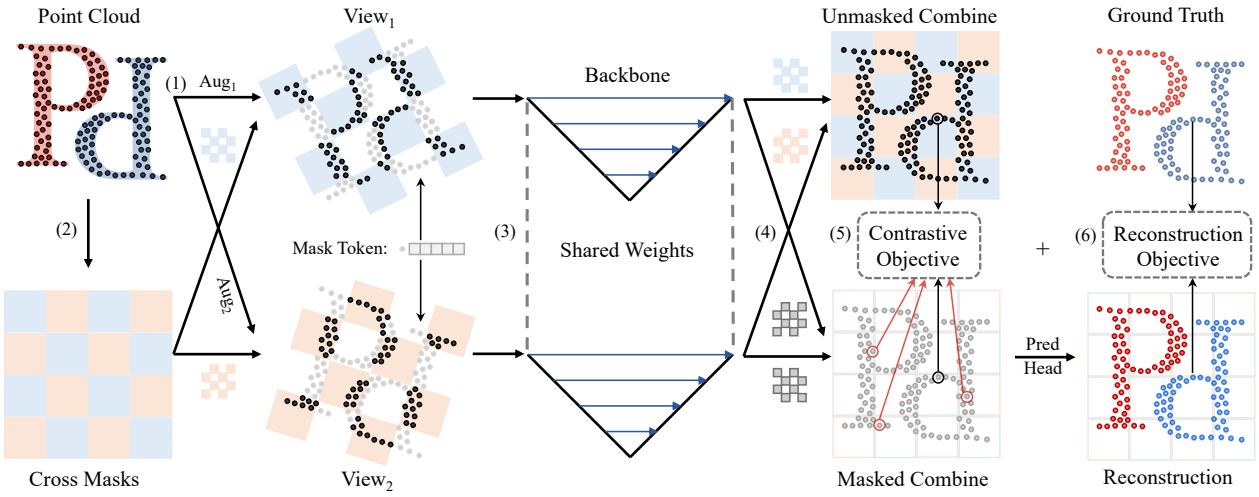


Figure 3. **Our MSC framework**. (1) Generating a pair of contrastive views with a well-curated data augmentation pipeline consisting of photometric, spatial, and sampling augmentations. (2) Generating a pair of complementary masks and applying them to the pair of contrastive views. Replacing masked point features with a learnable mask token vector. (3) Extracting point representation with a given U-Net style backbone for point cloud understanding. (4) Reassembling masked contrastive views to masked points combination and unmasked points combination. (5) Matching points share similar positional relationships in the two views as positive sample pairs and computing InfoNCE loss to optimize contrastive objective. (6) Predicting masked point color and normal and computing Mean Squared Error loss and Cosine Similarity Loss with ground truth respectively, to optimize the reconstruction objective.

4. Approach

Based on the analysis of the stumbling blocks for large-scale pre-training in Section 3, we first introduce our optimized contrastive learning design in Section 4.1 to make the process more efficient. Then we solve the long-term problem of mode collapse with an additional reconstructive learning design in Section 4.2. The final optimization target is described in Section 4.3. Combining these exquisite designs, we build the whole framework, namely *Masked Scene Contrast* (MSC), and a visual illustration of our MSC is available in Figure 3.

4.1. Contrastive Learning

Framework. Different from the previous protocol of matching RGB-D frames decomposed from indoor scenes, our contrastive learning framework directly operates on the point cloud data. Given a point cloud $\boldsymbol{X} = (\boldsymbol{P}, \boldsymbol{C})$, where $\boldsymbol{P} \in \mathbb{R}^{n\times 3}$ represents the spatial features (coordinate) of the points and $\boldsymbol{C} \in \mathbb{R}^{n\times 3}$ represents the photometric features (color) of the points, the contrastive learning framework can be summarized as follows:

- *View generation.* For a given point cloud $\boldsymbol{X}$, we generate query view $\boldsymbol{X}_r$ and key view $\boldsymbol{X}_k$ of the original point cloud with a sequence of stochastic data augmentations, which includes photometric, spatial, and sampling augmentations.
- *Feature extraction.* Encoding point cloud features $\boldsymbol{F}_r$ and $\boldsymbol{F}_k$ with a U-Net style backbone $\zeta(\cdot)$ to $\hat{\boldsymbol{F}}_r$ and $\hat{\boldsymbol{F}}_k$ respectively.
- *Point matching.* The positive samples of contrastive learning are point pairs with close spatial positions in the two views. For each point belonging to the query view, we calculate the correspondence mapping $\mathcal{P} = \{(i, j)\}_{n'}$ to points of the key view. If $(i, j) \in \mathcal{P}$ then point $(\boldsymbol{p}_i, \boldsymbol{c}_i)$ and point $(\boldsymbol{p}_j, \boldsymbol{c}_j)$ constructs a pair across two views.
- *Loss computation.* Computing the contrastive learning loss on the representation of two views $\hat{\boldsymbol{F}}_r$ and $\hat{\boldsymbol{F}}_k$ and the correspondence mapping $\mathcal{P}$. An encoded query view should be similar to its key view.

Data augmentation. As a pioneering work in image contrastive learning, SimCLR [7] reveals that a well-curated data augmentation pipeline is crucial for learning strong representations. Unlike supervised learning, contrastive learning requires much stronger data augmentations to prevent trivial solutions. However, an effective data augmentation recipe is still absent in 3D representation learning. The frame matching scheme in prior works [28, 56] simply applies a randomly rotating operator to contrastive targets. Even if the RGB-D frames can be viewed as a natural random crop, the augmentation space is still far from diverse enough. The pretext task it forms is not yet challenging enough to facilitate the contrastive learning framework to learn robust representations for downstream tasks.

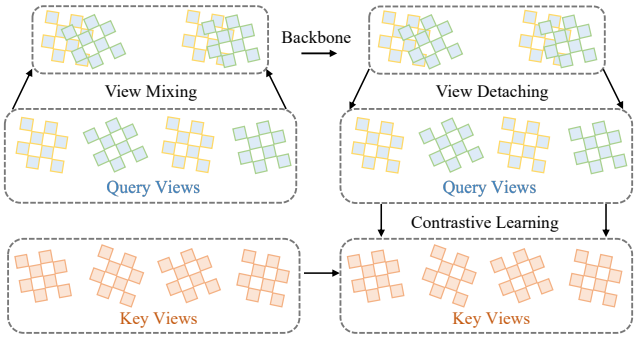


Figure 4. **View Mixing**. Randomly mix up query views while keeping key views unmixed for a given batch of pairwise contrastive views. Detaching mixed query view after feature extraction for contrastive comparison with matched key views.

As presented in Figure 2b, our well-designed stochastic data augmentation pipeline includes photometric augmentations, spatial augmentations, and sampling augmentations. Inspired by the advanced photometric augmentation validated by our 2D counterparts [7, 20], we further strengthen the photometric augmentation component introduced by Choy et al. [10] with random brightness, contrast, saturation, hue, and gaussian noise jittering for photometric augmentation. Besides that, random rotating, flipping, and scaling constitute our spatial augmentations, and the sampling augmentation is composed of random cropping and grid sampling.

Empirically, the order of data augmentations is also a key component of our recipe. For example, grid sampling after random rotation leads to cross grids for sampling, which further increases the distinction between contrastive views and has a better augmentation effect. The specific data augmentation settings are available in the appendix, and a comparison with previous methods is presented in Figure 2.

View mixing. Recently, Nekrasov et al. [34] proposes a data augmentation technique for 3D understanding models by mixing two scenes as a hybrid training sample, which can significantly suppress model overfitting. Inspired by the mixing mechanism, we integrate the logic of mixing as part of the contrastive learning objective. As illustrated in Figure 4, for a batch of pairwise views, we randomly mix up the query views while maintaining the key views unchanged before the *feature extraction* process. The simple operation can effectively increase the robustness of the backbone and improve the robustness of the point cloud representation.

Contrastive target. We follow the design of PointContrast on the contrast target and apply InfoNCE loss to the matched points. Given correspondence mapping $\mathcal{P} = \{(i, j)\}_{n'}$ produced by *point matching* and points representation $\hat{\boldsymbol{F}}_r$ and $\hat{\boldsymbol{F}}_k$ embedded during *feature extraction*, the contrastive loss is:

$$s_{ij} = \frac{\boldsymbol{f}_{i'}^{rT} \boldsymbol{f}_{j'}^{k}}{||\boldsymbol{f}_{i'}^{rT}|| \cdot ||\boldsymbol{f}_{j'}^{k}||}, \tag{1}$$

$$\mathcal{L}_{\text{InfoNCE}} = \sum_{i}^{n} -\log \frac{\exp(s_{ii}/\tau)}{\sum_{j}^{n} \exp(s_{ij}/\tau)}, \tag{2}$$

note that $\boldsymbol{S} = \{s_{ij}\} \in \mathbb{R}^{n \times n}$ is the pairwise cosine similarity matrix between positive samples and negative samples, while τ is the temperature factor scaling cosine similarity. In practice, we control temperature factor τ as 0.4, which is the same as previous works [28, 56].

4.2. Reconstructive Learning

As is mentioned in Section 3, one of the stumbling blocks for large-scale representation is mode collapse, and our solution is to scale up the difficulty of the unsupervised pre-training task. Motivated by the success of masked image modeling [23, 57] in 2D representations, we propose masked point modeling, which can be naturally integrated into our contrastive learning framework. Benefiting from this design, our framework can fully use non-overlapped regions of contrastive views that cannot be utilized by contrastive learning.

Contrastive cross mask. The key design that enables additional construction learning in our contrastive learning framework is the contrastive cross mask. For a given query view and key view of a single point cloud, we partite the unioned point set into non-overlapping grid partitions by their original position before spatial augmentation. Given a mask rate r range from 0 to 0.5, we randomly generate a pair of masks $\boldsymbol{M}_r, \boldsymbol{M}_k \in \mathbb{R}^{1 \times n_{r,k}}$, in which there are no shared masked patches. Then, we follow the practice of SimMIM [57] to apply the pair of masks to the two views respectively by replacing the input feature with a learnable mask token vector $\boldsymbol{t} \in \mathbb{R}^c$. Consequently, the feature extraction process can be rewritten as follows:

$$\hat{\boldsymbol{F}}_{r,k} = \zeta((1 - \boldsymbol{M}_{r,k})\boldsymbol{F}_{r,k} + \boldsymbol{M}_{r,k}\boldsymbol{T}_{r,k}), \tag{3}$$

where $\boldsymbol{T}_{r,k} \in \mathbb{R}^{n_r, n_k \times c}$ is the expand matrix of mask token vector $\boldsymbol{t}$ to fit the feature dimensions.

Reconstruction target. The features of the point cloud are composed of two parts, the coordinates that determine the geometric structure and the colors that represent the texture features. We build up reconstruction targets for the two groups of features separately.

The reconstruction of point cloud texture is straightforward, we predict the photometric value of each point with a linear projection. We compute the mean squared error (MSE) between the reconstructed and original color of masked points as the color reconstruction loss:

$$\mathcal{L}_{\text{c}} = \frac{\sum_i^{n_r} m_i^r ||\boldsymbol{x}_i^r - \hat{\boldsymbol{x}}_i^r||_2^2 + \sum_i^{n_k} m_i^k ||\boldsymbol{x}_i^k - \hat{\boldsymbol{x}}_i^k||_2^2}{n_r' + n_k'}, \tag{4}$$

where n'_r and n'_k represent the number of mask points belonging to refer view and key view, m^i_r and m^i_k mean the i-th element of M_r and M_k respectively.

Point coordinates play an important role in describing the geometric structure of point clouds, and it is worth noting that directly reconstructing the coordinates of masked points is not reasonable since masked points are only sampled from 3D object surface rather than the continuous surface itself. Reconstructing points coordination would lead to an overfitted representation. To overcome the challenge, we introduce the concept of surfel reconstruction. Surfel is an abbreviation for a *surface element* or *surface voxel* in the discrete topology literature [26] and primitives rendering [36]. For each masked point, we reconstruct the normal vector of the corresponding surfel and compute the mean cosine similarity between estimations and surfel normals as a contrastive loss:

$$\mathcal{L}_{\text{n}} = \frac{\sum_i^{n_r} m_i^r \boldsymbol{x}_i^{rT} \hat{\boldsymbol{x}}_i^r + \sum_i^{n_k} m_i^k \boldsymbol{x}_i^{kT} \hat{\boldsymbol{x}}_i^k}{n'_r + n'_k}, \tag{5}$$

where n'_r and n'_k represent the number of mask points belonging to refer view and key view, m^i_r and m^i_k mean the i-th element of M_r and M_k respectively.

4.3. Loss Function

Our framework combines the contrastive target, the color reconstruction target, and the surfel reconstruction to make the unsupervised task more scalable. The overall loss function is a weighted sum of Eq. 2, Eq. 4, and Eq. 5 which is written as follow:

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{\text{InfoNCE}} + \lambda_c \mathcal{L}_{\text{c}} + \lambda_n \mathcal{L}_{\text{n}}, \tag{6}$$

where λ_c and λ_n are the weight parameters that balance the three loss components. Empirically we find that performance is robust to the choice of weight parameters, and we make $\lambda_c = \lambda_n = 1$ in practice.

5. Experiments

We conduct extensive experimental evaluations to validate the capability of our framework, built upon the point cloud perception codebase *Pointcept* [14]. We first ablate our designs with an efficient pre-training pipeline that only utilizes ScanNet point cloud in Section 5.1, while without compromising performance. Then we explore large-scale pre-training across multiple datasets and compare our performance with previous results in Section 5.2.

5.1. Main Properties

We ablate the main designs and intriguing properties of our MSC in Table 1, and the default setting is available in the caption. We enable *efficient pre-training* by introducing our view generation pipeline, which is ablated in Table 1a.

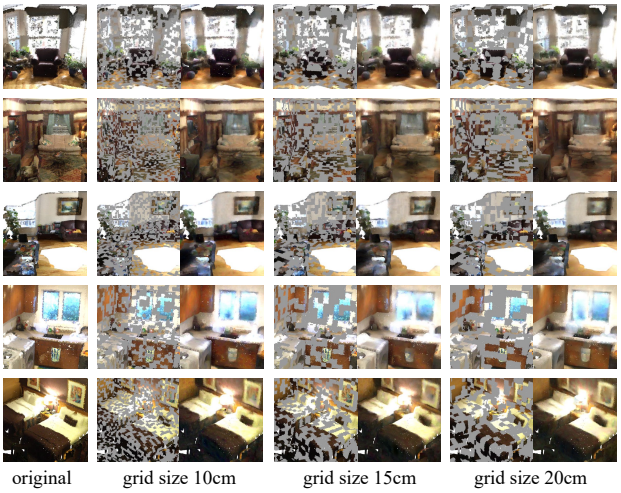


Figure 5. **Masked scenes and color reconstructions**. We visualize one of the cross masks of each scene with a mask rate of 50% (left) and color reconstruction of the masked point combinations (right). We pre-train our MSC with a mask patch size of 10cm and generalize to results to different mask sizes. Compared with the original point clouds, the loss of detail cannot be avoided, while boundary and texture are well preserved by our model.

All of our ablation experiments only require 20G ScanNet point cloud data and around 14 hours pertaining on a single machine containing 8 NVIDIA RTX3090.

View generation. In Table 1a, we show the results with different generation strategies of contrastive views. Unlike PointContrast [56], which builds on the raw RGB-D frames, our strategy directly utilizes the scene-level point cloud. This significantly reduces storage requirements (96% less) and allows more efficient use of the training data. With a 30% equivalent number of training iterations, our method can attain 120× training epochs, making full use of the training data for the first time. This results in 0.4 points higher FT mIoU with 4.4× speedup. When combined with an additional mask point modeling strategy, the performance can be further boosted by 0.6 points, yet still with a notable speedup.

Number of positive pairs. In Table 1b, we show the effects of different numbers of positive pairs. Our method sees consistent improvements with an increasing number of positive pairs, while for PointContrast [56], the information in a large batch of positive pairs cannot be effectively utilized. Our intuition is that the ability to process scene-level views, which are larger in scale and contain much more information than frame-level views, enables our method to take advantage of a larger number of positive pairs.

Data augmentation. In Table 1c, we analyze the effect of different data augmentation combinations. Adopting either spatial augmentation or photometric augmentation leads to sub-optimal performance, and the combination of

View generation methods	Pre-train data	Storage	Batch size	Iters	Epochs	FT mIoU (%)	Hours (h)	Speedup
Frame matching (PointContrast [56])	ScanNet Raw	500G	32	100k	5	74.0	48	1.0×
Scene augmentation w/o mask (ours)	ScanNet v2	20G	32	30k	600	74.4	**11**	**4.4**×
Scene augmentation w mask (ours)	ScanNet v2	20G	32	30k	600	**75.0**	14	3.4×

(a) **View generation.** Views produced by our enhanced data augmentation are stronger than the original RGB-D frames. Scene-level views can significantly speed up pre-training and make contrastive learning more effective. The performance can be further boosted with additional masked point modeling.

#Pos pairs	PC [56]	MSC (ours)
1024	73.8	74.3
2048	74.0	74.5
4098	73.7	74.9
8192	73.9	**75.0**

(b) **Number of positive pairs.** A larger amount of sampled positive pairs are necessary for scene-level views.

Spatial	Photometric	FT mIoU (%)
w/o aug	w/o aug	72.1
w aug	w/o aug	73.4
w/o aug	w aug	72.8
w aug	w aug	**74.4**

(c) **Data augmentation.** The combination of spatial and photometric augmentation makes the view generation pipeline comes to work.

Query view	Key view	FT mIoU (%)
w/o mix	w/o mix	74.1
w mix	w/o mix	**74.4**
w/o mix	w mix	74.2
w mix	w mix	73.7

(d) **View mixing.** Randomly mixing query views while leaving key views unmixed is a sweet point.

Mask	Task	FT mIoU (%)
w/o cross	w/o contrast	74.1
w cross	w/o contrast	74.4
w/o cross	w contrast	74.7
w cross	w contrast	**75.0**

(e) **Cross mask.** Masks containing shared masked patches have a negative influence on contrastive learning.

Mask grid size (m)	FT mIoU (%)
0.05	74.3
0.1	75.0
0.15	**75.0**
0.2	74.8

(f) **Mask grid size.** Our design works with mask patches with a grid size larger than 0.1m, and we consider 0.15m as a default setting.

Color	Normal	FT mIoU (%)
w/o	w/o	74.4
w	w/o	74.9
w/o	w	74.6
w	w	**75.0**

(g) **Reconstruction target.** Both targets have a positive effect, while color reconstruction has a dominant impact on indoor scenes.

Table 1. **Ablation experiments.** We adopt *SparseUNet* and *efficient* pre-training on ScanNet [16] point cloud data to ablate our designs. We report fine-turning (FT) mIoU (%) results on ScanNet 20 classes semantic segmentation as the default metric. If not specified, the default setting is as follows: the pre-training period is 600 epochs, the masking ratio is 30% and the masked patch has a grid size of 0.15m in the real-world space, view mixing probability is 0.8. All of our designs are enabled by default. Default settings are marked in gray.

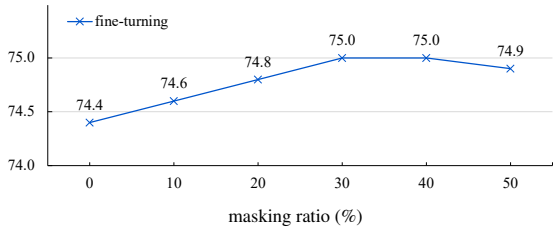


Figure 6. **Masking ratio**. A masking ratio ranging from 30% to 40% works well with our design, and a higher mask rate negatively influences contrastive learning. The y-axes represent ScanNet semantic segmentation validation mIoU (%).

both helps make our view generation pipeline come to work. Concerning relative significance, since the spatial augmentation is highly coupled with the masking strategy, the gain from it is higher than the photometric augmentation. But still, both are necessary.

View mixing. In Table 1d, we explore different configs for the view mixing strategy. Randomly mixing the query views while leaving the key views unchanged yields the best performance. Our intuition is that the key views, which form the vocabulary of the InfoNCE loss, should be relatively stable, thus reducing the ambiguity of the learning target. In contrast, introducing more diversity to the queries can be more helpful.

Cross mask. In Table 1e, we study the cross mask strategy. This strategy ensures that the two augmented views have no overlapping tokens. As reported in the table, whether with or without the contrastive learning target, this strategy ensures fewer shortcuts to the task and enables consistent downstream improvements.

Mask grid size. In Table 1f, we show the results ablating the grid size for producing masks on point clouds. Our design works with a grid size larger than 0.1m, and we consider 0.15m as a default setting. As shown in Figure 5, our reconstruction module is robust to extending mask grid size, which indicates that high quilty representation is captured during pre-training.

Masking ratio. In Figure 6, we depict the effect of the ratio of masked tokens. A masking ratio ranging from 30% to 40% works well with our design, and a higher masking rate has a negative impact on the overall performance. This varies from the conclusion of MAE [23], in which a higher masking ratio of 75% achieves top performance. Our hypothesis is that the contrastive learning objective built on the masked point clouds might favor a lower masking ratio, and the results in Figure 6 reflect a trade-off between the contrastive learning objective and the reconstructive objective. And pure reconstructive learning on point clouds is an interesting direction for future explorations.

Reconstruction target. In Table 1g, we ablate the effects of two components of our reconstruction target: color re-

Datasets	Backbones	Semantic Seg. (mIoU)			
		SC	PC [56]	CSC [28]	MSC (ours)
ScanNet	SparseUNet	72.2	74.1 (+1.9)	73.8 (+1.6)	**75.5** (+3.3)
ScanNet200	SparseUNet	25.0	26.2 (+1.2)	26.4 (+1.4)	**28.8** (+3.8)

(a) **Semantic segmentation.** We conduct pre-training on SparseUNet and compare semantic segmentation mIoU (%) results on ScanNet and ScanNet200 [39] validation set.

Datasets	Backbones	Instance Seg. (mAP@0.5)			
		SC	PC [56]	CSC [28]	MSC (ours)
ScanNet	SparseUNet	56.9	58.0 (+1.1)	59.4 (+2.5)	**59.6** (+2.7)
ScanNet200	SparseUNet	24.5	24.9 (+0.4)	25.2 (+0.7)	**26.8** (+2.3)

(b) **Instance segmentation.** We conduct pre-training on SparseUNet and compare instance segmentation mAP@0.5 (%) results driven by *PointGroup* [29] on ScanNet and ScanNet200 [39] validation set.

LR	Semantic Seg.				LA	Semantic Seg.			
Pct.	SC	CSC	VIBUS	MSC	Pts.	SC	CSC	VIBUS	MSC
100%	72.2	73.8	-	**75.3**	Full	72.2	73.8	-	**75.3**
1%	26.0	28.9	28.6	**29.2**	20	41.9	55.5	61.0	**61.2**
5%	47.8	49.8	47.4	**50.7**	50	53.9	60.5	65.6	**66.8**
10%	56.7	59.4	60.5	**61.0**	100	62.2	65.9	68.9	**69.7**
20%	62.9	64.6	64.8	**64.9**	200	65.5	68.2	69.6	**70.7**

(c) **Data efficiency.** We follow the ScanNet Data Efficient benchmark and compare the validation results SparseUNet with previous methods.

Table 2. **Results comparison.** We adopt cross-dataset pre-training utilizing ScanNet and ArkitScenes point cloud scenes for comparison of downstream task results. The pre-training setting is fixed as the default described in Table 1. More specific pre-training details are available in the Appendix. *SC* denotes train from scratch.

construction and normal reconstruction. Given the premise that indoor scenes are used, both targets show a positive effect on overall performance, while color reconstruction has a higher impact. The intuition is that the difference in texture reflected by color has a higher impact on the task of semantic segmentation, while the normal is helpful but has less influence (consider the same task on a 2D image).

5.2. Results Comparison

In this section, we extend the scale of pre-training by merging multiple datasets and compare downstream task fine-turning performance with previous unsupervised pre-training frameworks [28, 56]. Specifically, we adopt the default model setting ablated in Section 5.1 and train on both ScanNet [16] and ArkitScenes [5] point clouds, extending pre-training assets from 1,513 scenes to 6,560 scenes.

Semantic segmentation. In Table 2a, we report the semantic segmentation results on ScanNet and ScanNet200 [39] benchmark with SparseUNet and compare them with previous results. Our improvements are consistent and significant with larger pre-training assets for both benchmarks. With SparseUNet backbone, we outperform the current state-of-art pretraining framework by 1.7 points on ScanNet and 2.4 points on ScanNet200. Meanwhile, it is worth noting that driven by powerful MSC, we set a new best validation result on ScanNet semantic segmentation and pushed the previous SOTA to 75.5% with a baseline model.

Instance segmentation. In Table 2b, we report the instance segmentation results on ScanNet and ScanNet200 [39] with SparseUNet backbone driven by *PointGroup* [29]. Comparing them with previous results, we still see consistent improvements. Specifically, our framework achieves 59.6% on the ScanNet validation set, which is 3.3 points higher than training from scratch and 0.2 points promotion compared with the previous state-of-art performance. The boost is more significant on ScanNet200, which is 1.6 points higher than the previous SOTA.

Data efficiency. In Table 2c, we compare the ScanNet Data Efficient [28] results with previous methods. Our MSC shows consistently superior performance even compared with the latest data-efficient learning framework [45].

6. Conclusion and Discussion

In this paper, we tackle the problem of scalable unsupervised 3D representation learning. To this end, we present Masked Scene Contrast (MSC), an efficient, effective, and scalable framework that directly operates on scene-level views with contrastive learning and masked point modeling. Benefiting from the efficient scene-level point cloud processing pipeline and the effective training objectives, our method harvests high efficiency and superior generality, and enables large-scale pre-training across multiple datasets.

The key factor that empowers our method's scalability to larger-scale pre-training lies in the efficient pipeline that can directly learn from point cloud data, rather than the raw RGB-D frames. The efficiency, however, does not only mean processing data at scale. When only the standard dataset ScanNet is used for pre-training, our method still achieves uncompromised performance, yet with at least 3× speedup over previous works. This is especially meaningful considering the exhaustively long experimental time in the pre-training community, and it can better facilitate the verification of new ideas in future works.

It should also be noted that given the limit in computing resources, and the absence of a pre-training dataset that is sustainably at scale, the scalability of our method is not fully presented. In other words, our method opens the possibility of large-scale pre-training on 3D point cloud data for the first time, and we call for a large-enough 3D scene dataset that can fully unleash this potential. We hope our method can inspire future works that take the first step to real large-scale 3D pre-training, as the 2D community does.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China (No. 62201484), HKU Startup Fund, and HKU Seed Fund for Basic Research.

References

[1] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *CVPR*, 2016. 3

[2] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Michael G. Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. In *ECCV*, 2022. 2

[3] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *ICLR*, 2022. 2

[4] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022. 2

[5] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In *NeurIPS Workshops*, 2021. 2, 3, 8

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *CVPR*, 2021. 2

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2, 4, 5

[8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 3

[9] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *CVPR*, 2017. 2

[10] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, 2019. 1, 2, 5

[11] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *ICCV*, 2019. 3

[12] Ruihang Chu, Yukang Chen, Tao Kong, Lu Qi, and Lei Li. Icm-3d: Instantiated category modeling for 3d instance segmentation. *RA-L*, 2021. 2

[13] Ruihang Chu, Xiaoqing Ye, Zhengzhe Liu, Xiao Tan, Xiaojuan Qi, Chi-Wing Fu, and Jiaya Jia. Twist: Two-way inter-label self-training for semi-supervised 3d instance segmentation. In *CVPR*, 2022. 2

[14] Pointcept Contributors. Pointcept: A codebase for point cloud perception research. https://github.com/Pointcept/Pointcept, 2023. 6

[15] Spconv Contributors. Spconv: Spatially sparse convolution library. https://github.com/traveller59/spconv, 2022. 2

[16] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 2, 3, 7, 8

[17] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. In *TPAMI*, 2015. 2

[18] Alaaeldin El-Nouby, Gautier Izacard, Hugo Touvron, Ivan Laptev, Hervé Jegou, and Edouard Grave. Are large-scale datasets necessary for self-supervised pre-training? *arXiv:2112.10740*, 2021. 2

[19] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, 2018. 2

[20] 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. In *NeurIPS*, 2020. 1, 2, 5

[21] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 2021. 2

[22] Kaveh Hassani and Mike Haley. Unsupervised multi-task feature learning on point clouds. In *ICCV*, 2019. 2

[23] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 2, 5, 7

[24] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 2

[25] Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. In *ICML*, 2020. 1

[26] Gabor T Herman. Discrete multidimensional jordan surfaces. *CVGIP*, 1992. 6

[27] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2018. 2

[28] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *CVPR*, 2021. 2, 3, 4, 5, 8

[29] 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. 8

[30] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019. 2

[31] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3d lidar using fully convolutional network. In *RSS*, 2016. 2

[32] Songtao Liu, Zeming Li, and Jian Sun. Self-emd: Self-supervised object detection without imagenet. *arXiv:2011.13677*, 2020. 2

[33] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *IROS*, 2015. 2

[34] Alexey Nekrasov, Jonas Schult, Or Litany, Bastian Leibe, and Francis Engelmann. Mix3D: Out-of-Context Data Augmentation for 3D Scenes. In *3DV*, 2021. 5

[35] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv:1807.03748*, 2018. 3

[36] Hanspeter Pfister, Matthias Zwicker, Jeroen Van Baar, and Markus Gross. Surfels: Surface elements as rendering primitives. In *SIGGRAPH*, 2000. 6

[37] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 2

[38] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 2

[39] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *ECCV*, 2022. 8

[40] Aditya Sanghi. Info3d: Representation learning on 3d objects using mutual information maximization and contrastive learning. In *ECCV*, 2020. 2

[41] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. In *NeurIPS*, 2019. 2

[42] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, 2017. 2

[43] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *ICCV*, 2015. 2

[44] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*, 2019. 2

[45] Beiwen Tian, Liyi Luo, Hao Zhao, and Guyue Zhou. Vibus: Data-efficient 3d scene parsing with viewpoint bottleneck and uncertainty-spectrum modeling. *ISPRS*, 2022. 8

[46] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In *NeurIPS*, 2020. 2

[47] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv:1807.03748*, 2018. 2

[48] Pascal Vincent, H. Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR*, 2010. 2

[49] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *ICCV*, 2019. 2

[50] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *CVPR*, 2022. 2

[51] Xin Wen, Bingchen Zhao, Anlin Zheng, Xiangyu Zhang, and Xiaojuan Qi. Self-supervised visual representation learning with semantic grouping. In *NeurIPS*, 2022. 2

[52] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. In *NeurIPS*, 2022. 1, 2

[53] Zhirong Wu, Zihang Lai, Xiao Sun, and Stephen Lin. Extreme masking for learning instance and distributed visual representations. *arXiv:2206.04667*, 2022. 2

[54] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 2

[55] Jiahao Xie, Xiaohang Zhan, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Unsupervised object-level representation learning from scene images. In *NeurIPS*, 2021. 2

[56] 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, 2, 3, 4, 5, 6, 7, 8

[57] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *CVPR*, 2022. 2, 5

[58] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. Pointweb: Enhancing local neighborhood features for point cloud processing. In *CVPR*, 2019. 2

[59] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. In *ICCV*, 2021. 2

[60] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Image BERT pre-training with online tokenizer. In *ICLR*, 2022. 2