

<https://doi.org/10.1038/s40494-025-02057-3>

# Self-supervised learning via disentangled representation and self-distillation for 3D terracotta warriors

Check for updates

Xin Cao, Linrong Ye, Xinxin Han, Jia Zhang, Xingxing Hao, Linzhi Su &amp; Kang Li✉

The restoration of the Terracotta Warriors faces challenges due to the lack of large-scale, high-quality annotated datasets. We present PointDecoupler, a novel contrastive learning framework that explicitly disentangles augmentation-invariant representations (AIR) and augmentation-variant representations (AVR) to improve efficiency and generalization. Our method includes two core components: (1) a novel decoupling architecture with an adaptive loss function that systematically disentangles and utilizes AVR information to optimize downstream adaptability; (2) a cross-layer contrastive mechanism inspired by self-distillation, enabling intermediate layers to acquire discriminative features from the final layer. This dual strategy improves feature quality and supports early-exit subnetworks that reduce computational cost without sacrificing performance. Experiments on standard point cloud benchmarks demonstrate consistent gains in classification and segmentation. We further applied PointDecoupler to the Terracotta Warriors dataset, achieving promising results and demonstrating its potential in cultural heritage restoration.

The Terracotta Warriors, as a symbol of Chinese civilization and national identity, have suffered severe fragmentation due to natural erosion and human activities, posing significant challenges for their conservation and restoration<sup>1</sup>. Traditional manual restoration methods are laborious and slow for large-scale projects<sup>2</sup>, making advanced digital preservation crucial. Laser scanning technology now offers new ways to digitally preserve and restore these artifacts. Accurate fragment classification is key for effective matching and assembly, while precise segmentation helps overcome the lack of annotated data and supports structural and feature analysis of the Terracotta Warriors. Traditional feature extraction methods predominantly rely on handcrafted descriptors requiring substantial domain expertise<sup>3,4</sup>. Recent deep learning approaches like PointNet<sup>5</sup>, PointNet++<sup>6</sup>, and DGCNN<sup>7</sup> demonstrate strong performance but depend heavily on large annotated datasets. The annotation process for 3D point clouds remains particularly challenging due to structural complexity and interface limitations, resulting in scarce high-quality real-world data. While synthetic datasets partially mitigate this issue, domain shift between synthetic and real data persists as a major performance limitation.

Self-supervised learning (SSL), successful in NLP and 2D vision, presents a promising alternative by extracting discriminative features without labels<sup>8</sup>. Through pretraining on unlabeled data followed by targeted fine-tuning, this paradigm provided robust model initialization and enhanced downstream performance. In recent years, self-supervised representation learning has emerged as a mainstream approach for point cloud

representation learning. Numerous reconstruction-based self-supervised methods have been proposed, such as FoldingNet<sup>9</sup>, MAP-VAE<sup>10</sup>, and 3DCapsuleNet<sup>11</sup>, which employ self-reconstruction as a means for feature learning. Current 3D self-supervised methods typically employ encoder-decoder architectures with reconstruction objectives<sup>9,12</sup>. Self-supervised methods utilizing Transformers also have gained significant attention. For instance, Point-BERT<sup>13</sup> predicts discrete tokens using a masked modeling task, while Point-MAE<sup>14</sup> learns representations by randomly masking patches of point clouds and reconstructing the missing parts. Similarly, OcCo<sup>15</sup> conducts self-supervised pretraining by reconstructing intentionally occluded regions of point clouds, facilitating the learning of geometric-aware representations. To further improve robustness against geometric transformations, RI-MAE<sup>16</sup> introduces a rotation-invariant masked auto-encoder that enhances generalization under spatial rotations. Fan et al.<sup>17</sup> propose a method that predicts masked patch positions instead of reconstructing fine-grained geometry, improving semantic learning efficiency. In parallel, diffusion-based models have emerged as promising alternatives for point cloud modeling and generation. DiffPMAE<sup>18</sup> leverages denoising-based reconstruction to improve large-scale point cloud representation, while PointDif<sup>19</sup> reconstructs complete 3D shapes by progressively denoising from Gaussian noise, without relying on label supervision. Sun et al.<sup>20</sup> further extend diffusion models to tasks such as shape reconstruction and pose estimation, demonstrating their broader applicability in 3D vision. However, due to the sparse and discrete nature of point cloud data,

School of Computer Science, Northwest University, Xi'an, China. ✉e-mail: [likang@nwu.edu.cn](mailto:likang@nwu.edu.cn)

reconstructing 3D objects is not only computationally expensive and inefficient, but also difficult to capture fine geometric details.

Contrastive Learning's success in 2D vision has inspired its application to point cloud self-supervised representation learning. Unlike reconstruction methods, it emphasizes learning discriminative features by minimizing distances between positive pairs and maximizing those between negative pairs via contrastive loss<sup>21</sup>. However, most current 3D point cloud contrastive learning methods are adapted from 2D vision and often overlook the unique characteristics of 3D data. For instance, PPKT<sup>22</sup> transfers image pre-trained models to point clouds through transformations, Jigsaw3D adapts the 2D jigsaw puzzle task to 3D, PointContrast<sup>23</sup> focuses on multi-view consistency for high-level scene understanding, and CrossPoint<sup>24</sup> combines 3D point clouds with 2D images for cross-modal learning. STRL<sup>25</sup> extends BYOL<sup>26</sup> to 3D, using online and target network interactions. To enhance positive and negative pair design, Du et al.<sup>27</sup> propose using self-similar patches within a point cloud. Additionally, Zhang et al.<sup>28</sup> explore integrating contrastive learning with online clustering.

Disentangled Representation Learning (DRL) aims to decompose latent generative factors into semantically meaningful components, thereby improving model interpretability, controllability, robustness, and generalization. DRL has been widely applied across domains such as image generation, video analysis, NLP, multimodal learning, and recommendation systems. Existing methods can be categorized by model type, representation structure, supervision signal, and independence hypothesis. From the perspective of model types, generative models have proven to be powerful tools for disentangled representation learning. For instance, Variational Autoencoders<sup>29</sup> are classical examples that learn latent representations by maximizing the Evidence Lower Bound. VAEs and their variants (e.g.,  $\beta$ -VAE, DIP-VAE)<sup>30</sup> promote disentanglement via regularized latent spaces, while GAN-based methods such as InfoGAN<sup>31</sup> leverage mutual information maximization. Recent diffusion models like DisDiff<sup>32</sup> further advance disentanglement by minimizing mutual information between latent variables. Beyond generative models, discriminative SSL approaches have recently explored representation decoupling within the contrastive learning paradigm. For instance, FactorCL<sup>33</sup> and DeCUR<sup>34</sup> decompose representations based on data augmentations or modality differences using mutual information bounds and redundancy reduction. From the perspective of representation structure, DRL methods can be classified as dimensional-wise or vector-wise (based on factor granularity) and flat or hierarchical (based on structural organization). Generative models like VAEs and InfoGANs are often used for fine-grained disentanglement tasks, while vector-wise approaches like DR-GAN<sup>35</sup> and MAP-IVR<sup>36</sup> target complex real-world tasks. Hierarchical disentanglement has been explored via hierarchical VAEs<sup>37</sup>, specialized loss functions<sup>38</sup>, and partial activation hypotheses. Supervision strength also influences disentanglement quality: unsupervised models rely on reconstruction or adversarial loss; weakly supervised ones (e.g., DisUnknown<sup>39</sup>, ML-VAE)<sup>40</sup> incorporate partial labels or sample grouping; and fully supervised methods (e.g., DC-IGN<sup>41</sup>, DNA-GAN)<sup>42</sup> control variation in latent variables or align them with label semantics. Additionally, independence-based approaches like CausalVAE<sup>43</sup> and DEAR<sup>44</sup> employ Structural Causal Models to achieve causal disentanglement, with DEAR further incorporating weak supervision losses. While disentangled representation learning has achieved remarkable progress in areas like image and video generation, its application in contrastive learning remains in an exploratory stage, with even fewer studies focusing on the field of 3D point clouds.

Knowledge Distillation, introduced by Hinton et al.<sup>45</sup>, transfers knowledge from a complex teacher model to a compact student model to reduce computational and storage demands while enhancing student model performance. Traditional offline distillation involves pre-training a teacher model to guide the student model via soft labels or intermediate representations. However, this approach is computationally expensive. Online distillation addresses this by synchronously training teacher and student models together. Self-distillation methods have recently been explored, where student models extract knowledge from their own multilayer

representations. Li et al.<sup>46</sup> proposed a self-distillation framework for long-tail recognition tasks using mixed supervision from soft and hard labels. Xu et al.<sup>47</sup> introduced a self-distillation approach for 2D images by leveraging structured knowledge from auxiliary tasks in SSL. These methods reduce reliance on external teacher models. Although knowledge distillation has made progress in 2D image tasks, its application to 3D point clouds remains underexplored. The unique challenges of 3D point cloud data necessitate new knowledge distillation approaches. Some studies have attempted to extend distillation to 3D point cloud tasks. For example, PointDistiller uses local distillation and reweighted learning to compress 3D inspection models, significantly improving inspection performance. However, these methods mainly focus on model compression for point cloud tasks and lack exploration of self-supervised representation learning for 3D data.

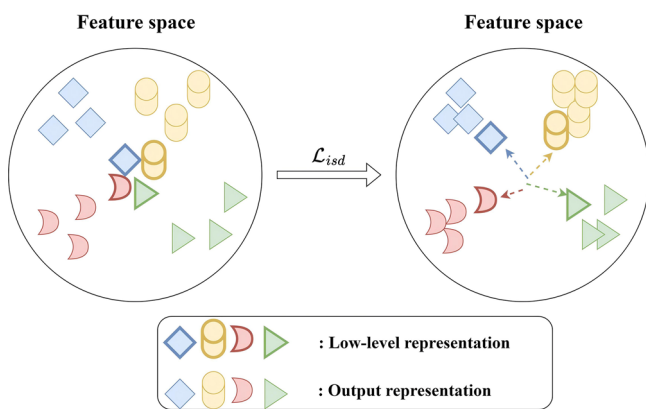
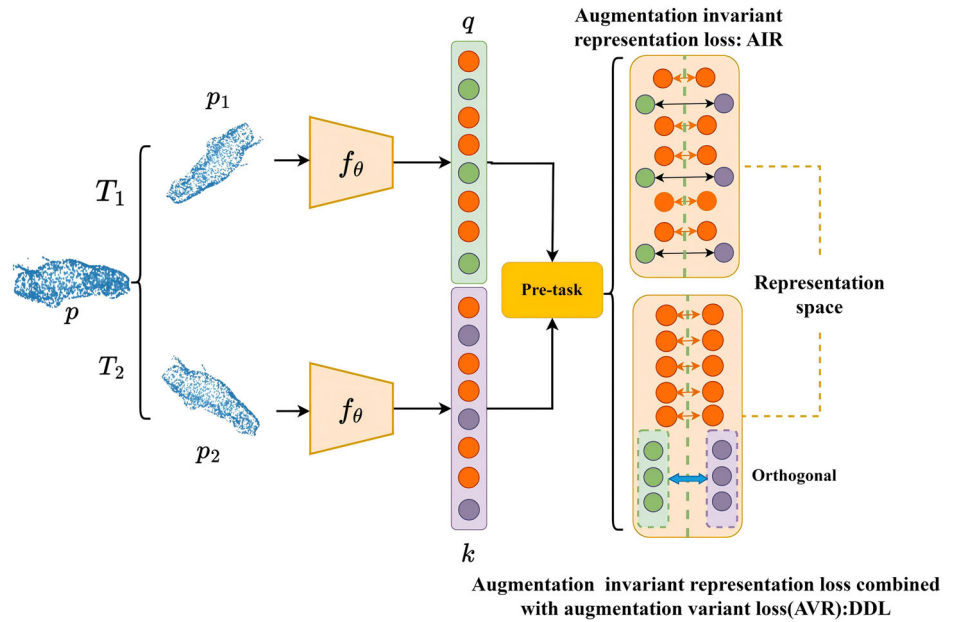
Contemporary 2D vision research demonstrates that explicit characterization of AVR complements AIR to enhance downstream performance. Current point cloud methods predominantly focus on AIR while neglecting AVR from geometric transformations (rotation, translation, and noise), thereby limiting model generalization. For instance, STRL's exclusive AIR supervision through loss functions induces oversensitivity to specific augmentation strategies. Building on DDCL's 2D insights<sup>48</sup>, we posit that unsupervised AVR causes feature entanglement with AIR in representation space, ultimately degrading model capacity. This motivates our explicit AIR-AVR decoupling strategy.

We propose PointDecoupler, a novel SSL framework that explicitly disentangles augmentation-invariant representation (AIR) and augmentation-variant representation (AVR) to enhance feature learning and downstream performance. The main contributions of this paper are summarized as follows: (1) we present PointDecoupler, a streamlined contrastive learning framework that eliminates reconstruction complexity while enhancing representational capacity through exclusive use of positive sample pairs; (2) we propose the orthogonality-constrained decoupling loss that systematically disentangles invariant and variant representations in contrastive learning paradigms, enabling independent AVR learning while jointly supervising both representations (Fig. 1); and (3) we pioneer the integration of self-distillation with representation decoupling, enabling efficient hierarchical feature learning from positive samples alone. This approach achieves representation alignment from intermediate to final layers by integrating self-distillation (Fig. 2). Unlike conventional self-distillation methods that match final outputs, our approach aligns hierarchical features to ensure early-exit subnetworks achieve full-network performance levels, thereby enhancing feature discriminability and task performance. The framework eliminates negative sampling dependencies while maintaining positive-sample exclusivity. By maximizing similarity between augmented variants and disentangling AVR, we enhance mid-layer representations for improved instance discrimination. Our method achieves state-of-the-art performance without cross-modal data or reconstruction objectives. Comprehensive evaluations on ModelNet, ScanObjectNN, ShapeNetPart, and S3DIS demonstrate significant improvements in shape classification, part segmentation, and semantic segmentation tasks, with direct applications to Terracotta Warrior restoration.

## Methods

Deep neural networks efficiently transfer learned representations across datasets and target tasks, but the traditional supervised learning paradigm encounters significant limitations when data or labels are scarce. To address this, SSL has gained traction for its label-free training and strong performance. Among SSL strategies, contrastive learning has shifted from using numerous positive and negative samples to relying solely on positive pairs. Recent study<sup>49</sup> highlights that improving a model's sensitivity to specific data characteristics can significantly enhance the quality of its representations. Focusing on point cloud data, we revisit positive-only contrastive learning and emphasize the need to disentangle AIR and AVR. We propose PointDecoupler, a novel self-supervised framework that avoids decoders and negative samples. As shown in Fig. 3, it comprises: (1) augmentation decoupling module: separates AIR and AVR in feature space to learn more

**Fig. 1 | Simplified diagram of PointDecoupler.** Unlike traditional methods that only pull positive pairs (orange) closer in the representation space, PointDecoupler further enforces orthogonality between augmentation-variant components (green and purple), enabling explicit disentanglement of AIR and AVR.



**Fig. 2 | Representations on a hypersphere.** The student’s lower layer and the teacher’s output are projected onto the hypersphere. The intermediate self-distillation loss explicitly aligns the representations of the student’s lower layer with those of the teacher’s output, effectively guiding the lower-layer representations toward the output representations and enhancing feature consistency.

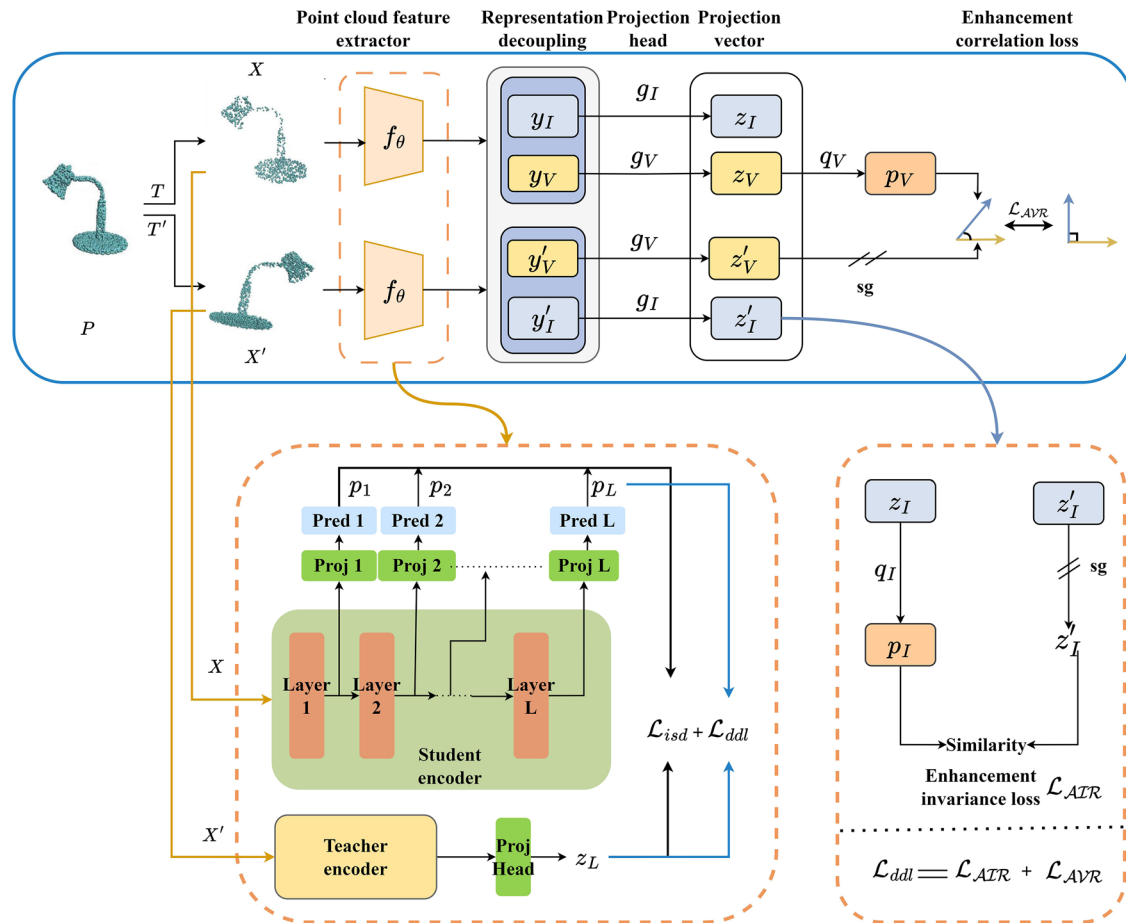
discriminative and generalizable representations. (2) Self-distillation module: aligns intermediate and final layer outputs, enabling early-exit sub-networks to perform comparably to full models and improving feature extraction.

Specifically, given a set of point cloud data  $P = \{P_i \in \mathbb{R}^{N \times 3}\}_{i=1}^{|P|}$ , we apply data augmentation operations, such as rotation, scaling, and dithering, to each sample  $p$  to generate augmented samples  $x$  and  $x'$ . These are fed into the feature extractor  $f_\theta$  to extract high-dimensional feature representations. As shown in Fig. 3, the framework consists of two key modules: (1) enhanced decoupling module (blue box): the feature extractor  $f_\theta$  maps the input  $x$  to a feature  $y$ , which is subsequently decoupled into the augmented invariant representation  $y_I$  and the augmented variable representation  $y_V$ . These are projected into the embedding space using the projection heads  $g_I$  and  $g_V$  to obtain:  $z_I = g_I(f(x; \theta_f); \psi_g)$ ,  $z_V = g_V(f(x; \theta_f); \psi_g)$ . Here,  $\theta_f$  and  $\psi_g$  denote the trainable parameters of the feature extractor and projection head, respectively. Next, the augmented invariant embedding  $z_I$  is passed through the predictor  $q_I$  to predict the corresponding  $z'_I$  output of the lower

branch network, while the augmented variable embedding  $z_V$  is passed through the predictor  $q_V$  to compute the orthogonal loss with the corresponding  $z'_V$  output of the lower branch network. (2) The self-distillation module (orange dashed box): to enhance the feature extractor  $f_\theta$ , intermediate layer features are extracted from each sub-layer after  $x$  was input into  $f_\theta$ . These intermediate layer features are decoupled and subsequently mapped into the embedding space via their respective projection heads, resulting in the representations  $z_l$  for each layer, where  $l$  denotes the layer index of the feature extractor. As shown in Fig. 3,  $z_l$  and  $p_l$  are abbreviations for the intermediate representations and predictors, respectively. Taking  $z_l$  as an example, it is further decoupled into  $z'_l$  and  $z'_V$ , and passed through corresponding predictors to produce  $p'_l$  and  $p'_V$ , which are then used to compute the loss with the final outputs  $z'_I$  and  $z'_V$  of the student network. By aligning the intermediate layer representations with the final output representations, the self-distillation module ensures strong model performance at various depth truncations. The core objective of PointDecoupler is to train the point cloud feature extractor  $f_\theta$  in a self-supervised manner, such that all projection heads and predictors can be discarded in downstream tasks.

### Representation decoupling

Some research demonstrates that model performance in downstream tasks is influenced by both augmentation sensitivity and task specificity. Existing contrastive learning methods often require hand-crafted architectures or losses tailored to specific augmentations, limiting their generality. The representation decoupling module introduced in this section enables the extraction of AVR without the need for augmentation-specific adaptations. Rather than excluding augmentations, it captures the variations induced by transformations (e.g., rotation and deformation), thereby improving the model’s robustness and generalization to unseen data. During the model training process, AVR explicitly reflects the impact of augmentation operations, which enhances the model’s generalization ability to non-augmented data and facilitates learning more robust and generalized feature representations. To enable the model to effectively learn augmentation-related information, this section splits the output features of the last layer of the point cloud feature encoder into two components: AIR and AVR. These two components are supervised simultaneously by their respective loss functions, namely AIR loss



**Fig. 3 | The architecture of PointDecoupler.** The blue part is the representative decoupling module, and the orange dotted box area in the lower left corner is the self-distillation decoupling module.

and AVR loss, to ensure effective learning of distinct characteristics. The specific decoupling formula is as follows (taking the upper branch as an example):

$$y_I = Matrix_{n,I} \cdot f_{1:n-1}(x) \tag{1}$$

$$y_V = Matrix_{n,V} \cdot f_{1:n-1}(x) \tag{2}$$

$$y = cat(y_I, y_V) \tag{3}$$

$$Matrix_{n,I} \in \mathbb{R}^{DR \cdot d \times C_n}, Matrix_{n,V} \in \mathbb{R}^{(1-DR) \cdot d \times C_n}$$

Here,  $Matrix_n$  represents the weight matrix of the final layer in the feature extractor, while  $f_{1:n-1}(x)$  denotes the output of the preceding layers of the feature extractor, excluding the final layer. The operator  $cat(\cdot)$  refers to the feature concatenation operation, and DR stands for the Disentangling Ratio, which controls the balance between AIR and AVR to enable effective disentanglement. In this context, d represents the dimensionality of the point cloud feature vector output by  $f_{1:n-1}(x)$ , and  $C_n$  is the dimensionality of the final layer of the network.

For the supervision of the AIR component, to ensure that the distances between features of differently augmented point cloud data in the feature space remain consistent, this section utilizes the Euclidean distance to measure the degree of similarity between positive sample pairs. Meanwhile, feature learning is achieved by optimizing the prediction error between the teacher network and the student network. The specific loss function is

defined as follows:

$$\mathcal{L}_{AIR} = \frac{1}{2} S(p_I, sg(z'_I)) + \frac{1}{2} S(p'_I, sg(z_I)) \tag{4}$$

$$S(p, z') = \|p - z'\|_2^2 = 2 - 2 \cdot \frac{\langle p, z' \rangle}{\|p\|_2 \|z'\|_2} \tag{5}$$

Where  $p_I$  and  $p'_I$  are the output of the prediction head,  $z_I$  and  $z'_I$  are the output of the projection head, sg represents the gradient stop operation, and  $\langle \cdot, \cdot \rangle$  represents the dot product of the vector.

For the supervision of AVR, we propose a novel loss function called Augmented Decoupling Loss. The design objective of this loss function is to ensure that the projection vectors generated by the same set of point cloud data under different augmented versions are orthogonal to each other. Taking point cloud data augmentation as an example, translation operations and scaling operations, respectively, cause changes in the position and scale of the point cloud, resulting in significantly different features generated by these two augmentation methods. By constraining the orthogonality of AVR features, the model is encouraged to separately learn the feature information introduced by these distinct augmentation methods. Moreover, the orthogonality of AVR features provides the model with greater adaptability. Since different augmentation strategies affect data characteristics in different ways, this property allows the model to effectively learn and distinguish these changes, thereby enhancing its robustness and generalization ability under various augmentation scenarios. The formula for augmented

decoupling loss is as follows:

$$\mathcal{D}(z_v, z'_v) \triangleq \left| \frac{z_v \cdot z'_v}{\|z_v\|_2 \|z'_v\|_2} - \xi \right| \quad (6)$$

$$\mathcal{L}_{AVR} \triangleq \frac{1}{2} \mathcal{D}(p_v, \text{sg}(z'_v)) + \frac{1}{2} \mathcal{D}(p'_v, \text{sg}(z_v)) \quad (7)$$

The hyperparameter  $\xi$  is set to 0 by default, and  $\text{sg}$  indicates that the gradient stops the operation. Therefore, we can give the final decoupling loss:

$$\mathcal{L}_{ddl} = \gamma \mathcal{L}_{AVR} + \mathcal{L}_{AIR} \quad (8)$$

Here,  $\gamma$  is set to 1 by default. Only the parameters of the student network and the predictor are updated during each training round. In contrast, the parameters of the teacher network are updated using the exponential moving average (EMA) after each training round. This strategy, similar to the approach used in BYOL, is adopted in our method but will not be elaborated upon here.

### Self-distillation

To further enhance the representational capacity of the model, we introduce self-distillation technology on the basis of distortion decoupling. The network framework is illustrated in Fig. 3. The overall network consists of two branches: the teacher network and the student network. Both branches share the same core structure; however, the key difference lies in the fact that the output of each layer in the student network is projected through an independent projection head, while the output of the teacher network is predicted by a predictor. It is worth noting that the output of each layer in the student network is also decoupled and combined with the decoupling loss for calculation. Here, we describe the approach from a macro perspective and do not delve into these details.

Inspired by self-distillation methods in supervised learning, we extend this concept by enabling the intermediate layer representations of the student network to learn from the final layer representations through contrastive loss. By introducing a self-distillation mechanism, we guide the intermediate layer representations of the student network to mimic the output representations of the teacher network, thereby providing an explicit learning signal to the intermediate layers. This design reduces the performance loss when the student network exits early through a subnetwork, narrowing the gap between the subnetwork's performance and that of the full network. At the same time, this approach alleviates the burden on the final layer to perform the pretext task, resulting in overall better feature representations. Moreover, the self-distillation loss provides an additional supervisory signal to the intermediate layers, which significantly benefits model training. On the one hand, it accelerates model convergence, allowing the model to achieve ideal performance in fewer training iterations, thereby saving computational resources and time costs. On the other hand, this additional supervisory signal improves training stability, reducing fluctuations and instability during the training process, and thus further enhancing the reliability of the model.

The intermediate layer self-distillation loss  $\mathcal{L}_{isd}$  is first given, which attempts to maximize the mutual information of the intermediate layer  $l$  and the final layer  $L$  output, as follows:

$$\mathcal{L}_{isd} = \frac{1}{L-1} \sum_{l=1}^{L-1} \mathcal{L}_{ddl}(p_l, \text{sg}(z_L)) \quad (9)$$

Among them,  $p_l$  is the representation of the  $L$ -layer of the student encoder, which is obtained by the feature projection of the corresponding predictor.  $z_L$  is the output result of the teacher network encoder after passing the projection head. The stop gradient operator  $\text{sg}(z_L)$  means that the

gradient does not propagate through  $z_L$ , so the intermediate layer loss does not affect  $z_L$  when backpropagated.

### Overall joint objectives

Building on the self-distillation loss of the intermediate layers, it works in conjunction with the original contrastive loss  $\mathcal{L}_{ddl}$  to guide the model in learning effective features. Specifically,  $\mathcal{L}_{ddl}$  is responsible for optimizing the network from a global perspective to ensure the quality of the final layer representation, while  $\mathcal{L}_{isd}$  focuses on enhancing the instance discrimination capability of the intermediate layers. This ensures that the final layer can learn more robust and meaningful feature representations. Based on the above analysis, the objective function can be defined as follows:

$$\mathcal{L}_{SDSSL} = \mathcal{L}_{ddl}(p_L, z_L) + \alpha \mathcal{L}_{isd} \quad (10)$$

Among them,  $\alpha$  is used to control the weight of self-distillation losses.

At the same time, it can be observed that for frameworks with predictors, the direct use of formula (10) can bring some performance improvement, but there remains potential for further optimization. This is because the mid-tier predictor is updated solely using gradients from  $\mathcal{L}_{isd}$ , whereas the feature encoder benefits from both  $\mathcal{L}_{isd}$  and  $\mathcal{L}_{ddl}$ . Consequently, the optimality of the mid-tier predictor cannot be fully guaranteed, even though it is a critical component of SSL training, as discussed in ref. 26. Furthermore, simply increasing  $\alpha$  (the weight of  $\mathcal{L}_{isd}$ ) leads to suboptimal performance in the final layer of the encoder. This adjustment inadvertently impacts the middle layers of the backbone network, which negatively affects overall performance. To address this issue, we introduce an additional loss function,  $\mathcal{L}_{pred}$ , to further optimize the performance of the mid-tier predictor. The formulation of this loss function is as follows:

$$\mathcal{L}_{pred} = \sum_{l=1}^L \mathcal{L}_{ddl}(\text{pred}(\text{sg}(h_l)), \text{sg}(z_L)) \quad (11)$$

Where  $h$  is the representation of layer  $l$  after the student passes through the projector, and in order to update only the predictor, the  $\text{sg}(\cdot)$  operator is used for  $h$ . By doing so, a better distillation predictor can be obtained, finally giving the final objective function:

$$\mathcal{L}_{total} = \mathcal{L}_{ddl}(p_L, z_L) + \alpha \mathcal{L}_{isd} + \beta \mathcal{L}_{pred} \quad (12)$$

Here,  $\beta$  is set to 1 by default.

### Ethics approval and consent to participate

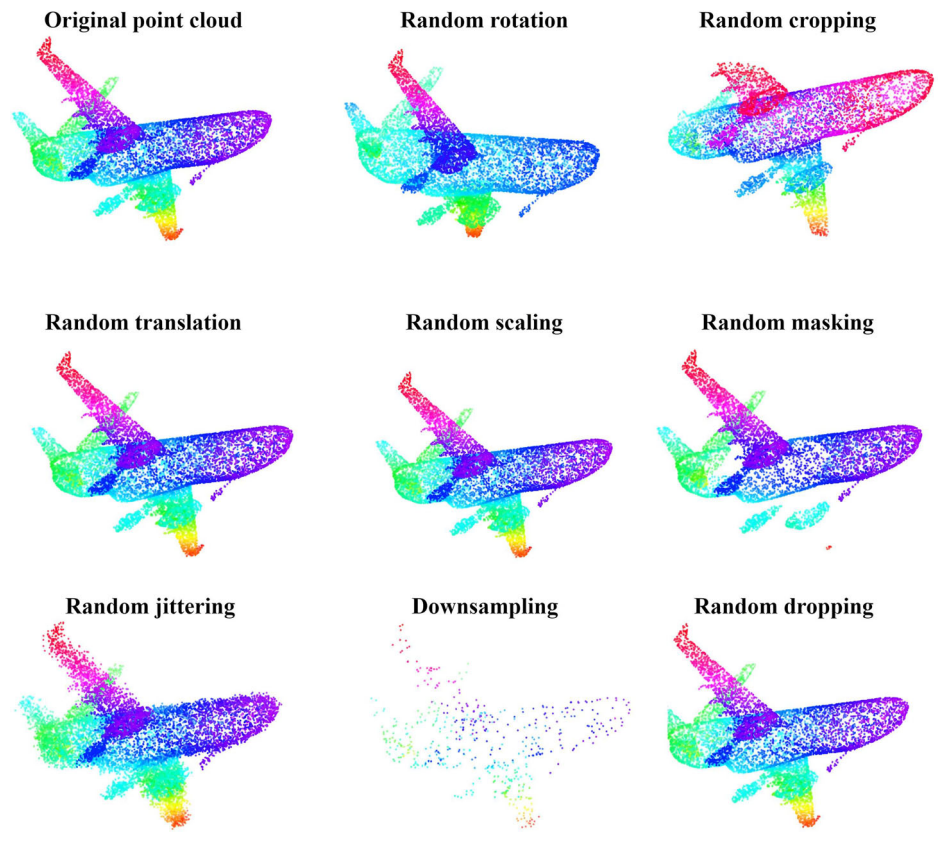
Written informed consent has been obtained from the School of Information Science and Technology of Northwest University and all authors for this article, and consent has been obtained for the data used.

## Results

### Pre-training

For pre-training, we utilized the ShapeNet dataset, a comprehensive repository of synthetic 3D shapes comprising over 50,000 unique models across 55 common object categories. To ensure comparability, we adhered to the training protocol established by STRL. This procedure involved randomly sampling 2048 points from each model in the dataset. The visualized results of the data augmentation techniques we employed are shown in Fig. 4 below. These augmentation techniques can be broadly categorized into two types. The first category is synthetic sequence generation augmentation, which applies global transformations to the point cloud by combining multiple augmentation techniques. Specifically, each technique within this category is applied with a probability of 0.5. For example, random rotation is performed by sampling a random angle for each axis within  $\pm 15^\circ$  and rotating the point cloud accordingly. Random translation is achieved by globally shifting the entire point cloud within 10% of its dimensional bounds. Additionally, random scaling is applied by multiplying the point cloud by a

Fig. 4 | Visualization of different data augmentation.



random scaling factor uniformly sampled in the range [0.8, 1.25]. These operations collectively enable the generation of synthetic sequences that represent global transformations of the point cloud. The second category is spatial augmentation, which focuses on altering the local geometry of the point cloud to enhance the model's ability to learn spatial structure representations. This includes operations such as random cropping, where a 3D cuboid patch is cropped from the point cloud. The volume of this patch is uniformly sampled between 60% and 100% of the original point cloud, and the aspect ratio is controlled within the range [0.75, 1.33]. Random excavation removes a cuboid section from the point cloud, with the dimensions of the removed region sampled within the range [0.1, 0.4] of the original point cloud's dimensions. Random jitter perturbs the 3D positions of points by adding random offsets uniformly sampled within the range [0, 0.05]. Random discard removes a subset of points from the point cloud based on a discard ratio that is uniformly sampled in the range [0, 0.7], while subsampling reduces the point cloud to a fixed size by randomly selecting the required number of points to match the encoder's input dimensions. For synthetic data, normalization is applied to scale the point cloud to fit within a unit sphere. Among these operations, random cropping and random excavation introduce more significant changes to the spatial structure of the point cloud. To leverage their impact effectively, we apply these two operations with a probability of 0.5. These augmentation strategies collectively ensure that the model learns robust and diverse feature representations, improving its performance across various tasks.

For fair comparison, we employ PointNet and DGCNN as point cloud feature extractors. The projection head and predictor in PointDecoupler are designed as multi-layer perceptrons. Each projection head consists of two fully connected (FC) layers, with output dimensions of [4096/2048, 256], including batch normalization and ReLU activation layers. Specifically, the hidden dimension of the projection head for the final layer is set to 4096, while for the middle layer, it is set to 2048. The input to the first FC layer of the projection head is derived from the encoder's output. For the projection head  $g_I$  the input dimension is  $S_{\text{encoder}} \times \text{DR}$ , while for  $g_V$ , the input

dimension is  $S_{\text{encoder}} \times (1 - \text{DR})$ , where  $S_{\text{encoder}}$  represents the output size of the point cloud feature extractor. The predictor has the same structure as the projection head, with a fixed output dimension of [4096, 256]. We use the LARS optimizer along with a cosine decay learning rate schedule. The learning rate schedule includes a warm-up phase lasting 10 epochs, and no restart is required. The parameters of the teacher network are updated using an EMA mechanism. The EMA coefficient  $\tau$  is updated using the formula:  $\tau = 1 - (1 - \tau_{\text{start}}) \cdot (\cos(\frac{\pi k}{K}) + 1)/2$ , where  $k$  is the current training step and  $K$  is the total number of training steps. The learning rate is initialized at 0.001 and follows the cosine decay strategy. Additionally, the weight parameter  $\alpha$  is scheduled using a cosine strategy, gradually increasing from 0 to 1.0. Training is conducted end-to-end. For PointNet, we train with a batch size of 32 for 50 epochs; for DGCNN, a batch size of 4 for 100 epochs. At the end of pre-training, all projection heads  $g(\cdot)$  and predictors  $p(\cdot)$  are discarded, retaining only the encoder  $f(\cdot)$  of the student network for subsequent downstream tasks.

### 3D object classification

The 3D object classification task aims to classify the given point cloud data and accurately predict the specific object class to which each point cloud belongs. We evaluate the shape understanding ability and the generalization performance of pre-trained models using two widely-used benchmark datasets: ModelNet40 and ScanObjectNN. ModelNet40 is a synthetic object dataset consisting of 40 categories and 12,311 CAD models, which we use to evaluate the classification performance on synthetic objects. ScanObjectNN is a highly challenging, realistic 3D point cloud dataset collected from real indoor scenes, which we use to evaluate the performance in real natural scenes. The dataset contains 15 categories with a total of 2880 objects, of which 2304 are used for training and 576 for testing. We evaluate two mechanisms, linear classification and fine-tuning, to verify the effectiveness of the proposed approach. For the linear classification task, we adopt the protocol outlined in previous works<sup>25,50</sup>. Specifically, we train a linear Support Vector Machine on the global features extracted from the training sets

**Table 1 | Comparison of the linear SVM classification on ModelNet40**

Method	ModelNet40
3D-GAN <sup>51</sup>	83.3
Latent-GAN	85.7
SO-Net <sup>52</sup>	87.3
RotationNet <sup>53</sup>	90.4
FoldingNet <sup>9</sup>	88.4
MRTNet <sup>54</sup>	86.4
3D-PointCapsNet <sup>11</sup>	88.9
MAP-VAE <sup>10</sup>	88.4
VIP-GAN <sup>55</sup>	90.2
DepthContrast <sup>56</sup>	85.4
ClusterNet <sup>57</sup>	86.8
Jigsaw <sup>50</sup> + PointNet	87.3
Rotation <sup>58</sup> + PointNet	88.6
OcCo <sup>15</sup> + PointNet	88.7
STRL <sup>25</sup> + PointNet	88.3
CrossPoint <sup>24</sup> + PointNet	89.1
<b>PointDecoupler (ours) + PointNet</b>	<b>89.3</b>
Self-Contrast + DGCNN	89.6
Jigsaw + DGCNN	90.6
Rotation + DGCNN	90.8
STRL + DGCNN	90.9
OcCo + DGCNN	89.2
CrossPoint + DGCNN	91.2
<b>PointDecoupler (ours) + DGCNN</b>	<b>91.5</b>

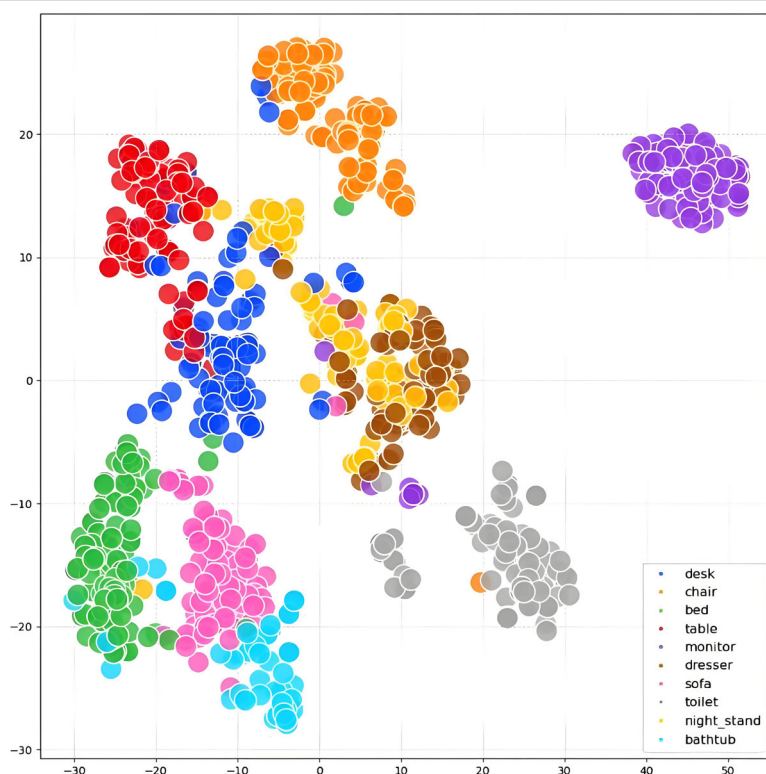
The linear classifier is fitted on the training set of ModelNet40 using the pre-trained model, and the model performance is evaluated on the test set.

of ModelNet40 and ScanObjectNN and evaluate its performance on the corresponding test sets. For the fine-tuning task, we conduct experiments on the ModelNet40 dataset. During the fine-tuning process, the parameter settings follow the scheme proposed in ref. 5, with the training epochs reduced from 250 to 125. This adjustment is made because pre-trained weights accelerate the convergence of supervised training, thereby reducing the computational time and resources required for training.

Table 1 presents the experimental results of the linear classification task on the ModelNet40 dataset. Regardless of whether PointNet or DGCNN is used as the backbone network, the methods proposed in this study outperform the state-of-the-art approaches on the ModelNet40 dataset. Notably, while CrossPoint relies on multimodal data, the proposed PointDecoupler is based solely on single-modal point cloud data. Despite this, PointDecoupler achieves classification accuracy improvements of 0.2% and 0.3% over CrossPoint when using PointNet and DGCNN as backbone networks, respectively. Furthermore, even with a simpler backbone like PointNet, PointDecoupler outperforms many SSL methods built on more complex architectures.

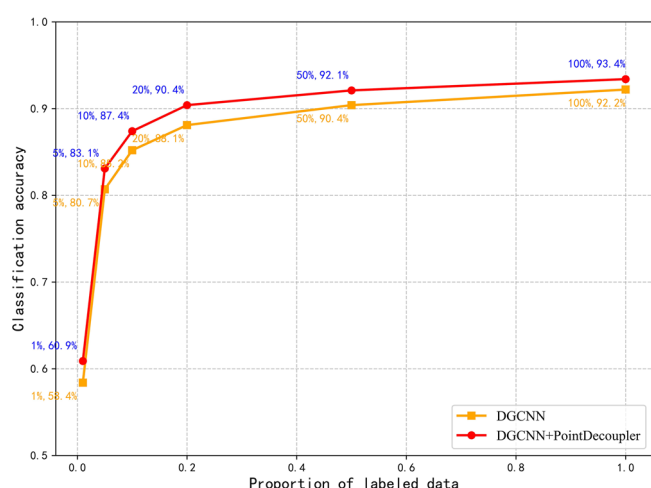
Figure 5 illustrates the feature visualization results after pre-training the model in a self-supervised manner using the PointNet backbone network. In this analysis, we apply the t-SNE technique to reduce the dimensionality of features extracted from the test set of the ModelNet10 dataset for visualization. As shown in Fig. 5, the design of representational decoupling and self-distillation enables effective categorization, even without explicit supervised training using labeled data. This demonstrates that the SSL approach proposed in this study can effectively capture discriminative features between categories, exhibiting strong generalization capabilities in scenarios with unlabeled data. We further evaluate the learned point cloud representation model through supervised fine-tuning. Specifically, on the ModelNet40 dataset, the encoder weights of the pre-trained model are used as the initial weights for the point cloud feature extractor. The DGCNN network is fine-tuned using the dataset labels. The results, presented in Table 2, show that the proposed PointDecoupler improves the final classification accuracy by 1.2% compared to DGCNN initialized with random weights.

**Fig. 5 | PointDecoupler's t-SNE feature visualization on ModelNet10.**



**Table 2 | Classification results of PointDecoupler on ModelNet40 using fine-tuning strategy**

Method	Supervised	OA
PointNet <sup>5</sup>	√	89.2
PointNet++ <sup>6</sup>	√	90.7
PointCNN <sup>59</sup>	√	92.2
DGCNN <sup>7</sup>	√	92.2
ShellNet <sup>60</sup>	√	93.1
PointTransformer <sup>61</sup>	√	93.7
Transformer-OcCo <sup>15</sup>	×	92.1
Point-Bert <sup>13</sup>	×	93.2
OcCo	×	93.0
Point-MPP <sup>17</sup>	×	93.3
<b>PointDecoupler</b>	×	<b>93.4</b>



**Fig. 6 | Classification accuracy of different proportions of labeled data.**

Additionally, even without leveraging the Transformer framework, PointDecoupler achieves classification accuracy comparable to that of Point-BERT.

We also demonstrate the effectiveness of the proposed pre-training model in semi-supervised learning scenarios, particularly in cases with limited labeled samples, where significant improvements in classification performance are observed. Specifically, experiments are conducted by randomly selecting different proportions of labeled training data while ensuring that at least one sample is included for each category. The pre-trained model is then fine-tuned on these limited samples using supervised training, and its classification performance is evaluated on the full test set. The experimental results, shown in Fig. 6, indicate that when the proportion of labeled training samples is 1% and 20%, the proposed model improves classification accuracy by 2.5% and 2.3%, respectively. These findings suggest that the SSL method proposed in this study can more effectively enhance the performance of downstream tasks, particularly in scenarios with fewer labeled samples.

To assess practical feasibility, we report parameter counts and FLOPs on the ModelNet40 object classification task (Table 3). PointDecoupler achieves 93.4% accuracy with only 4.2 M parameters and 2.3 G FLOPs, without relying on Transformer architectures. Its self-distillation-based design enables the model to achieve competitive performance with substantially reduced computational complexity, highlighting its potential for efficient deployment.

To validate the effectiveness of the proposed method on real-world point cloud data, we evaluated the classification task on the ScanObjectNN

**Table 3 | Computational efficiency comparison on ModelNet40 for object classification**

Method	Params. (M)	FLOPs (G)	OA (%)
PointNet <sup>5</sup>	3.5	0.5	89.2
PointNet++ <sup>6</sup>	1.5	1.7	90.7
DGCNN <sup>7</sup>	1.8	2.4	92.2
PCT	2.9	2.3	93.2
PointMAE <sup>14</sup>	22.1	2.4	93.2
Point-Bert <sup>13</sup>	22.1	2.4	92.7
PointDecoupler (ours)	4.2	2.3	93.4

**Table 4 | Comparison of classification on ScanObjectNN**

Encoder	Method	Acc.
PointNet	OcCo <sup>15</sup>	69.5
	Jigsaw <sup>60</sup>	55.2
	STRL <sup>25</sup>	74.2
	CrossPoint <sup>24</sup>	75.6
	<b>PointDecoupler (ours)</b>	<b>75.8</b>
DGCNN	OcCo <sup>15</sup>	78.3
	Jigsaw <sup>60</sup>	59.5
	STRL <sup>25</sup>	77.9
	CrossPoint <sup>24</sup>	81.7
	<b>PointDecoupler (ours)</b>	<b>82.5</b>

PointDecoupler achieves improvements compared to other self-supervised methods on both PointNet and DGCNN, which illustrates the effectiveness of our method in real-world scene classification.

dataset. Table 4 reports the linear classification results on this dataset. Compared to state-of-the-art self-supervised methods, the classification accuracy of the proposed PointDecoupler improves significantly by 0.8% when DGCNN is used as the backbone network. This result demonstrates that the feature representations learned by PointDecoupler can be effectively transferred from synthetic data to real-world point cloud scenarios, showcasing strong generalization capabilities.

### 3D object part segmentation

Object part segmentation is an important and challenging 3D recognition task, where the goal is to assign a part class label to each point, such as a table leg or a car tire. We conducted experiments on the ShapeNetPart dataset, which contains 16,991 objects from 16 categories, with a total of 50 parts distributed among 2–6 parts per object. As a widely used benchmark dataset, ShapeNetPart can effectively evaluate the performance of object part segmentation methods. We used DGCNN as the backbone network in the pre-training phase and applied fine-tuning to optimize the performance of our method. To evaluate the performance of our method, we used the mean intersection over union (mIoU) metric, which is highly accurate and commonly used.

As shown in the experimental results in Table 5, the proposed PointDecoupler method achieves significant improvements in component segmentation performance, surpassing many state-of-the-art methods in terms of the mIoU metric. Moreover, compared to other SSL methods, PointDecoupler performs consistently well on both the PointNet and DGCNN backbone networks. This demonstrates that the joint design of the self-distillation module and the representational decoupling module enables the model to effectively learn more discriminative feature representations.

Further analysis of the results reveals that the proposed model can capture fine-grained local features, thereby achieving more precise part segmentation. Figure 7 illustrates the part segmentation results, highlighting the exceptional performance of the proposed method in handling fine

details across different categories. For instance, the model is able to clearly distinguish the wheels of a motorcycle and the tail of an aircraft from other parts, fully showcasing its ability to capture detailed features and deliver superior segmentation performance.

As introduced in Pre-training, our framework combines global transformations (synthetic sequence generation) and local geometric changes (spatial augmentation). To evaluate their roles, we ablate each augmentation during pretraining on ModelNet40 and assess frozen features via linear SVM (Table 6). Removing global transformations (e.g., rotation and scaling) leads to minor drops, while removing local augmentations (e.g., crop and excavation) results in larger declines, confirming their importance for robust, occlusion-invariant features. We hypothesize that these augmentations, introducing more significant changes to the point cloud’s spatial structure, result in more distinct feature signals for the AVR. Consequently, our decoupling mechanism can more effectively separate these signals under the orthogonality constraint. This process forces the model to learn a deeper understanding of the object’s core identity by successfully disentangling it from these drastic variations.

### 3D object semantic segmentation

Semantic segmentation is a challenging task that aims to assign a semantic label to each point in a point cloud, enabling the grouping of regions with meaningful significance. This task is particularly important in complex indoor and outdoor scenes, which are often characterized by

substantial background noise. To evaluate the representational capacity and generalization capability of our model, we conducted semantic segmentation experiments on the Stanford Large-Scale 3D Indoor Spaces (S3DIS) dataset. S3DIS is a widely used 3D indoor scene dataset that comprises scanned data from 272 rooms across 6 zones, covering a total area of approximately 6000 square meters. The dataset defines 13 semantic categories and provides fine-grained, point-wise semantic labels, where each point is annotated with comprehensive 9-dimensional feature information, including spatial coordinates (XYZ), color attributes (RGB), and normalized positional coordinates.

In this study, DGCNN is utilized as the backbone network, and the pre-trained model is transferred to the 3D semantic segmentation task on the S3DIS dataset for evaluation. The experimental setup follows the approach of Qi<sup>5</sup> et al. and Wang<sup>7</sup> et al., where each room is divided into 1 m × 1 m blocks. During the fine-tuning phase, the pre-trained model is trained for 100 epochs on each region of the S3DIS dataset, as outlined in<sup>5</sup>. The optimizer employed is SGD with an initial learning rate of 0.1, and the learning rate is adjusted using a cosine decay strategy. Additionally, 4096 points are randomly sampled from each block for training throughout the fine-tuning process.

We further focus on evaluating the semi-supervised learning performance of the proposed method in scenarios with limited labeled data. Specifically, the experiment fine-tunes the pre-trained model on a single region from regions 1 to 5 and evaluates its performance on region 6. The experimental results, presented in Table 7, demonstrate that the pre-trained model outperforms the DGCNN model trained from scratch under all experimental settings. Notably, the performance improvement is particularly significant when the number of labeled samples is limited.

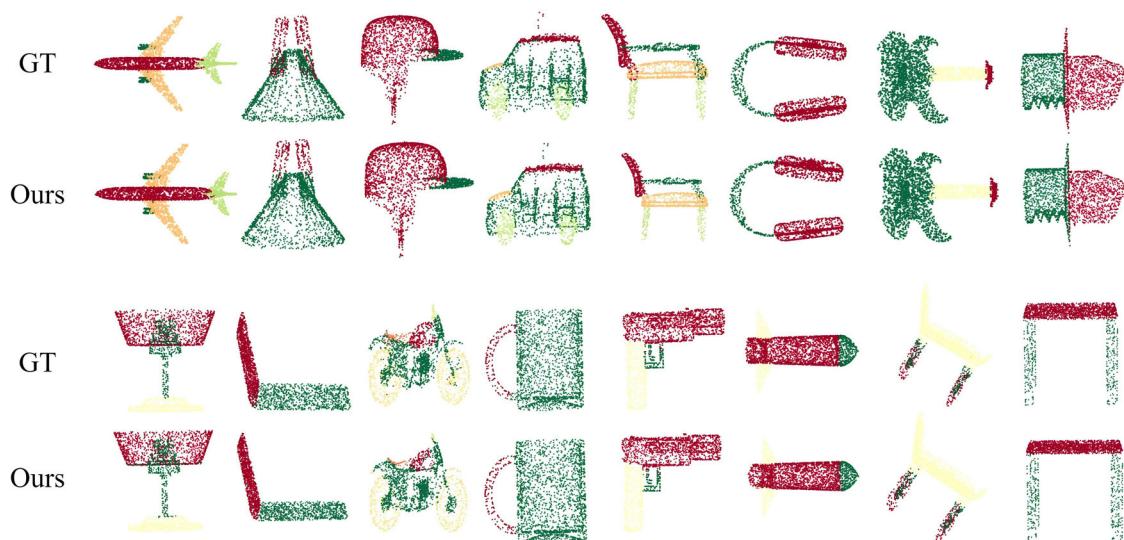
**Table 5 | Part segmentation results on the ShapeNetPart dataset**

Encoder	Method	mIoU
PointNet	Jigsaw3D <sup>50</sup>	82.2
	OcCo <sup>15</sup>	83.4
	CrossPoint <sup>24</sup>	82.7
	<b>PointDecoupler</b>	83.1
DGCNN	Jigsaw3D <sup>50</sup>	84.3
	OcCo <sup>15</sup>	85.0
	CrossPoint <sup>24</sup>	85.3
	<b>PointDecoupler</b>	85.4

PointDecoupler outperforms supervised learning methods with random initial weights and other SSL methods with pre-trained weights.

**Table 6 | Ablation study: data augmentation**

Data augmentation	Accuracy
Full	89.3%
Remove rotation	88.7%
Remove scaling	89.0%
Remove translation	88.5%
Remove crop	88.2%
Remove excavation	87.9%
Remove crop + excavation	87.7%



**Fig. 7 | Visualization of PointDecoupler’s segmentation results on the ShapeNetPart.**

**Ablations and analysis**

As described in the last part, to optimize representation learning, we integrate a decoupling module with a self-distillation module. The decoupling module enhances robustness to augmentation operations by learning their associated information, while the self-distillation module further optimizes the learning process of shallow representations by improving their linear separability and enhancing their synergy with high-level representations. To assess their individual and combined contributions, we conduct ablation studies across four settings: main path only, self-distillation only (excluding the AVR component from the decoupling loss and replacing it with the traditional forward-contrastive learning loss function), decoupling only (removing all components related to self-distillation and retaining only the decoupling loss), and full joint learning. Pre-training is performed with PointNet and DGCNN backbones, followed by SVM-based classification on ModelNet40 and ScanObjectNN.

Results (Fig. 8) show that the performance of using only the main path was significantly lower than when any additional module was incorporated. The combined framework consistently outperforms both modules used in isolation, validating the superiority of the joint learning strategy.

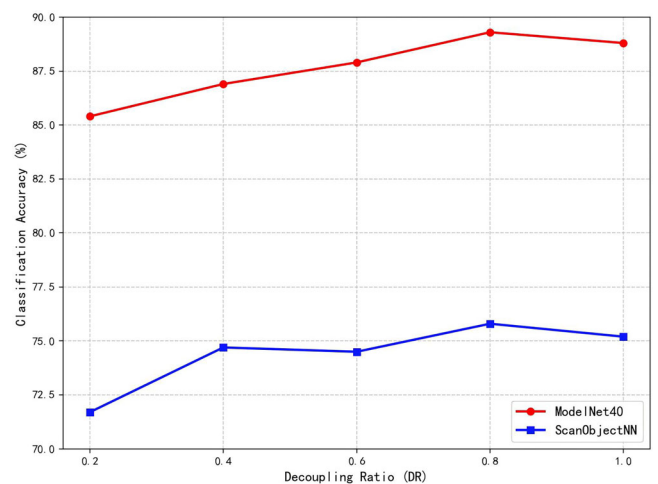
**Table 7 | PointDecoupler’s segmentation results in different regions of S3DIS using a fine-tuning strategy**

Area	Method	OA	mIoU
Area 1 (3687 samples)	From scratch	84.57	57.85
	PointDecoupler	85.96	59.97
Area 2 (44,407 samples)	From scratch	70.56	38.86
	PointDecoupler	71.45	39.67
Area 3 (16,507 samples)	From scratch	77.68	49.50
	PointDecoupler	79.16	51.88
Area 4 (36,627 samples)	From scratch	73.55	38.50
	PointDecoupler	74.51	41.60
Area 5 (68,527 samples)	From scratch	76.85	48.63
	PointDecoupler	78.95	51.10

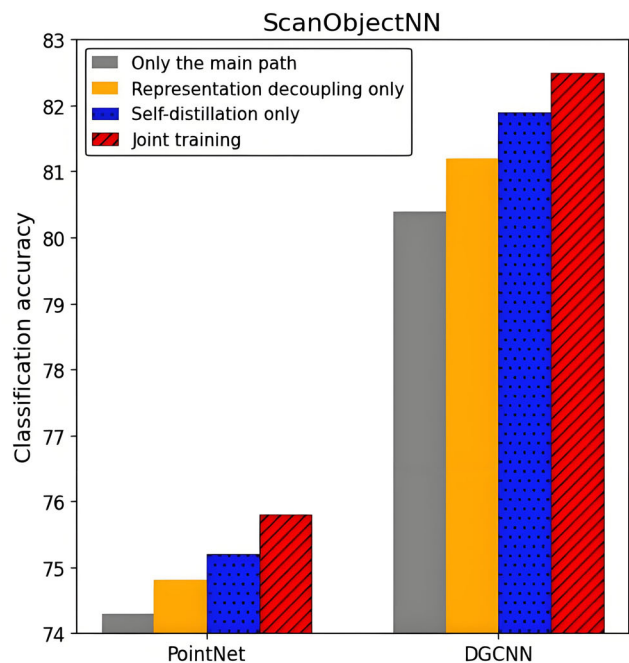
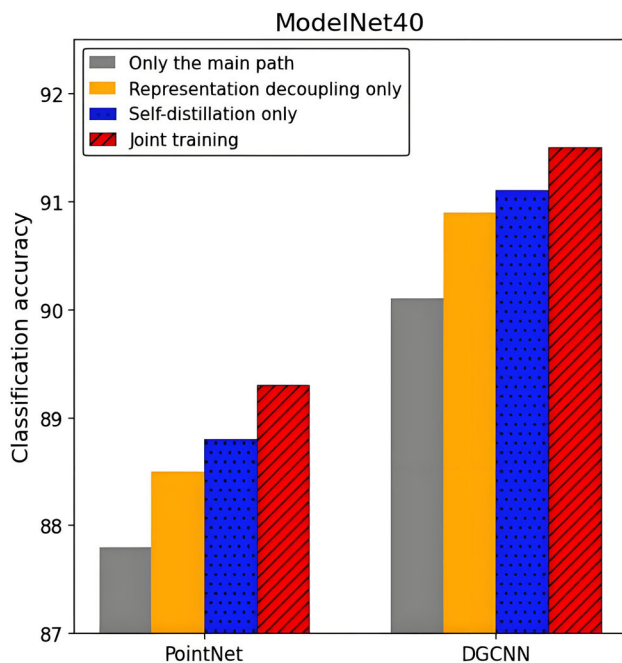
The DR is used to describe the proportion of AIR to AVR in the overall representation. This ratio plays a critical role in determining whether the decoupling loss can function effectively. Therefore, we further investigate the impact of different DR settings on the performance of downstream classification tasks. Specifically, we use PointNet as the point cloud feature encoder for pre-training under various DR settings and evaluate the SVM-based classification performance on the ModelNet40 and ScanObjectNN datasets.

As illustrated in Fig. 9, the overall classification performance of the model improves as the proportion of AIR gradually increases. However, when the proportion of AIR exceeds 0.8, the classification performance begins to decline. Based on these observations, DR is typically set to 0.8 during the pre-training phase to achieve optimal classification results.

During training, we applied a ratio annealing strategy for both Eqs. (10 and 12). Figure 10 shows the performance change across ranges of  $\alpha$  and  $\beta$ . For both parameters, the performance generally increases until their values reach 1. As discussed,  $\alpha$ , which controls  $L_{isd}$ , has a greater impact on the performance than  $\beta$ .



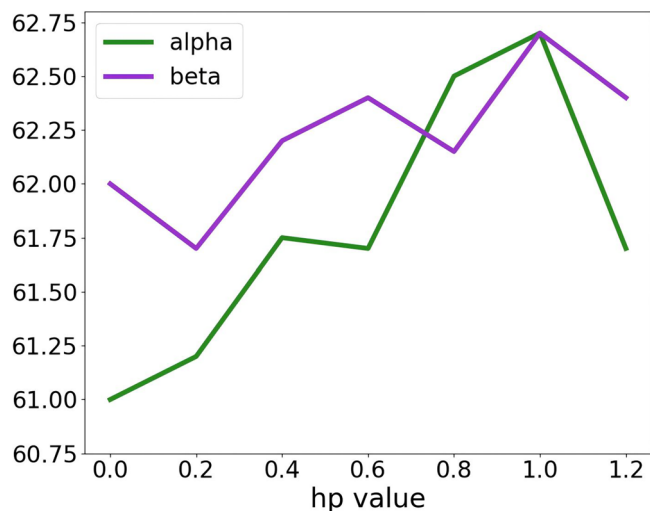
**Fig. 9 | Graph of sensitivity comparison experiment results of DR.**



**Fig. 8 | Ablation study of individual framework components.**

### Terracotta warriors dataset

The Terracotta Warriors, renowned as one of the Eight Wonders of the World, represent a significant ceramic cultural relic in China. Their virtual restoration holds great importance for cultural heritage preservation and transmission. This study focuses on the 3D digitization and processing of Terracotta Warrior fragments for neural network analysis. Our dataset was



**Fig. 10 | Hyperparameter sweeping.** We vary  $\alpha$  and  $\beta$  to see their effect. When sweeping for a parameter, the other parameter is fixed to 1.0.

acquired using a Creaform VIU 718 handheld 3D scanner in the Visualization Laboratory. Due to the high resolution of the resulting point clouds, which poses challenges for direct neural network input, we employed a preprocessing step. The Clustering Decimation method, available in the Meshlab tool, was utilized to downsample the point cloud data. This approach effectively preserves structural information while reducing each fragment to a uniform 2048 points. The dataset was categorized according to the anatomical parts of the Terracotta Warriors: arms, heads, legs, and bodies (as illustrated in Fig. 11). The sample distribution across these categories is presented in Table 8. For our experimental protocol, we adopted an 80–20 split, allocating 80% of the data for training and the remaining 20% for testing.

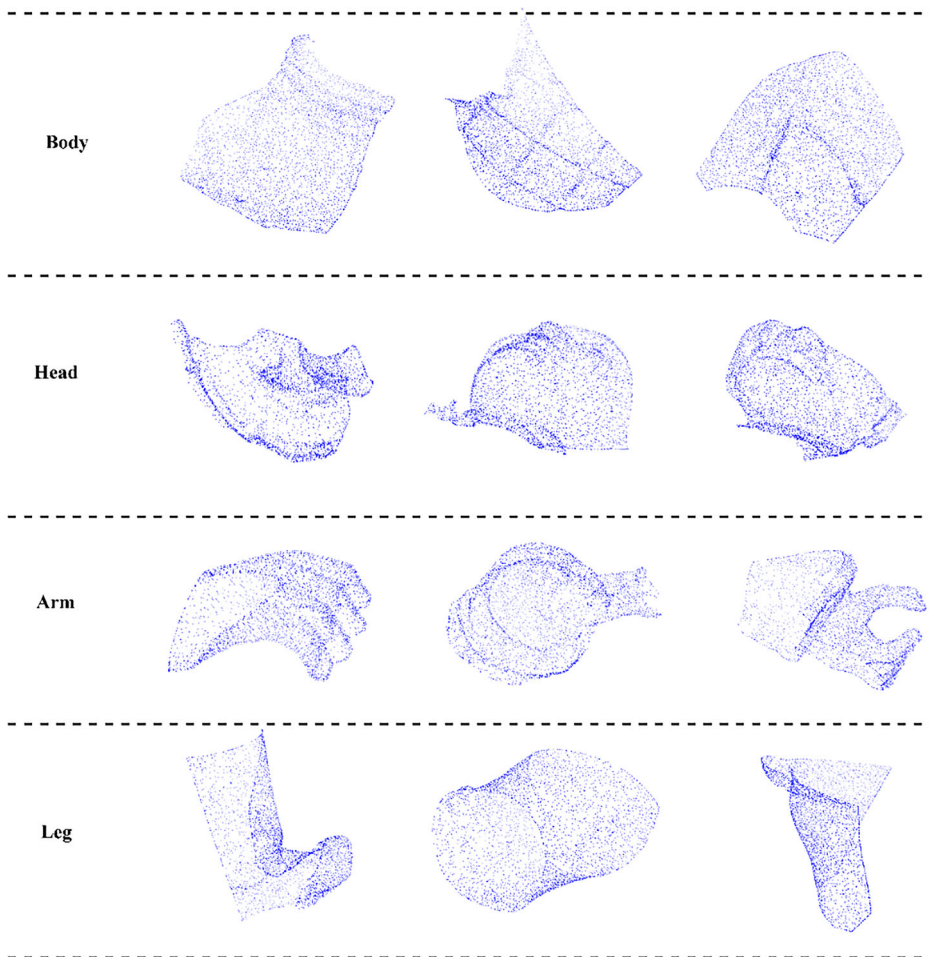
### Classification of terracotta warrior fragments

Accurate and efficient classification of cultural relic fragments is crucial for improving the efficiency and precision of cultural relic restoration, thereby providing robust technical support for cultural heritage professionals. To verify the effectiveness of the pre-trained model proposed in this study in terms of its representation capability, we apply the method to the Terracotta Warriors fragment dataset and conduct experimental evaluations.

**Table 8 | Number of fragments for each class in the terracotta warriors fragments dataset**

Label	Arm	Body	Head	Leg	Total
Train	4178	4738	2430	4274	15,620
Test	1045	1185	607	1068	3905

**Fig. 11 | Illustration of the terracotta warrior dataset** categorized by anatomical parts.



In the experiment, DGCNN is employed as the backbone network for fine-tuning and testing on the Terracotta Warriors fragment dataset. The results in Table 9 demonstrate that the proposed method significantly improves fragment classification, particularly excelling over supervised methods when labeled data is scarce.

### Segmentation of the terracotta warriors

Segmentation of Terracotta Warriors plays a crucial role in the effective and accurate restoration of cultural relics, particularly in the virtual reconstruction of ceramic artifacts. Unlike the Terracotta fragment classification task, our segmentation study utilizes complete Terracotta Warrior models. We compiled a dataset of 150 complete Terracotta models using 3D scanners and data augmentation techniques, and all terracotta warrior models

were uniformly downsampled into 4096 point clouds. Traditionally, the three-dimensional model of a Terracotta Warrior is divided into six parts: head, body, left arm, right arm, left leg, and right leg. However, to enhance the restoration process and rigorously evaluate our method’s performance, we manually annotated the original Terracotta models into eight distinct segments: head, body, left hand, left arm, right hand, right arm, left leg, and right leg. We employed an 8–2 split for our dataset, allocating 80% for training and 20% for testing.

We use a pre-trained DGCNN model to fine-tune the Terracotta Warriors segmentation dataset and evaluate its performance. The segmentation results are presented in Table 10. Compared with existing unsupervised segmentation methods specifically designed for the Terracotta Warriors, the proposed method achieves superior performance.

Figure 12 illustrates the visualized results of the segmentation task. It can be observed that the proposed method accurately identifies the eight main parts of the Terracotta Warriors and achieves precise segmentation of intricate details, such as the boundary between the hand and the arm. These results demonstrate that the proposed method not only excels in overall segmentation performance but also effectively captures fine-grained features of complex components. This further validates the effectiveness of the proposed method in practical application scenarios.

**Table 9 | Fragment classification accuracy compared with other methods on the 3D Terracotta Warrior Fragment Datasets**

Method	Supervised	Acc.
Method in <sup>2</sup>	√	87.64
PointNet <sup>5</sup>	√	88.93
Method in <sup>1</sup>	√	91.41
UMA-NET <sup>62</sup>	×	93.90
<b>PointDecoupler (Ours)</b>	×	<b>94.12</b>

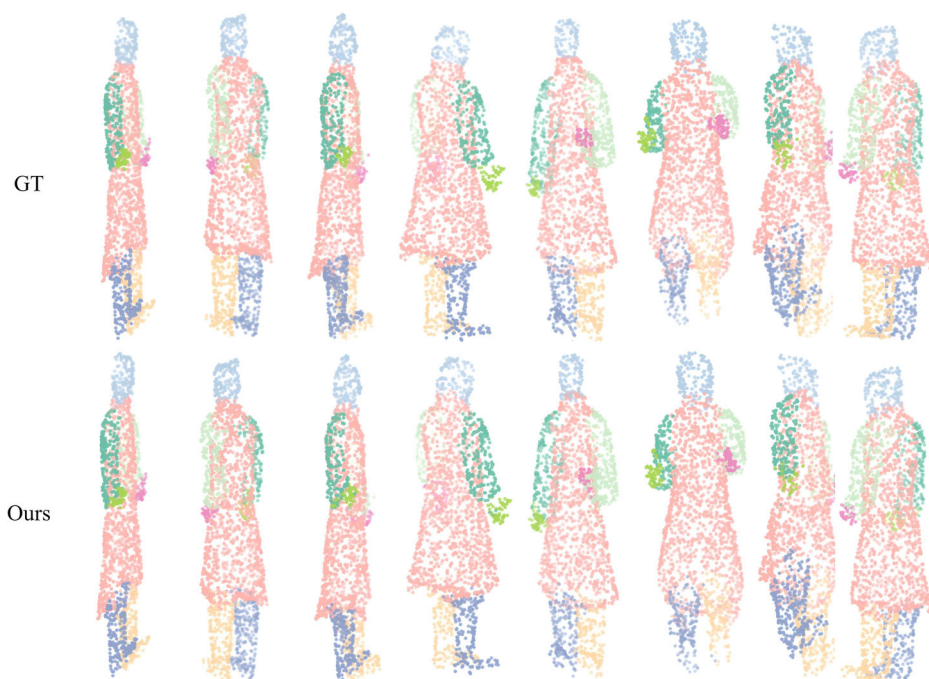
**Table 10 | Comparison of segmentation performance with different methods on the terracotta warriors dataset**

Method	mIoU
SRG <sup>63</sup> + DGCNN	65.6
SRG <sup>63</sup> + PointNet	54.4
EGG <sup>64</sup> + DGCNN	68.6
EGG <sup>64</sup> + PointNet	62.4
<b>PointDecoupler (ours)</b>	<b>74.3</b>

### Discussion

In this paper, we propose PointDecoupler, a novel SSL framework for point clouds, and demonstrate its effectiveness across various downstream tasks, including the restoration of the Terracotta Warriors. Unlike traditional contrastive methods that focus solely on semantic consistency via AIR, PointDecoupler explicitly models the interaction between AIR and AVR features through a decoupling loss, enhancing robustness to augmentation and sensitivity to geometric structures. It also employs self-distillation to guide low-level features using high-level representations, improving feature discrimination. While these achievements highlight the method’s strengths, we acknowledge the need for deeper investigation into the interpretability of learned AVR and their precise impact on model behavior. Specifically, due to the random combination of data augmentations, the current design cannot attribute the learned variant representations to individual augmentation operations, which makes a fine-grained analysis challenging. Future research directions include developing explainable analysis tools to

**Fig. 12 | Segmentation results of PointDecoupler on the terracotta warriors dataset.**



elucidate feature contribution mechanisms, exploring structured augmentation strategies to enable a clearer analysis of disentanglement, optimizing the architecture for enhanced computational efficiency, and extending the framework to multi-modal settings by incorporating complementary data modalities such as RGB imagery. The framework's potential extensions to broader 3D understanding tasks—including large-scale scene reconstruction and fine-grained object detection—promise to advance both theoretical foundations and practical applications in cultural heritage preservation.

### Data availability

Data underlying the results presented in this paper can be obtained from the internet. The Terracotta Warriors data will be available upon reasonable request.

### Code availability

The code used in the current research is available from the corresponding author upon request.

Received: 24 March 2025; Accepted: 21 September 2025;

Published online: 10 October 2025

### References

- Yang, K. et al. Classification of 3D terracotta warriors fragments based on geospatial and texture information. *J. Visualization*. **24**, 251–259 (2021).
- Du, G. et al. Classifying fragments of terracotta warriors using template-based partial matching. *Multimedia Tools Appl* **77**, 19171–19191 (2018).
- Rasheed, N. A. & Nordin, M. J. Classification and reconstruction algorithms for the archaeological fragments. *J. King Saud Univ.-Comput. Inf. Sci.* **32**, 883–894 (2020).
- Lin, X., Xue, B. & Wang, X. Digital 3D reconstruction of ancient chinese great wild goose pagoda by TLS point cloud hierarchical registration. *ACM J. Comput. Cult. Heritage*. **17**, 1–16 (2024).
- Qi, C. R., Su, H., Mo, K. & Guibas, L. J. Pointnet: deep learning on point sets for 3D classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 77–85 (IEEE, 2017).
- Qi, C. R., Yi, L., Su, H. & Guibas, L. J. Pointnet++: deep hierarchical feature learning on point sets in a metric space. In *Proc. 31st International Conference on Neural Information Processing Systems* 5105–5114 (Curran Associates Inc., 2017).
- Wang, Y. et al. Dynamic graph cnn for learning on point clouds. *ACM Transac. Graphics* **38**, 1–12 (2019).
- Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. A simple framework for contrastive learning of visual representations. In *Proc. 37th International Conference on Machine Learning (ICML)* 1597–1607 (PMLR, 2020).
- Yang, Y., Feng, C., Shen, Y. & Tian, D. Foldingnet: point cloud auto-encoder via deep grid deformation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 206–215 (IEEE, 2018).
- Han, Z., Wang, X., Liu, Y.-S. & Zwicker, M. Multi-angle point cloud-VAE: unsupervised feature learning for 3D point clouds from multiple angles by joint self-reconstruction and half-to-half prediction. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* 10441–10450 (IEEE, 2019).
- Zhao, Y., Birdal, T., Deng, H. & Tombari, F. 3D point capsule networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 1009–1018 (IEEE, 2019).
- Achlioptas, P., Diamanti, O., Mitliagkas, I. & Guibas, L. Learning representations and generative models for 3D point clouds. In *Proc. 35th International Conference on Machine Learning (ICML)* 40–49 (PMLR, 2018).
- Yu, X. et al. Point-BERT: pre-training 3D point cloud transformers with masked point modeling. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 19291–19300 (IEEE, 2022).
- Pang, Y. et al. Masked autoencoders for point cloud self-supervised learning. In *Proc. 17th European Conference on Computer Vision (ECCV)* 604–621 (Springer, 2022).
- Wang, H. et al. Unsupervised point cloud pre-training via occlusion completion. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* 9782–9792 (IEEE, 2021).
- Su, K. et al. RI-MAE: rotation-invariant masked autoencoders for self-supervised point cloud representation learning. In *Proc. 39th Annual AAAI Conference on Artificial Intelligence* 7015–7023 (AAAI Press, 2025).
- Fan, S., Gao, W. & Li, G. Point-MPP: point cloud self-supervised learning from masked position prediction. *IEEE Trans. Neural Networks Learn. Syst.* **36**, 12964–12976 (2024).
- Li, Y., Madarasingha, C. & Thilakarathna, K. DiffPMAE: diffusion masked autoencoders for point cloud reconstruction. In *Proc. 18th European Conference on Computer Vision (ECCV)* 362–380 (Springer, 2024).
- Zheng, X. et al. Point cloud pre-training with diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 22935–22945 (IEEE, 2024).
- Sun, J. et al. Diffusion-driven self-supervised learning for shape reconstruction and pose estimation. Preprint at <https://arxiv.org/abs/2403.12728> (2024).
- He, K. et al. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 9726–9735 (IEEE, 2020).
- Liu, Y.-C. et al. Learning from 2d: contrastive pixel-to-point knowledge transfer for 3D pretraining. Preprint at <https://arxiv.org/abs/2104.04687> (2021).
- Xie, S. et al. Pointcontrast: unsupervised pre-training for 3D point cloud understanding. In *Proc. 16th European Conference on Computer Vision (ECCV)* 574–591 (Springer, 2020).
- Afham, M. et al. Crosspoint: self-supervised cross-modal contrastive learning for 3D point cloud understanding. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 9892–9902 (IEEE, 2022).
- Huang, S., Xie, Y., Zhu, S.-C. & Zhu, Y. Spatio-temporal self-supervised representation learning for 3D point clouds. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* 6515–6525 (IEEE, 2021).
- Grill, J.-B. et al. Bootstrap your own latent—a new approach to self-supervised learning. In *Proc. 34th International Conference on Neural Information Processing Systems* 21271–21284 (Curran Associates Inc., 2020).
- Du, B. A., Gao, X., Hu, W. & Li, X. Self-contrastive learning with hard negative sampling for self-supervised point cloud learning. In *Proc. 29th ACM International Conference on Multimedia* 3133–3142 (Association for Computing Machinery, 2021).
- Zhang, L. & Zhu, Z. Unsupervised feature learning for point cloud understanding by contrasting and clustering using graph convolutional neural networks. In *Proc. 7th International Conference on 3D Vision* 395–404 (IEEE, 2019).
- Kingma, D. P. & Welling, M. Auto-encoding variational bayes. Preprint at <https://arxiv.org/abs/1312.6114> (2013).
- Kumar, A., Sattigeri, P. & Balakrishnan, A. Variational inference of disentangled latent concepts from unlabeled observations. Preprint at <https://arxiv.org/abs/1711.00848> (2017).
- Chen, X. et al. InfoGAN: interpretable representation learning by information maximizing generative adversarial nets. In *Proc. 30th International Conference on Neural Information Processing Systems* 2180–2188 (Curran Associates Inc., 2016).

32. Yang, T., Wang, Y., Lv, Y. & Zheng, N. Disdiff: unsupervised disentanglement of diffusion probabilistic models. Preprint at <https://arxiv.org/abs/2301.13721> (2023).
33. Liang, P. P. et al. Factorized Contrastive Learning: going beyond multi-view redundancy. In *Proc. 37th International Conference on Neural Information Processing Systems* 32971–32998 (Curran Associates Inc., 2023).
34. Wang, Y. et al. Decoupling common and unique representations for multimodal self-supervised learning. In *Proc. 18th European Conference on Computer Vision (ECCV)* 286–303 (Springer, 2024).
35. Tran, L., Yin, X. & Liu, X. Disentangled representation learning gan for pose-invariant face recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 1283–1292 (IEEE, 2017).
36. Liu, L. et al. Activity image-to-video retrieval by disentangling appearance and motion. In *Proc. 35th Annual AAAI Conference on Artificial Intelligence* 2145–2153 (AAAI Press, 2021).
37. Li, Z., Murkute, J. V., Gyawali, P. K. & Wang, L. Progressive learning and disentanglement of hierarchical representations. Preprint at <https://arxiv.org/abs/2002.10549> (2020).
38. Ross, A. & Doshi-Velez, F. Benchmarks, algorithms, and metrics for hierarchical disentanglement. In *Proc. 38th International Conference on Machine Learning (ICML)* 9084–9094 (PMLR, 2021).
39. Xiang, S. et al. Disunknown: Distilling unknown factors for disentanglement learning. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* 14790–14799 (IEEE, 2021).
40. Bouchacourt, D., Tomioka, R. & Nowozin, S. Multi-level variational autoencoder: learning disentangled representations from grouped observations. In *Proc. 32th Annual AAAI Conference on Artificial Intelligence* (AAAI Press, 2018).
41. Kulkarni, T. D., Whitney, W. F., Kohli, P. & Tenenbaum, J. Deep convolutional inverse graphics network. In *Proc. 29th International Conference on Neural Information Processing Systems* 2539–2547 (MIT Press, 2015).
42. Xiao, T., Hong, J. & Ma, J. DNA-GAN: learning disentangled representations from multi-attribute images. Preprint at <https://arxiv.org/abs/1711.05415> (2017).
43. Yang, M. et al. Causalvae: Disentangled representation learning via neural structural causal models. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 9588–9597 (IEEE, 2021).
44. Shen, X. et al. Disentangled generative causal representation learning. *The Ninth International Conference on Learning Representations*. <https://openreview.net/forum?id=agyFqcmgl6y> (ICLR, 2020)
45. Hinton, G., Vinyals, O. & Dean, J. Distilling the knowledge in a neural network. Preprint at *arXiv* <https://arxiv.org/abs/1503.02531> (2015).
46. Li, T., Wang, L. & Wu, G. Self supervision to distillation for long-tailed visual recognition. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* 610–619 (IEEE, 2021).
47. Xu, G., Liu, Z., Li, X. & Loy, C. C. Knowledge distillation meets self-supervision. In *Proc. 16th European Conference on Computer Vision (ECCV)* 588–604 (Springer, 2020).
48. Wang, J., Song, S., Su, J. & Zhou, S. K. Distortion-disentangled contrastive learning. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* 75–85 (IEEE, 2024).
49. Dangovski, R. et al. Equivariant contrastive learning. Preprint at *arXiv* <https://arxiv.org/abs/2111.00899> (2021).
50. Sauder, J. & Sievers, B. Self-supervised deep learning on point clouds by reconstructing space. In *Proc. 33rd International Conference on Neural Information Processing Systems* (Curran Associates Inc., 2019).
51. Wu, J. et al. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *Proc. 30th International Conference on Neural Information Processing Systems* (Curran Associates Inc., 2016).
52. Li, J., Chen, B. M. & Lee, G. H. So-net: Self-organizing network for point cloud analysis. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 9397–9406 (IEEE, 2018).
53. Kanezaki, A., Matsushita, Y. & Nishida, Y. Rotationnet: joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 5010–5019 (IEEE, 2018).
54. Gadelha, M., Wang, R. & Maji, S. Multiresolution tree networks for 3D point cloud processing. In *Proc. 15th European Conference on Computer Vision (ECCV)* 102–122 (Springer, 2018).
55. Han, Z., Shang, M., Liu, Y.-S. & Zwicker, M. View inter-prediction GAN: unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions. In *Proc. 33th Annual AAAI Conference on Artificial Intelligence* 8376–8384 (AAAI Press, 2019).
56. Zhang, Z., Girdhar, R., Joulin, A. & Misra, I. Self-supervised pretraining of 3D features on any point-cloud. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* 10232–10243 (IEEE, 2021).
57. Chen, C. et al. Clusternet: Deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 4989–4997 (IEEE, 2019).
58. Poursaeed, O. et al. Self-supervised learning of point clouds via orientation estimation. In *Proc. 8th International Conference on 3D Vision (3DV)* 1018–1028 (IEEE, 2020).
59. Li, Y. et al. Pointcnn: convolution on x-transformed points. In *Proc. 32nd International Conference on Neural Information Processing Systems* 828–838 (Curran Associates Inc., 2018).
60. Zhang, Z., Hua, B.-S. & Yeung, S.-K. Shellnet: efficient point cloud convolutional neural networks using concentric shells statistics. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* 1607–1616 (IEEE, 2019).
61. Zhao, H. et al. Point transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* 16239–16248 (IEEE, Montreal, 2021).
62. Liu, J. et al. UMA-Net: an unsupervised representation learning network for 3D point cloud classification. *J. Optical Soc. Am. A* **39**, 1085–1094 (2022).
63. Hu, Y. et al. Unsupervised segmentation for terracotta warrior with seed-region-growing CNN (SRG-Net). In *Proc. 5th International Conference on Computer Science and Application Engineering* 1–6 (Association for Computing Machinery, Sanya, 2021).
64. Hu, Y. et al. Self-supervised segmentation for terracotta warrior point cloud (EGG-Net). *IEEE Access* **10**, 12374–12384 (2022).

## Acknowledgements

We would like to acknowledge Dr. Chuang Niu from Rensselaer Polytechnic Institute for providing technical guidance and expertise that greatly assisted our research. We thank Emperor Qinshihuang's Mausoleum Site Museum for providing the Terracotta Warriors data. This work was supported in part by the Project Supported by National Key R&D Program of China No. 2024YFF0907604, National Natural Science Foundation of China No. 62572394, Key Research and Development Program of Shaanxi Province (2019GY-215, 2021ZDLSF06-04, and 2024SF-YBXM-681).

## Author contributions

Conceptualization: X.C. and L.Y. Methodology: X.C. and Xingxing.H. Software: J.Z. Validation: Xinxin.H. Formal analysis: L.S. Investigation: X.C. Resources: K.L. Data curation: K.L. Writing—original draft preparation: X.C.

Writing—review and editing: X.C. Supervision: K.L. Project administration: K.L. Funding acquisition: K.L. All authors have read and agreed to the published version of the manuscript.

### Competing interests

The authors declare no competing interests.

### Additional information

**Correspondence** and requests for materials should be addressed to Kang Li.

**Reprints and permissions information** is available at <http://www.nature.com/reprints>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025