remote sensing

Article

PointUR-RL: Unified Self-Supervised Learning Method Based on Variable Masked Autoencoder for Point Cloud Reconstruction and Representation Learning

Kang Li [1], Qiuquan Zhu [1], Haoyu Wang [1], Shibo Wang [1], He Tian [1], Ping Zhou [2] and Xin Cao [1,*]

[1] School of Information Science and Technology, Northwest University, Xi'an 710127, China; likang@nwu.edu.cn (K.L.); zhuqiuquan@stumail.nwu.edu.cn (Q.Z.); wanghy@stumail.nwu.edu.cn (H.W.); wangshibo@stumail.nwu.edu.cn (S.W.); tianhe@stumail.nwu.edu.cn (H.T.)

[2] Emperor Qin Shihuang's Mausoleum Site Museum, Key Scientific Research Base of Ancient Polychrome Pottery Conservation, Xi'an 710600, China; bmyzhouping@sina.com

* Correspondence: caoxin@nwu.edu.cn

Abstract: Self-supervised learning has made significant progress in point cloud processing. Currently, the primary tasks of self-supervised learning, which include point cloud reconstruction and representation learning, are trained separately due to their structural differences. This separation inevitably leads to increased training costs and neglects the potential for mutual assistance between tasks. In this paper, a self-supervised method named PointUR-RL is introduced, which integrates point cloud reconstruction and representation learning. The method features two key components: a variable masked autoencoder (VMAE) and contrastive learning (CL). The VMAE is capable of processing input point cloud blocks with varying masking ratios, ensuring seamless adaptation to both tasks. Furthermore, CL is utilized to enhance the representation learning capabilities and improve the separability of the learned representations. Experimental results confirm the effectiveness of the method in training and its strong generalization ability for downstream tasks. Notably, high-accuracy classification and high-quality reconstruction have been achieved with the public datasets ModelNet and ShapeNet, with competitive results also obtained with the ScanObjectNN real-world dataset.

Keywords: self-supervised learning; point cloud reconstruction; representation learning; variable masked autoencoder; contrastive learning

check for updates

Citation: Li, K.; Zhu, Q.; Wang, H.; Wang, S.; Tian, H.; Zhou, P.; Cao, X. PointUR-RL: Unified Self-Supervised Learning Method Based on Variable Masked Autoencoder for Point Cloud Reconstruction and Representation Learning. *Remote Sens.* **2024**, *16*, 3045. https://doi.org/10.3390/rs16163045

Academic Editor: Mohammad Awrangjeb

Received: 8 July 2024
Revised: 13 August 2024
Accepted: 16 August 2024
Published: 19 August 2024

1. Introduction

In recent years, the advancement of deep learning has had a profound impact on various sectors, reshaping daily routines, professional practices, and the broader societal framework. Point clouds, as a key data format in 3D technology, provide accurate representations of objects in three-dimensional space and are essential for a range of practical 3D applications, such as autonomous driving and digital restoration. The introduction of PointNet [1] marked the beginning of a new era, enabling models to efficiently process point cloud data and paving the way for the integration of deep learning into this field. Initially, the field was dominated by supervised learning methods [2,3], which rely heavily on extensive sets of labeled data to train models based on the relationship between input data and their corresponding labels, such as classification tags. These methods are renowned for their high accuracy when ample labeled data is available. However, the acquisition of a vast amount of high-quality labeled data is often a challenging and costly task, which can limit the generalizability and scalability of models. To address this, self-supervised learning has emerged [4] to reduce reliance on labeled data and enhance model generalization. This approach has been introduced in response to the difficulties associated with obtaining large datasets of high-grade labeled data, thereby facilitating the development of more flexible and scalable models in the field of point cloud processing.

Self-supervised learning is conducted without reliance on labeled data, enabling models to identify and exploit the inherent structures and characteristics present within unlabeled datasets. It is primarily implemented through two methods: generative learning and contrastive learning. Generative learning encompasses models that learn the distribution of data, either to reconstruct the input data or to project potential future states, such as generative adversarial networks (GANs) [5,6] and autoencoders [7]. In contrast, contrastive learning (CL) [8] adopts a discriminative approach, using similarity metrics to bring closely related samples closer together while distancing those that are dissimilar. These strategies have been highly praised in the fields of computer vision and natural language processing. However, compared to the fields of natural language processing (NLP) and image processing, the progress of self-supervised learning in the three-dimensional domain has been relatively slow. This is mainly due to the unordered and spatially irregular nature of point cloud data, which sets it apart from regular image blocks and text in NLP. Such a structure poses significant challenges to the development of representation learning techniques. Inspired by the remarkable progress in natural language processing (NLP) and image analysis, a multitude of researchers have shifted their focus towards self-supervised learning in the domain of 3D point clouds. This redirection has spurred the development of a variety of methods specifically designed for self-supervised learning in point clouds, including approaches based on contrastive learning [9,10] as well as generative learning [11,12]. Among the current methods, techniques based on masked autoencoders have become prominent in 3D point cloud processing. The core principle of these approaches involves partially masked the input point cloud data, feeding the visible data into an autoencoder, and then reconstructing or recovering the hidden segments. In the realm of self-supervised learning, reconstruction and representation learning are key tasks that currently lack a unified method due to their structural differences. In reconstruction modeling, high-dimensional data is output based on low-dimensional inputs such as class labels, text embeddings, or random noise. However, an overemphasis on reconstruction fidelity could potentially hinder the model's ability to learn effective representations. Conversely, in representation learning, a high-dimensional image is input to produce a low-dimensional, compact embedding that is beneficial for downstream tasks. Yet, while superior representation learning capabilities are desirable, they may come at the expense of point cloud reconstruction quality.

The pursuit of unifying multiple tasks has garnered extensive interest among researchers [13]. A notable method, mage [14], has successfully integrated image generation and representation learning in image processing, achieving high-quality outcomes in both areas. Drawing inspiration from this accomplishment, in this work, the PointUR-RL has been proposed as a unified method that employs a variable masked autoencoder (VMAE) to achieve both point cloud reconstruction and representation learning. At the heart of PointUR-RL is the principle that reconstruction is equivalent to generating from fully masked point clouds, while representation learning corresponds to encoding from completely unmasked point clouds. Therefore, VMAE is utilized to unify these two tasks. PointUR-RL facilitates a smooth adaptation of reconstruction training and representation learning. Additionally, a contrastive learning (CL) module has been integrated to ensure the model excels in high-fidelity representation learning and produces distinctive learned embeddings.

The key contributions of this work are as follows:

- We introduce PointUR-RL, distinct from other self-supervised methods, which unifies point cloud reconstruction and representation learning through the use of a variable masked autoencoder. Furthermore, the incorporation of a contrastive learning module enhances the model's ability to learn representations, improving the separability of the learned features and ensuring the quality of these two tasks.
- Optimized for point cloud processing, PointUR-RL is capable of smoothly adapting to point cloud data with varying masked ratios during the pre-training period and naturally achieves class-unconditional point cloud reconstruction.

- The experimental results demonstrate that the pre-trained model of PointUR-RL is effective and possesses strong generalization capabilities for downstream tasks. It has achieved high accuracy in classification and high-quality point cloud reconstruction on public datasets such as ModelNet and ShapeNet. Additionally, it has shown good generalization performance on the ScanObjectNN real-world dataset.

2. Related Work

2.1. Self-Supervised Learning

Self-supervised learning (SSL) has been widely applied in the field of point cloud representation learning, as demonstrated by a multitude of methods [15,16]. Characterized by its capacity to identify and learn from the inherent structure and features of the data, SSL avoids reliance on external annotations or supervision. The core of SSL lies in the strategic design of preset tasks that enable the model's self-optimization process. Building on successful experience from the image domain, similar pretext tasks have been integrated into point cloud processing. For example, in the realm of contrastive learning, Chhipa et al. [17] introduced DepthContrast, a framework emphasizing the depth aspects of point clouds. Afham et al. [18] presented CrossPoint, a method for cross-modal contrastive learning designed to develop transferable 3D point cloud representations. Huang et al. [19] proposed STRL, an innovative approach fostering an unsupervised learning paradigm through the strategic interaction of online and target networks, thereby enhancing the learning experience. Liu et al. [20] have proposed an innovative approach to learning 3D representations, named Fac, which emphasizes the contrast between foreground and background features, thereby enhancing the model's ability to capture and distinguish fundamental characteristics within 3D data. In parallel, self-supervised learning through autoencoders has achieved significant progress. Wang et al. [21] introduced OcCo, a pioneering encoder-decoder architecture tailored to effectively reconstruct point cloud data affected by partial occlusions, thus advancing point cloud processing capabilities in the face of visibility challenges. Inspired by BERT [22], Yu et al. [23] proposed Point-BERT, which employs a masked point modeling (MPM) task for the pre-training of point cloud transformers. This method utilizes a discrete variational autoencoder (dVAE) to generate discrete point tokens rich in local information, with the pre-training objective being the restoration of the original tokens at masked locations, guided by a tokenizer. However, the requirement for dVAE pre-training in Point-BERT introduces additional complexity. To address this, Pang et al. [24] proposed Point-MAE, a masked auto-encoder method that simplifies the pre-training process by focusing on a masked task, subsequently enhancing the model's performance through downstream tasks. In recent research, Wu et al. [25] have advanced the state-of-the-art in point cloud processing by introducing a streamlined and efficient Transformer architecture known as Point Transformer V3. This improvement not only accelerates computational speed but also maintains exceptional performance across a variety of 3D tasks.

2.2. Autoencoder

Autoencoders (AEs) consist of two main components: an encoder and a decoder. The encoder plays a crucial role in capturing the essence of point data, as seen in networks such as PointNet [1] and EdgeConv [26]. It learns the intrinsic structure and patterns of the data through deep learning networks, mapping high-dimensional data into a low-dimensional latent space. The decoder then translates this compressed representation back into the original high-dimensional format, with the goal of accurately reconstructing the point cloud. The accuracy of the reconstruction is measured using metrics such as Chamfer distance (CD) and earth mover's distance (EMD), which act as loss functions to guide the optimization process.

Traditional autoencoders (AEs) are designed to obtain an abstract feature representation of input samples by minimizing the error between the input and reconstructed samples. However, this approach can result in AEs learning features that are merely an identity representation of the original input, which does not ensure the extraction of the sample's essential characteristics. These have led to the development of advanced algorithms such as sparse autoencoders (SAE), denoising autoencoders (DAE) [27], contractive autoencoders (CAE) [28], and variational autoencoders (VAE) [7]. Among these, denoising autoencoders (DAE) have become particularly popular for their ability to enhance model robustness through the incorporation of input noise. Masked autoencoders extend this principle by introducing noise via data masking. For instance, BERT [22] in the NLP domain uses masked language modeling to predict masked tokens based on the context. Similarly, in computer vision, approaches like MAE [29] and SimMIM [30] mask parts of images, challenging the autoencoder to infer and reconstruct the obscured regions. Inspired by these advancements in other fields, innovations in the 3D domain, such as Point-BERT [23] and Point-MAE [24], have adopted a masked point cloud modeling strategy. This strategy involves randomly masking sections of point cloud data and then using the autoencoder to regenerate these masked areas, which enhances the model's feature learning capabilities. Traditional autoencoders (AEs) are designed to obtain an abstract feature representation of input samples by minimizing the error between the input and reconstructed samples. However, this approach can result in AEs learning features that are merely an identity representation of the original input, which does not ensure the extraction of the sample's essential characteristics. These have led to the development of advanced algorithms such as sparse autoencoders (SAE), denoising autoencoders (DAE) [27], contractive autoencoders (CAE) [28], and variational autoencoders (VAE) [7]. Among these, denoising autoencoders (DAE) have become particularly popular for their ability to enhance model robustness through the incorporation of input noise. Masked autoencoders extend this principle by introducing noise via data masking. For instance, BERT [22] in the NLP domain uses masked language modeling to predict masked tokens based on the context. Similarly, in computer vision, approaches like MAE [29] and SimMIM [30] mask parts of images, challenging the autoencoder to infer and reconstruct the obscured regions. Inspired by these advancements in other fields, innovations in the 3D domain, such as Point-BERT [23] and Point-MAE [24], have adopted a masked point cloud modeling strategy. This strategy involves randomly masking sections of point cloud data and then using the autoencoder to regenerate these masked areas, which enhances the model's feature learning capabilities. In recent research, Li et al. [31] have introduced an innovative pretraining strategy for point models, harnessing the power of autoencoding and autoregressive techniques to enhance the representation learning of 3D shapes. This approach significantly boosts the generalization capability and performance on downstream tasks.

2.3. Transformer

Initially developed for text translation, Transformers [32] have quickly gained prominence in the natural language processing (NLP) domain, as demonstrated by their extensive adoption and significant impact [33,34]. Their innovative self-attention mechanism, which allows for direct mapping of dependencies between sequence positions, has garnered considerable interest in the field of computer vision [35]. The introduction of the Vision Transformer (ViT) [36] further solidified their position, leading to an exploration of Transformers in the 3D point cloud domain, a new application area with unique structural challenges that traditional models must address. Pioneering models such as the PCT [37], Pointformer [38], and Point Transformer [39] have established the groundwork for the application of Transformers to point cloud data, overcoming some of these initial challenges. On this foundation, the emergence of self-supervised learning models, including Point-BERT, Point-MAE, and Point-MA2E [40], has opened new frontiers in point cloud analysis. These models have broadened the scope of research, providing innovative methods for unsupervised feature learning directly from point cloud data. In subsequent research, Wu et al. [25] proposed the Point Transformer V3, which maintained exceptional performance across a variety of 3D tasks. Kolodiazhnyi et al. [41] introduced Oneformer3D, a unified approach to segment point clouds using a single Transformer model, simplifying the process and improving efficiency without sacrificing accuracy.

3. Methods

The overall structure is illustrated in Figure 1. The method for implementing PointUR-RL begins with segmenting the input point cloud into discrete blocks. A variable masking ratio, which ranges from 0.4 to 1, is then applied to obscure portions of these blocks randomly. The point cloud patches are further embedded through an embedding module. Following this, an encoder-decoder Transformer architecture is employed on the unmasked blocks to infer the concealed tokens. To enhance the discriminability of the learned representations, a simple yet effective contrastive loss, MoCo [42], is integrated into the encoder's output.

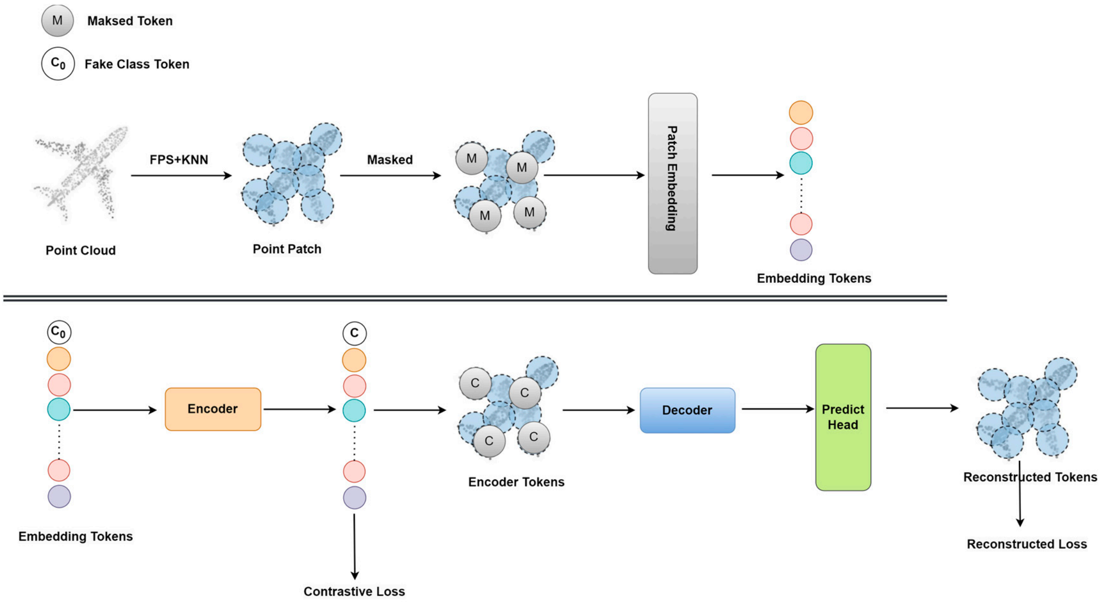

Figure 1. The structure of PointUR-RL.

3.1. Point Patch Generation

In the initial phase of PointUR-RL design, the objective was to achieve a comprehensive encoding of the entire point cloud dataset, with each point treated as an individual token for the Transformer encoder. However, this strategy faced a significant challenge: the sheer volume of data inherent in point cloud representations presented a considerable computational burden. Additionally, it became clear that point clouds possess a unique structure that sets them apart from images and textual data, which complicates their segmentation into patches or lexical units. To address this, inspiration was drawn from the approaches used in PointNet++, resulting in the decision to segment the input point cloud into irregular clusters. This was accomplished by utilizing the farthest point sampling (FPS) and k-nearest neighbors (KNN) algorithms, which were adapted and refined to meet the specific requirements of the framework as detailed below:

$$Center = FPS(\mathcal{P}), \ \ Center \in \mathbb{R}^{G\times 3} \tag{1}$$

$$idx = KNN(\mathcal{P}, Center), \ \ idx \in \mathbb{R}^{G\times M} \tag{2}$$

where $\mathcal{P} \in \mathbb{R}^{N\times 3}$ is the input point cloud; Center is the center point N of the point cloud obtained by fps; and idx refers to the collection of neighborhood points of each central point after applying the knn algorithm. In addition, after idx is obtained, the coordinates of the domain points are obtained and normalized. This allows for better integration. Finally, the set of domain points for each center point is obtained as $neighbor \in \mathbb{R}^{\mathrm{G}\times\mathrm{M}\times 3}$.

3.2. Masking Strategy

A strategy involving a variable masking ratio has been devised to bridge the gap between point cloud reconstruction modeling and representation learning. This strategy entails the random sampling of a masking ratio, denoted as m_r, from a truncated Gaussian distribution. This distribution is centered around 0.55, with a lower bound of 0.4 and an upper limit of 1, ensuring that the masking ratio stays within a practical range. Given a set of input point cloud patches, denoted by G, the masking ratio m_r is applied to ascertain the number of patches to be masked. These masked patches are subsequently replaced with a learnable mask token, represented as $[M]$. Inspired by MAE [29], the training process is focused exclusively on the unmasked patches. By focusing on the unmasked patches, a significant portion of the mask tokens is effectively discarded, resulting in a notable reduction in pre-training time and memory usage. This streamlined process not only enhances computational efficiency but also strengthens the capabilities of generative modeling and representation learning. The truncated Gaussian distribution, which dictates the sampling of the masking ratio, is illustrated in Figure 2.

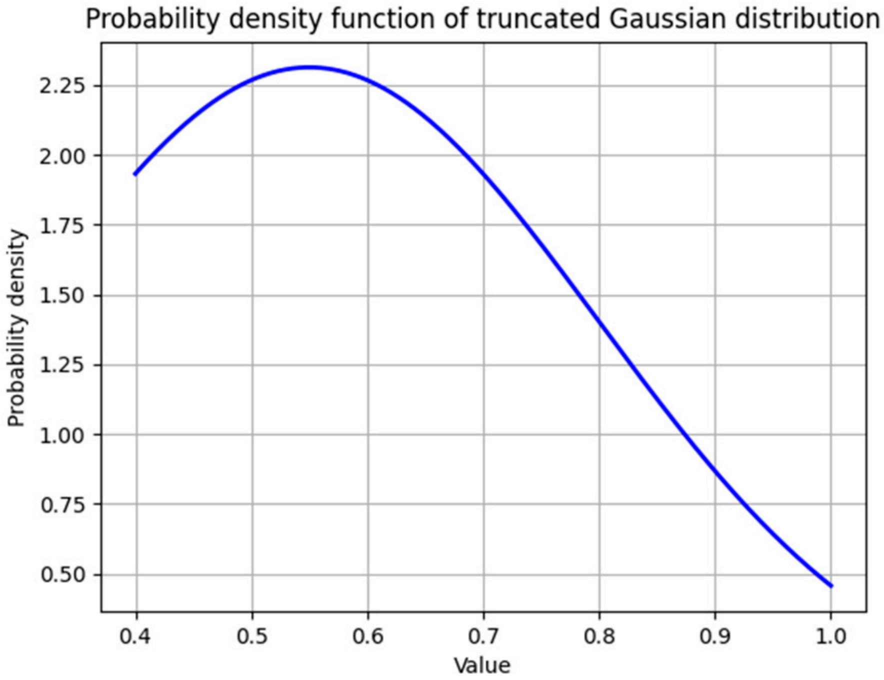

Figure 2. Truncated Gaussian distribution map.

3.3. Embedding Module

Our process is initiated by employing a lightweight PointNet for embedding, which converts each block of the point cloud into a series of embeddings. This conversion is achieved by leveraging a Multi-Layer Perceptron (MLP) coupled with max pooling, as depicted in Figure 3. The resulting embeddings are then divided into two distinct sets: the visible tokens, T_{vis} and the masked tokens, T_m. For T_m, it is replaced with a shared weighted learnable mask token. Both T_{vis} and T_m can be represented as

$$T_{vis} = (1 - m_r) \times T \; , \; T \; \in \mathbb{R}^{G \times C} \tag{3}$$

$$T_m = m_r \times T \; , \; T \; \in \mathbb{R}^{G \times C} \tag{4}$$

where T symbolizes the embeddings in $\mathbb{R}^{G \times C}$; G denotes the number of point patches derived from the FPS algorithm; and C represents the dimensionality of the embedding space.

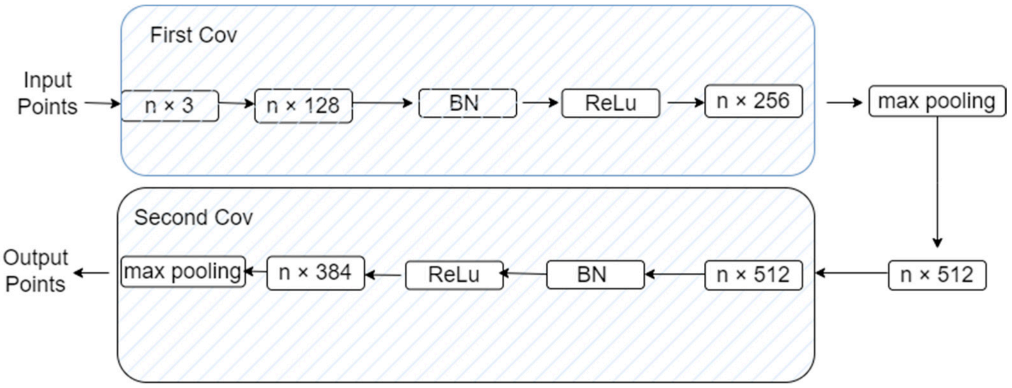

Figure 3. Embedding module.

As we proceed, it is recognized that point cloud blocks, similar to image blocks, necessitate normalization and are processed accordingly to ensure consistency. Following established methods, the embeddings are enhanced with positional information. This enhancement is realized by integrating a learnable multi-layer perceptron (MLP) that projects the central coordinates of each token into the embedding space, thus endowing the model with spatial awareness. It is noteworthy that the unique structures of the encoder and decoder within our architecture necessitate distinct positional embeddings. Initially, the focus is on the unmasked embeddings, which are the only ones that require positional embeddings at the encoding stage. However, post the encoding process, both unmasked and masked embeddings must be considered, with positional embeddings applied to ensure that all tokens are correctly positioned within the model's latent space.

3.4. Encoder-Decoder Design

Our encoder-decoder module, as shown in Figure 4, is characterized by an asymmetric architecture that incorporates a conventional Transformer model. Constructed with twelve Transformer blocks, the encoder is complemented by a decoder composed of four blocks, establishing an asymmetry that facilitates an efficient decoding process and enhances the training efficiency.

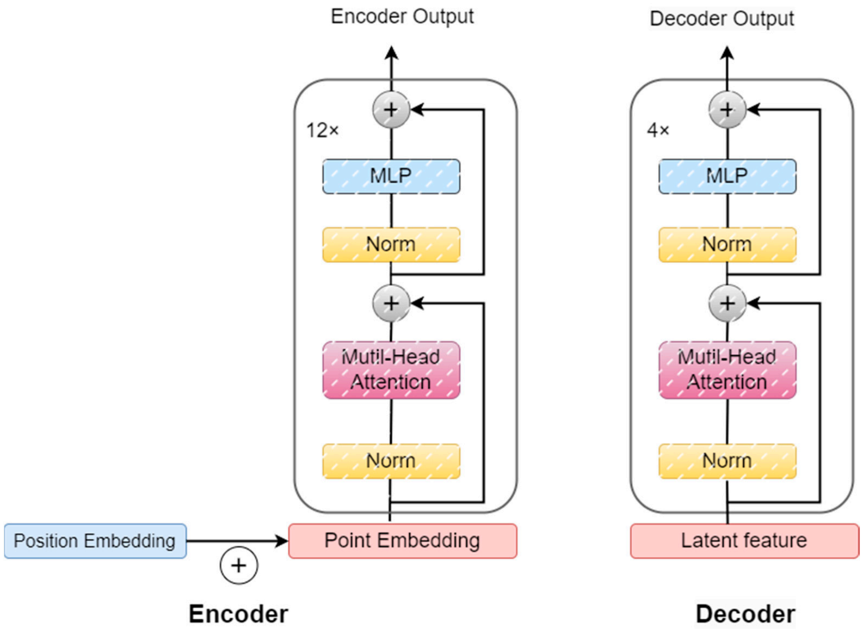

Figure 4. The structure of encoder and decoder.

After the point cloud blocks have been masked and embedded to obtain T_{vis}, a learnable "fake" class token $[C_0]$ is concatenated to the input sequence. Positional encoding is then added to the input sequence, and the resulting cascaded sequence T_{input} is fed into the standard Transformer encoder-decoder structure, as illustrated in Equation (5). Specifically, the Transformer encoder takes the unmasked embeddings as input and encodes them into a latent feature space. Prior to decoding, the encoder's output is populated to the full input length using the class token features [C] learned by the encoder. This is because the class token position is capable of summarizing the global features of the input point cloud. Consequently, the point cloud-specific [C] is utilized to populate the encoder output, rather than a learnable mask token shared across different point clouds, as depicted in Figure 1. This design choice is shown to enhance generative and representation learning performance compared to the use of a learnable mask token. Subsequently, the decoder utilizes the populated features to reconstruct the masked original tokens.

$$T_{input} = Pos_{concat(C_0,vis)} + Concat(C_0, T_{vis}) \tag{5}$$

3.5. Reconstruction

The decoding phase concludes with the reconstruction stage, where the goal is to reinstate the coordinates of points within each obscured point cloud patch. To achieve this, a straightforward prediction module is established for the reconstruction of the veiled patches. The feature vector T_{output}, emanating from the decoder, is mapped onto a vector space of dimensions equivalent to those of the point cloud patches. After a reshaping operation, the anticipated masked point cloud patches are derived. Subsequently, a comparison is made between the anticipated patches P_{pre} and the actual data P_{gt}. The reconstruction loss is ascertained employing the $\mathcal{L}_2$ Chamfer distance, as delineated by the subsequent formula:

$$\mathcal{L}_{ret} = \frac{1}{|P_{pre}|} \sum_{a \in P_{pre}} \min_{b \in P_{gt}} \|a - b\|_2^2 + \frac{1}{|P_{gt}|} \sum_{b \in P_{gt}} \min_{a \in P_{pre}} \|a - b\|_2^2 \tag{6}$$

3.6. Contrastive Co-Training

Inspired by advances in image processing, a contrastive loss has been integrated into our model to enhance the efficacy of representation learning within masked reconstruction tasks. This approach is designed to distill robust and discriminative features that are critical for subsequent applications. The MoCo [42] method, a leading technique in contrastive learning, is harnessed to cultivate superior representations. The implementation involves a series of modifications to the encoder's output: global average pooling is applied to distill the features into a concise representation, which is then followed by two multi-layer perceptron (MLP) layers that refine and project the features into a space optimized for contrastive learning. Finally, the information noise contrastive estimation (InfoNCE) loss is applied to the MLP output, ensuring that the representations are learned with high fidelity. The contrastive learning flowchart is shown in Figure 5.

$$\mathcal{L}_{contrastive} = -log \frac{exp(qk^+/\tau)}{\sum_{i=0}^{K} exp(qk_i/\tau)} \tag{7}$$

In Equation (7), the hyperparameter τ plays a pivotal role in dictating the shape of the distribution, which is crucial for balancing the focus between positive and negative samples during the learning process. The larger the value of τ, the smaller the values in the distribution become after exponentiation, leading to a smoother distribution. This is equivalent to the contrastive loss focusing equally on all negative samples, which can result in the learning model not being heavily influenced by any single sample. Conversely, the smaller the τ, the more concentrated the distribution becomes, causing the model to focus primarily on those particularly challenging samples. In fact, those negative samples are likely to be potential positive samples. k^+ denotes a positive sample pair and k_i

represents a negative sample. The numerator is q and represents the positive samples, and the denominator is actually the sum of k negative samples as it is from 0 to k, and so it represents $k + 1$ samples, which is all the keys in the dictionary.

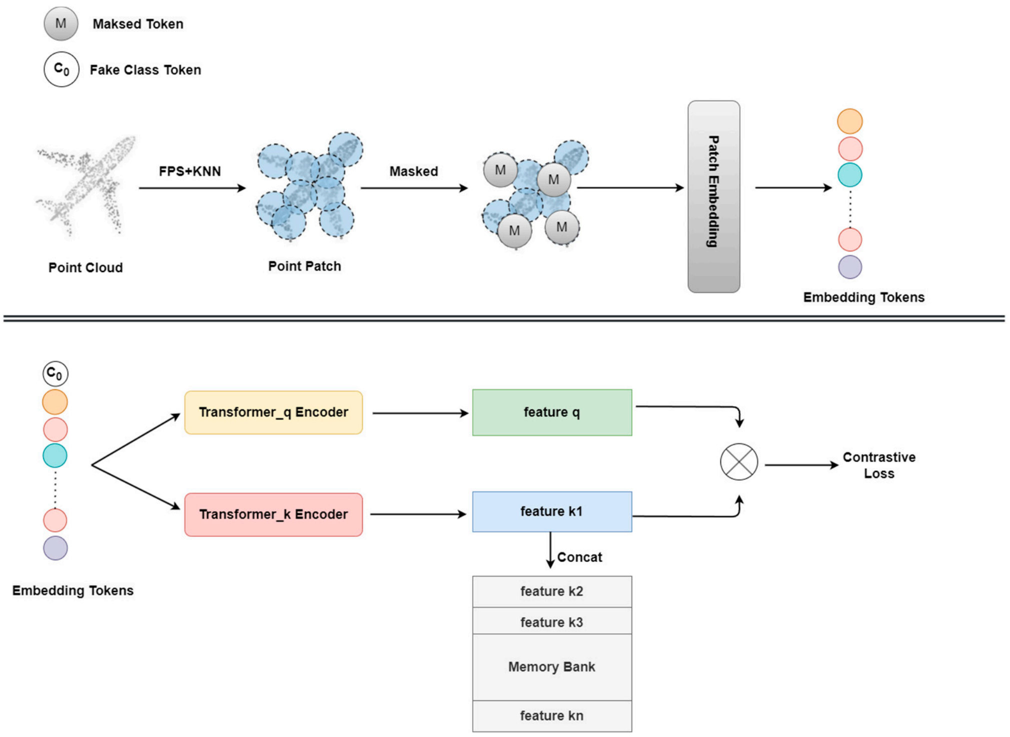

Figure 5. The flowchart of the contrastive learning.

Thus, the final loss function is:

$$\mathcal{L} = \mathcal{L}_{ret} + \lambda \cdot \mathcal{L}_{contrastive} \tag{8}$$

where $\lambda = 0.1$ balances the scale of the two losses. We do not use the extensive augmentations typically used in contrastive learning, such as point cloud data jitter, point cloud data rotation, and point cloud data rotation perturbation. This is because the reconstruction loss acts as a regularizer, preventing the encoder from learning the shortcut solution. Even without contrast loss, PointUR-RL results in better generation tasks and representation learning, and the performance of representation learning can be further improved with contrast loss.

4. Experiments and Results

4.1. Pretrain Setting

For the pre-training phase of our model, the ShapeNet dataset was selected. It is a comprehensive repository that comprises approximately 51,300 high-quality 3D models, representing 55 diverse object categories, and offering a rich variety of shapes and structures. Although ShapeNet is meticulously organized into a training set and a validation set, for our pre-training endeavors, the focus was exclusively on the training subset. The selection of ShapeNet was due to its provision of more diversified data and richer categories, ensuring that our model achieved good learning performance during the pre-training process.

Regarding the details of our training, all experiments were conducted on a Titan RTX graphics card. A total of 300 training papers were utilized, with a batch size set to 64. AdamW served as the optimizer, employing a learning rate of 0.001 and a weight decay of 0.05. To regulate the learning rate across training epochs, a CosLR scheduler was implemented. This scheduler modified the learning rate in accordance with a cosine function, potentially resulting in improved convergence properties. The initial 10 training epochs were designated for warm-up, a strategy that incrementally raised the learning rate from a minimal value to the initial rate, which led to more rapid and stable convergence. For data input, 1024 points were sampled from the dataset to constitute each point cloud, representing the fundamental unit of data for our model. In the processing of point clouds, the farthest point sampling (FPS) and k-nearest neighbors (KNN) algorithms were employed with parameters set to $numgroup = 64$ and $group\_size = 32$. These parameters indicated that 64 points were extracted from the 1024 as group centers, and around these centers, 32 points were searched based on their coordinates. The architecture of our model was founded on the Transformer framework, which is asymmetrically configured for the encoder and decoder to accommodate their distinct functions. The encoder was composed of 12 Transformer layers, each with six Attention heads, enabling it to capture intricate patterns within the data. Correspondingly, the decoder was made up of four layers, also equipped with six Attention heads each, which was essential for the precise reconstruction of the masked tokens.

4.2. Downstream Tasks

4.2.1. Object Classification on ModelNet40

ModelNet40, a dataset renowned for 3D object recognition and shape analysis, was created by researchers from the Department of Computer Science at Princeton University. It encompasses 40 categories of objects, including tables, chairs, and cars, and comprises a total of 12,311 CAD-generated mesh models. Divided into a training set with 9843 models and a test set with 2468 models, the dataset's high-quality CAD models render it suitable for a variety of 3D vision tasks.

For shape classification tasks, a fine-tuning strategy was implemented on the ModelNet dataset. In this regimen, the input point clouds were uniformly set to 1024 per model. The training extended over 300 epochs, with batches consisting of 64 samples each. AdamW was chosen as the optimizer, with a learning rate of 0.001 and a weight decay factor of 0.05. A cosine annealing learning rate scheduler, CosLR, was employed to manage the training process, and the architecture of the Transformer encoder was consistent with that used in the pre-training phase. To ensure experimental integrity, a uniform number of 1024 input points was maintained across all configurations. The results of our experiments are presented in Table 1.

Table 1. Object classification on ModelNet40 dataset.

Training Category	**Methods**	**Acc.**
Supervised methods	PointNet	89.2%
	PointNet++	90.7%
	PointCNN [43]	92.5%
	DGCNN	92.9%
	[ST] Transformer	91.4%
	[T] PCT [37]	93.2%
	[T] Point Transformer [39]	93.7%
Self-supervised methods	OcCo	93.0%
	STRL [19]	93.1%
	[ST] Transformer + OcCo	92.1%
	[ST] Point-BERT	93.2%
	[ST]Point-BERT (rec.)	93.1%
	[ST]Point-MAE	93.8%
	[ST]Point-MAE (rec.)	93.11%
	Ours	93.31%

Our method has been compared against a spectrum of classic supervised learning and self-supervised learning methods. Here, [T] signifies the improved Transformer-based methods, while [ST] denotes those predicated on the Transformer architecture. To ensure a fair comparison, seminal works such as Point-BERT and Point-MAE, foundational to the masked autoencoder concept, were reproduced using their official code and pre-trained models. The term "rec." refers to these reproduced results.

An analysis of the experimental data presented in Table 1 indicates that PointUR-RL surpassed the majority of supervised learning techniques. Specifically, our method realized accuracy improvements of 4.11%, 2.61%, and 0.41% over PointNet, PointNet++, and DGCNN, respectively, when compared with established supervised learning approaches. When contrasted with Transformer-based supervised learning methods, PointUR-RL's precision was heightened by 1.91% over Transformer and by 0.11% over PCT, although it slightly trailed Point Transformer by 0.39% in terms of precision. Furthermore, within the self-supervised learning domain, our method exhibited dominance over the majority of existing methods. It displayed a 0.21% precision enhancement over STRL, which relied on contrastive learning, and a 0.31% increase when juxtaposed with OcCo. In direct comparison with Point-BERT and Point-MAE, PointUR-RL surpassed the performance of Point-BERT, Point-BERT (rec.), and Point-MAE (rec.), yet it narrowly fell behind Point-MAE. We have thoroughly examined the reasons for the slight performance gap behind Point-MAE. Our approach aimed to unify two key tasks: point cloud reconstruction and representation learning. To balance these tasks, we have, to some extent, inevitably compromised the representation learning capability of our method, as evidenced by the high occlusion ratio of over 80% during the training process. Additionally, to ensure the fairness of the experiment, we utilized the official code and models provided by Point-MAE in our identical experimental setup. The results of the experiment demonstrate that our method outperformed Point-MAE (reconstructed), indicating that our approach has advantages over Point-MAE.

The experimental results clearly demonstrate the high effectiveness of our pre-trained model, achieving competitive results in the field of shape classification. To highlight the strength of our methodology, t-SNE has been utilized to visualize the feature distributions from our experiments, with the results depicted in Figure 6.

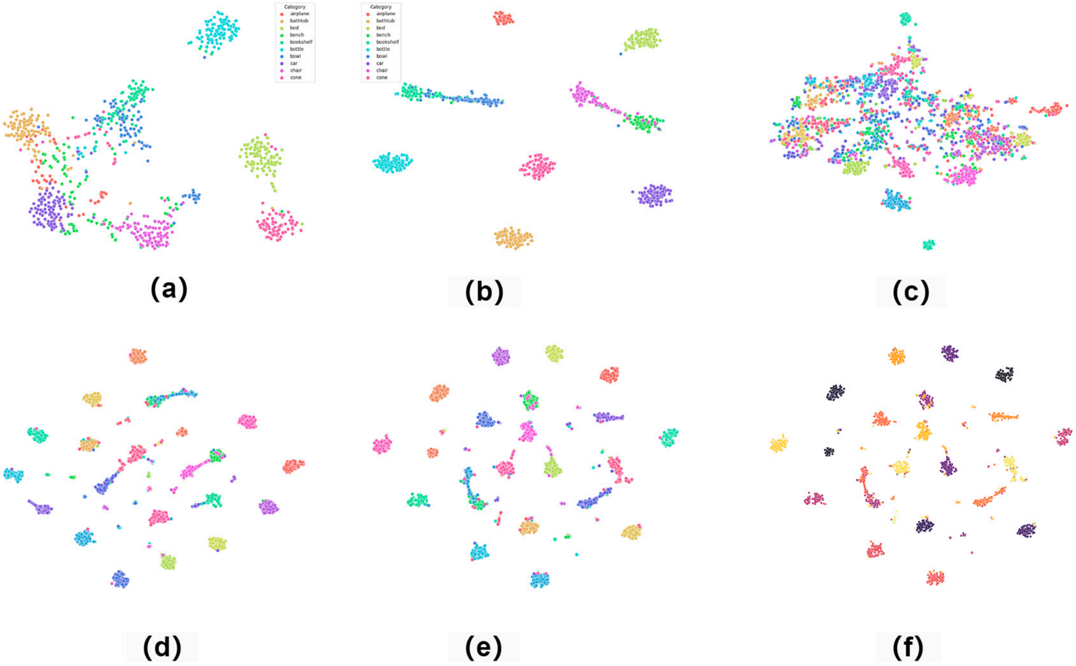

Figure 6. Visualization of feature distributions. The t-SNE visualization is displayed for feature vectors learned by our model in various stages: (**a**) after pre-training on ModelNet10, (**b**) after fine-tuning on ModelNet10, (**c**) after pre-training on ModelNet40, (**d**) after fine-tuning on ModelNet40, (**e**) after fine-tuning on ModelNet40 by Point-MAE, and (**f**) after fine-tuning on ModelNet40 by Point-MAE by Point-BERT.

4.2.2. Object Classification on Real-World Dataset

Our pre-training model, grounded solely in the object models of the ShapeNet dataset without scene features, was subjected to a robustness and generalization capability test on the challenging real-world dataset, ScanObjectNN. This dataset, consisting of approximately 15,000 realistically scanned objects across 15 classes with 2902 unique instances, introduces significant challenges due to the presence of occlusions and noise, which are common in point cloud analysis techniques. The performance of our model on ScanObjectNN thus serves as a testament to its generalization prowess. In accordance with the protocols established by prior research, experiments were conducted across three principal variants: OBJ-BG, OBJ-ONLY, and PB-T50-RS. The results of these evaluations are meticulously detailed in Table 2.

Table 2. Object classification on ScanObjectNN dataset.

Methods	OBJ-BG	OBJ-ONLY	RB-T50-RS
PointNet	73.3	79.2	68
SpiderCNN	77.1	79.5	73.7
PointNet++	82.3	84.3	77.9
DGCNN	82.8	86.2	78.1
BGA-DGCNN	-	-	79.7
GBNet [44]	-	-	80.5
PRANet [45]	-	-	81.0
Transformer	79.86	80.55	77.24
Transformer + OcCo	84.85	85.54	78.79
Point-BERT	87.43	88.12	83.07
Point-BERT (rec.)	87.43	86.91	83.10
Point-MAE	90.02	88.29	85.18
Point-MAE (rec.)	88.98	88.29	84.31
Ours	89.67	88.81	84.35

In the most formidable variant, “PB-T50-RS”, our model achieved a classification accuracy of 84.35%, surpassing the current state-of-the-art methods, Point-BERT, Point-BERT (rec.), and Point-MAE (rec.), by 1.28%, 1.25%, and 0.05%, respectively. In the “OBJ-ONLY” scenario, the highest precision rate attained was 88.81%. When evaluated on the “OBJ-BG” dataset, our model’s performance reached 89.67%, which was only marginally lower than Point-MAE’s 90.02%. These experimental outcomes demonstrate that PointURRL’s performance on ScanObjectNN was not only on par with Point-MAE, but also slightly exceeded that of Point-MAE (rec.). This underscores the exceptional representation learning capabilities of our pre-trained model, as well as its commendable generalization across diverse domains.

4.2.3. Object Reconstruction on ShapeNet55

Our model, integrating representation learning with point cloud reconstruction, placed significant emphasis on reconstruction capability as a pivotal measure of its overall performance. As a result, the reconstruction proficiency of our model was evaluated on the ShapeNet55 dataset, positioning it against leading contemporary methods. The L2 Chamfer distance (L2CD) was adopted as the evaluative metric for this assessment, with the findings encapsulated in Table 3.

Upon reviewing Table 3, it is revealed that our method exhibited commendable reconstruction efficacy. Specifically, in categories such as Airplane, Chair, and Table, our reconstruction outcomes were on par with the state-of-the-art in reconstruction performance. However, a dip in performance was noted for categories like Car and Sofa, which was attributed to the inherent challenge of our reconstruction process, which must contend with an average missing data rate of up to 70%. Furthermore, to present a more vivid depiction of our reconstruction capabilities, the reconstruction results for the ShapeNet55 dataset have been visualized, with the visual representation detailed in Figure 7.

Table 3. Quantitative comparison on ShapeNet-55.

Methods	Table	Chair	Plane	Car	Sofa
FoldingNet	2.53	2.81	1.43	1.98	2.48
PCN	2.13	2.29	1.02	1.85	2.06
TopNet	2.21	2.53	1.14	2.18	2.36
PFNet	3.95	4.24	1.81	2.53	3.34
GRNet	2.53	2.81	1.43	1.98	2.48
Ours	2.35	2.48	1.09	3.66	3.56

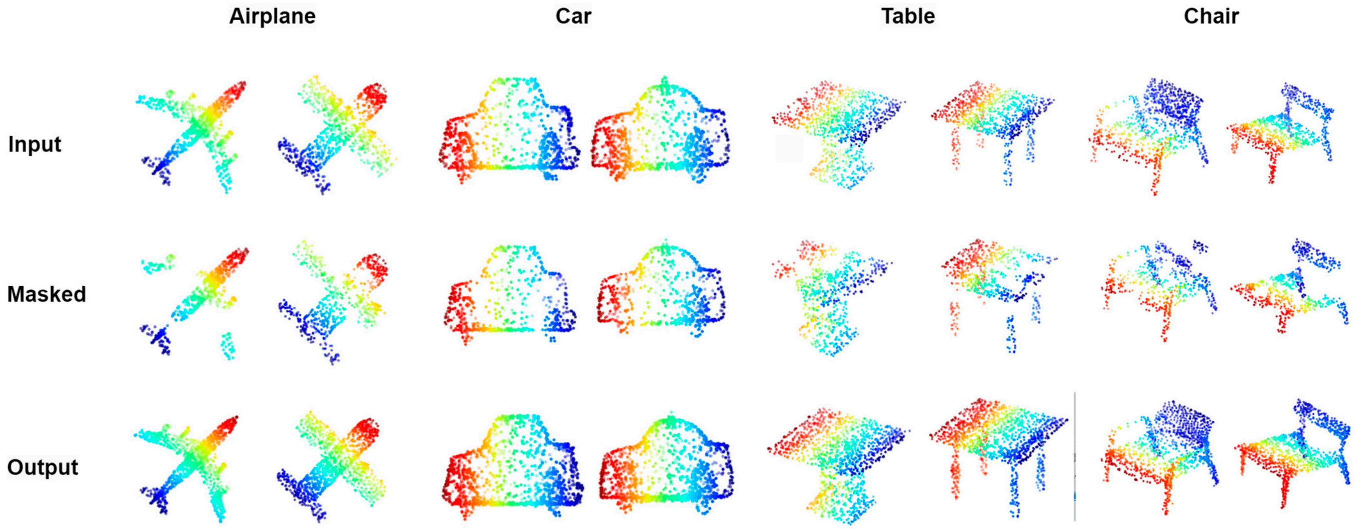

Figure 7. Visualization of point cloud reconstruction.

4.2.4. Part Segmentation

Partial segmentation of point cloud objects presents a formidable challenge for evaluating the proficiency of pre-trained models, which aim to assign category labels to individual points. To assess the representation learning ability of PointUR-RL, the ShapeNetPart dataset was utilized. This dataset consists of 16,881 samples spread over 16 categories, with 13,998 allocated for training and 2874 reserved for testing. In accordance with precedent studies, our method samples 2048 points per object for input, segmented into 128 patch blocks. The outcomes of our experiments are articulated in Table 4.

Table 4. Part segmentation on ShapeNetPart dataset.

Methods	*mIoU$_i$*	Airplane	Bag	Cap	Car	Chair	Earphone	Guitar	Knife
		Lamp	Laptop	Motor	Mug	Pistol	Rocket	Skateboard	Table
PointNet	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9
		80.8	95.3	65.2	93	81.2	57.9	72.8	80.6
PointNet++	85.1	82.4	79.0	86.7	77.3	90.8	71.8	91.0	85.9
		83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
DGCNN	85.2	84.0	83.4	86.7	77.8	90.6	74.7	91.2	87.5
		82.8	95.7	66.3	94.9	81.1	63.5	74.5	82.6
Transformer	85.1	82.9	**85.4**	87.7	78.8	90.5	80.8	91.1	87.7
		85.3	95.6	73.9	94.9	83.5	61.2	74.9	80.6
Point-BERT	85.6	84.3	84.8	88.0	79.8	91.0	**81.7**	91.6	87.9
		85.2	95.6	75.6	94.7	84.3	63.4	76.3	81.5
Point-MAE	**86.1**	84.3	85.0	88.3	80.5	91.3	78.5	92.1	87.7
		86.1	96.1	75.2	94.6	84.7	**63.5**	**77.1**	82.4
Ours	**85.66**	**84.7**	85.2	87.9	**80.9**	91.2	79.7	**92.1**	**87.9**
		85.4	96.0	**76.2**	**94.9**	**84.9**	62.2	75.9	80.7

In can be seen in Table 4 that the PointUR-RL achieved an mIoU score of 85.66%, an accomplishment that positions it ahead of Point-BERT and just marginally below Point-MAE. It is noteworthy that our method demonstrated superior performance in six distinct categories: airplane, motor, car, mug, pistol, guitar, and knife, where it outperformed alternative methodologies. Certain categories may include more complex or diverse shapes, which posed a challenge to the accurate segmentation of our model. For instance, categories like "chair" and "laptop" have more intricate structures, and our model learns complex local features when dealing with these categories. Additionally, due to the high occlusion ratio in our pre-training process, our model's ability to capture local information was insufficient, preventing effective representation learning of local details. This also led to suboptimal segmentation performance for certain categories.

4.2.5. Semantic Segmentation

The task of 3D point cloud scene semantic segmentation poses a significant challenge in evaluating the proficiency of pre-trained models. To verify the generalization capability of our approach, we validated our model using the S3DIS dataset. The S3DIS dataset, released by Stanford University, is a large-scale indoor 3D reconstruction dataset that includes indoor areas of six different buildings, each with detailed 3D point clouds and corresponding semantic labels. The point clouds in the dataset are divided into 13 main categories, such as ceilings, floors, walls, windows, doors, etc., providing rich annotated information for tasks such as indoor scene understanding, 3D object recognition, and semantic segmentation.

We tested our model on Area 5 while training on other areas. To ensure a fair comparison in the experiments, we tested other models under the same experimental setup. The experimental results are shown in Table 5. Compared with supervised models, our model's mIoU performance was significantly improved compared to PointNet, PointNet++, and PointCNN. When compared with other state-of-the-art self-supervised models, our mIoU performance slightly exceeded theirs, achieving good results. The experiments have proven that our pre-trained model is effective and has a wide generalization capability. Our method is capable of extracting contextual and semantic information, achieving fine-grained segmentation results.

Table 5. Semantic segmentation results on S3DIS Area 5. We report the mean IoU (%) and mean Accuracy (%).

Training Category	Methods	mIoU	mAcc
Supervised methods	PointNet	41.4	49.0
	PointNet++	53.5	-
	PointCNN	57.4	63.9
	KPConv	67.1	72.8
	SegGCN	63.6	70.4
	MKConv	67.7	75.1
Self-supervised methods	Point-BERT	68.9	76.1
	MaskPoint	68.6	74.2
	Point-MAE	68.4	76.2
	Ours	69.0	76.2

4.3. Ablation Studies

PointUR-RL is anchored by two pivotal components: the variable masked autoencoder and contrastive learning. To meticulously assess the impact of each, ablation experiments were conducted, meticulously designed to be executed within an identical experimental setting. This controlled approach ensured that the influence of these core elements could be accurately measured, allowing for a thorough understanding of their individual contributions to the model's performance.

4.3.1. Ablation Studies on Variable Masked Autoencoders

The variable masking autoencoder is an integral component of PointUR-RL, playing a pivotal role in representation learning. Our investigations indicate that the quality of the learned representation is significantly influenced by the distribution from which the masking ratio is sampled. To evaluate the impact of this variability, PointUR-RL's performance was benchmarked on the ShapeNet-55 dataset for average reconstruction loss and on the ModelNet40 for classification accuracy. The parameters of the truncated Gaussian distribution were manipulated, with the mode represented by μ and the standard deviation by σ. Two sets of control experiments were conducted: initially, the mode μ was fixed while setting $\sigma = 0$, thereby establishing a constant mask ratio, and the outcomes were assessed. Subsequently, σ was varied to identify the optimal masking ratio that yielded the best results. The findings from these meticulous experiments are systematically presented in Table 6, offering insights into the variable masking autoencoder's contribution to the overall system efficacy.

Table 6. Top-1 object classification on ModelNet40 and reconstruction loss on ShapeNet55 with different masking ratio distribution.

	μ=0.4	**μ=0.55**	**μ=0.6**	**μ=0.8**	**μ=0.55**		
	σ=0				**σ=0.1**	**σ=0.25**	**σ=0.5**
Object classification on ModelNet40	92.99%	93.23%	93.11%	93.19%	92.94%	**93.31%**	92.94%
Average Reconstruction on ShapeNet55	2.87	2.75	2.64	2.80	2.66	2.57	2.8

The experimental results indicate that for the ablation of μ, when μ was set to 0.55, the classification accuracy achieved was 93.23%, and the reconstruction effect was 2.75. Utilizing μ = 0.55 to ablate σ, it was discovered that a $\sigma = 0.25$ could yield an accuracy of 93.31% and an average reconstruction loss of 2.57.

4.3.2. Ablation Experiments with Contrastive Learning

Contrastive learning stands as a cornerstone of PointUR-RL, and we were motivated to investigate its impact on the results when integrated into the system. The hypothesis explored was that the inclusion of contrastive learning could significantly enhance the quality of outcomes. Additionally, our previous approach to the decoder involved using a per-point cloud specific token [C] to complement the encoder's output, diverging from the conventional use of shared, learnable mask tokens across various point clouds. This divergence prompted us to conduct experiments to evaluate the potential influence of this modification on the results. Given the ablation studies already performed on the variable occlusion autoencoder, we opted not to repeat the analysis here. Model A was defined as the scenario where neither contrastive learning nor the per-point cloud token [C] was utilized. Model B represented a setup devoid of a contrastive learning component. Model C proceeded without employing the per-point cloud token [C]. Model D, on the other hand, encapsulated our enhanced method, which incorporated both contrastive learning and the use of per-point cloud [C] in the decoder. The comparative effects of these models on the experiments are meticulously detailed in Table 7.

Table 7. Comparing the impact of different models on the quality of representation learning.

Model Configuration	**Variable Ratio**	**Inclusion of Contrastive Learning**	**Use of Class Fack Token [C]**	**Object Classification on ModelNet40**
Model A	√			92.58%
Model B	√		√	93.19%
Model C	√	√		92.74%
Model D	√	√	√	**93.31%**

As observed in Table 7, Model B's performance was lower than that of Model D by 0.19% when the contrastive learning component was not utilized. Similarly, Model C underperformed Model D by 0.57% when a specific per-point cloud token [C] was not used to complement the encoder's output. This indicates that the incorporation of both components enhanced the quality of our representation learning.

5. Conclusions and Discussion

In this paper, a self-supervised method to point cloud processing was introduced, unifying point cloud reconstruction and representation learning through a variable masking autoencoder. The encoder of our method is capable of adaptively processing the input point cloud, with the decoder subsequently reconstructing the obscured blocks. This innovative design seamlessly accommodates the dual objectives of point cloud reconstruction and feature representation, enhancing the model's versatility. Furthermore, contrastive learning is incorporated to bolster the model's ability to learn robust representations, thereby enhancing the separability of the learned features. The empirical results are compelling: our model demonstrates efficacy in the pre-training phase and exhibits strong generalization in subsequent reconstruction and representation tasks. It not only delivers high-accuracy classifications and superior reconstruction quality on public datasets such as ModelNet and ShapeNet, but also shows competitive performance on the real-world ScanObjectNN dataset.

However, we acknowledge that the irregular nature of point clouds presents certain challenges, potentially leading to a performance gap in reconstruction when compared to state-of-the-art techniques, a shortcoming we aim to address. Looking ahead, we are committed to enhancing our method's capacity for local feature learning. Our strategy involves transforming point clouds into semantic information for pre-training, which we believe will significantly elevate the performance of both reconstruction and representation learning tasks.

Author Contributions: Conceptualization, K.L. and Q.Z.; methodology, K.L. and Q.Z.; software, H.W.; validation, H.T. and S.W.; investigation, K.L.; resources, P.Z.; data curation, K.L.; writing—original draft preparation, K.L.; writing—review and editing, K.L.; supervision, X.C.; project administration, X.C.; funding acquisition, K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Key Research and Development Program of Shaanxi Province (2019GY-215, 2021ZDLSF06-04, 2024SF-YBXM-681), in part by the National Natural Science Foundation of China (61701403, 61806164).

Data Availability Statement: The underlying data that support the findings of this study are available in the public domain via the Internet. The datasets used for the research, including ModelNet, ShapeNet, and ScanObjectNN, are freely accessible online and can be obtained by interested researchers for the purpose of reproducing the results presented in this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
2. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep learning for 3d point clouds: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 4338–4364. [CrossRef] [PubMed]
3. Zhang, R.; Tan, J.; Cao, Z.; Xu, L.; Liu, Y.; Si, L.; Sun, F. Part-Aware Correlation Networks for Few-shot Learning. *IEEE Trans. Multimed.* 2024; *Early Access*.
4. Liu, X.; Zhang, F.; Hou, Z.; Mian, L.; Wang, Z.; Zhang, J.; Tang, J. Self-supervised learning: Generative or contrastive. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 857–876. [CrossRef]
5. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative adversarial networks: An overview. *IEEE Signal Process. Mag.* **2018**, *35*, 53–65. [CrossRef]
6. Zhang, R.; Li, L.; Zhang, Q.; Zhang, J.; Xu, L.; Zhang, B.; Wang, B. Differential feature awareness network within antagonistic learning for infrared-visible object detection. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *34*, 6735–6748. [CrossRef]

7. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
8. Ye, M.; Zhang, X.; Yuen, P.C.; Chang, S.-F. Unsupervised embedding learning via invariant and spreading instance feature. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6210–6219.
9. Achlioptas, P.; Diamanti, O.; Mitliagkas, I.; Guibas, L. Representation learning and adversarial generation of 3d point clouds. *arXiv* **2017**, arXiv:1707.02392.
10. Poursaeed, O.; Jiang, T.; Qiao, H.; Xu, N.; Kim, V.G. Self-supervised learning of point clouds via orientation estimation. In Proceedings of the 2020 International Conference on 3D Vision (3DV), Fukuoka, Japan, 25–28 November 2020; pp. 1018–1028.
11. Li, R.; Li, X.; Fu, C.-W.; Cohen-Or, D.; Heng, P.-A. Pu-gan: A point cloud upsampling adversarial network. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7203–7212.
12. Sarmad, M.; Lee, H.J.; Kim, Y.M. Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5898–5907.
13. Yang, G.; Huang, X.; Hao, Z.; Liu, M.-Y.; Belongie, S.; Hariharan, B. Pointflow: 3d point cloud generation with continuous normalizing flows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4541–4550.
14. Li, T.; Chang, H.; Mishra, S.; Zhang, H.; Katabi, D.; Krishnan, D. Mage: Masked generative encoder to unify representation learning and image synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 2142–2152.
15. Xiao, A.; Huang, J.; Guan, D.; Zhang, X.; Lu, S.; Shao, L. Unsupervised point cloud representation learning with deep neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 11321–11339. [CrossRef] [PubMed]
16. Eckart, B.; Yuan, W.; Liu, C.; Kautz, J. Self-supervised learning on 3d point clouds by learning discrete generative models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 21–24 June 2021; pp. 8248–8257.
17. Chhipa, P.C.; Upadhyay, R.; Saini, R.; Lindqvist, L.; Nordenskjold, R.; Uchida, S.; Liwicki, M. Depth contrast: Self-supervised pretraining on 3dpm images for mining material classification. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; pp. 212–227.
18. Afham, M.; Dissanayake, I.; Dissanayake, D.; Dharmasiri, A.; Thilakarathna, K.; Rodrigo, R. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 9902–9912.
19. Huang, S.; Xie, Y.; Zhu, S.-C.; Zhu, Y. Spatio-temporal self-supervised representation learning for 3d point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 6535–6545.
20. Liu, K.; Xiao, A.; Zhang, X.; Lu, S.; Shao, L. Fac: 3d representation learning via foreground aware feature contrast. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 9476–9485.
21. Wang, H.; Liu, Q.; Yue, X.; Lasenby, J.; Kusner, M.J. Unsupervised point cloud pre-training via occlusion completion. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 9782–9792.
22. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
23. 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, New Orleans, LA, USA, 18–24 June 2022; pp. 19313–19322.
24. Pang, Y.; Wang, W.; Tay, F.E.; Liu, W.; Tian, Y.; Yuan, L. Masked autoencoders for point cloud self-supervised learning. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; pp. 604–621.
25. Wu, X.; Jiang, L.; Wang, P.-S.; Liu, Z.; Liu, X.; Qiao, Y.; Ouyang, W.; He, T.; Zhao, H. Point Transformer V3: Simpler Faster Stronger. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 4840–4851.
26. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph. (tog)* **2019**, *38*, 1–12. [CrossRef]
27. Chen, X.; Liu, Z.; Xie, S.; He, K. Deconstructing denoising diffusion models for self-supervised learning. *arXiv* **2024**, arXiv:2401.14404.
28. Chen, X.; Ding, M.; Wang, X.; Xin, Y.; Mo, S.; Wang, Y.; Han, S.; Luo, P.; Zeng, G.; Wang, J. Context autoencoder for self-supervised representation learning. *Int. J. Comput. Vis.* **2024**, *132*, 208–223. [CrossRef]
29. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 16000–16009.
30. Xie, Z.; Zhang, Z.; Cao, Y.; Lin, Y.; Bao, J.; Yao, Z.; Dai, Q.; Hu, H. Simmim: A simple framework for masked image modeling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 9653–9663.

31. Li, Z.; Gao, Z.; Tan, C.; Ren, B.; Yang, L.T.; Li, S.Z. General Point Model Pretraining with Autoencoding and Autoregressive. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 20954–20964.
32. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
33. Lin, T.; Wang, Y.; Liu, X.; Qiu, X. A survey of transformers. *AI Open* **2022**, *3*, 111–132. [CrossRef]
34. Zhang, R.; Xu, L.; Yu, Z.; Shi, Y.; Mu, C.; Xu, M. Deep-IRTarget: An automatic target detector in infrared imagery using dual-domain feature extraction and allocation. *IEEE Trans. Multimed.* **2021**, *24*, 1735–1749. [CrossRef]
35. Liu, Y.; Zhang, Y.; Wang, Y.; Hou, F.; Yuan, J.; Tian, J.; Zhang, Y.; Shi, Z.; Fan, J.; He, Z. A survey of visual transformers. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *35*, 7478–7498. [CrossRef] [PubMed]
36. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in vision: A survey. *ACM Comput. Surv. (CSUR)* **2022**, *54*, 1–41. [CrossRef]
37. Guo, M.-H.; Cai, J.-X.; Liu, Z.-N.; Mu, T.-J.; Martin, R.R.; Hu, S.-M. Pct: Point cloud transformer. *Comput. Vis. Media* **2021**, *7*, 187–199. [CrossRef]
38. Pan, X.; Xia, Z.; Song, S.; Li, L.E.; Huang, G. 3d object detection with pointformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 21–24 June 2021; pp. 7463–7472.
39. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.H.; Koltun, V. Point transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 16259–16268.
40. Zhang, Y.; Lin, J.; Li, R.; Jia, K.; Zhang, L. Point-MA2E: Masked and Affine Transformed AutoEncoder for Self-supervised Point Cloud Learning. *arXiv* **2022**, arXiv:2211.06841.
41. Kolodiazhnyi, M.; Vorontsova, A.; Konushin, A.; Rukhovich, D. Oneformer3d: One transformer for unified point cloud segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 20943–20953.
42. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9729–9738.
43. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. Pointcnn: Convolution on x-transformed points. *arXiv* **2018**, arXiv:1801.07791.
44. Qiu, S.; Anwar, S.; Barnes, N. Geometric back-projection network for point cloud classification. *IEEE Trans. Multimed.* **2021**, *24*, 1943–1955. [CrossRef]
45. Cheng, S.; Chen, X.; He, X.; Liu, Z.; Bai, X. Pra-net: Point relation-aware network for 3d point cloud analysis. *IEEE Trans. Image Process.* **2021**, *30*, 4436–4448. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.