A Unified Approach to Interpreting Self-supervised Pre-training Methods for 3D Point Clouds via Interactions

Qiang Li, Jian Ruan, Fanghao Wu, Yuchi Chen, Zhihua Wei*, and Wen Shen*
Tongji University, Shanghai, China
{qli,jianruan,2432055,1953721,zhihua_wei,wenshen}@tongji.edu.cn

Abstract

Recently, many self-supervised pre-training methods have been proposed to improve the performance of deep neural networks (DNNs) for 3D point clouds processing. However, the common mechanism underlying the effectiveness of different pre-training methods remains unclear. In this paper, we use game-theoretic interactions as a unified approach to explore the common mechanism of pre-training methods. Specifically, we decompose the output score of a DNN into the sum of numerous effects of interactions, with each interaction representing a distinct 3D substructure of the input point cloud. Based on the decomposed interactions, we draw the following conclusions. (1) The common mechanism across different pre-training methods is that they enhance the strength of high-order interactions encoded by DNNs, which represent complex and global 3D structures, while reducing the strength of low-order interactions, which represent simple and local 3D structures. (2) Sufficient pre-training and adequate fine-tuning data for downstream tasks further reinforce the mechanism described above. (3) Pre-training methods carry a potential risk of reducing the transferability of features encoded by DNNs. Inspired by the observed common mechanism, we propose a new method to directly enhance the strength of high-order interactions and reduce the strength of low-order interactions encoded by DNNs, improving performance without the need for pre-training on large-scale datasets. Experiments show that our method achieves performance comparable to traditional pre-training methods.

1. Introduction

Self-supervised pre-training methods for 3D point clouds have developed rapidly in recent years [1, 9, 13, 16, 21, 23, 30, 32, 38, 43]. Pre-training methods first train DNNs on large-scale unlabeled datasets, then fine-tune the DNNs on downstream tasks, generally enhancing their performance. However, the common mechanism underlying different pre-training methods remains unclear, posing challenges for gaining insights into effective model training strategies.

In this paper, we aim to explore the common mechanism behind the performance improvements of different pre-training methods, thereby providing insights into pre-training, and offering better guidance for the training process. Recent studies have employed interactions to explain the reasoning processes of DNNs [20, 25, 41]. Inspired by these studies, we use interactions to provide a unified interpretation of different pre-training methods.

Specifically, given a point cloud x with n regions[1] indexed by $N = \{1, 2, \ldots, n\}$, an interaction represents the collaborations among regions within a specific 3D structure $S \subseteq N$, where each interaction has a numerical effect $I(S)$ on the network output. For example, as shown in Fig. 1, the interaction between the regions in $S_1 = \{\textit{wingtip}, \textit{wing root}\}$ form a concept of "*wing*", contributing $I(S_1)$ to push the classification result toward the class "*airplane*". It has been proven by [6, 44] that the output score of a DNN consistently equals the sum of the effects of all activated interactions, regardless of how the input regions are masked. In this way, interactions can be seen as the detailed inference patterns encoded by the DNN.

Based on interactions, we conduct comparative experiments to explore the common reasons behind the performance improvements of different pre-training methods. Specifically, we explore the impact of pre-training methods on the complexity of interactions encoded by DNNs. Here, the complexity refers to the number of regions contained in an interaction, *i.e.*, the order of the interaction. A high-order interaction, *e.g.*, S_3 in Fig. 1, captures collaborations among massive point regions, representing complex and global 3D structures. In contrast, a low-order interaction, *e.g.*, S_1, measures collaborations between a few regions, representing simple and local 3D structures. From the experiments, we draw the following key conclusions.

- **The common mechanism across different pre-training**

*Corresponding authors: Wen Shen and Zhihua Wei.

[1] We divide a point cloud into n regions following [26].

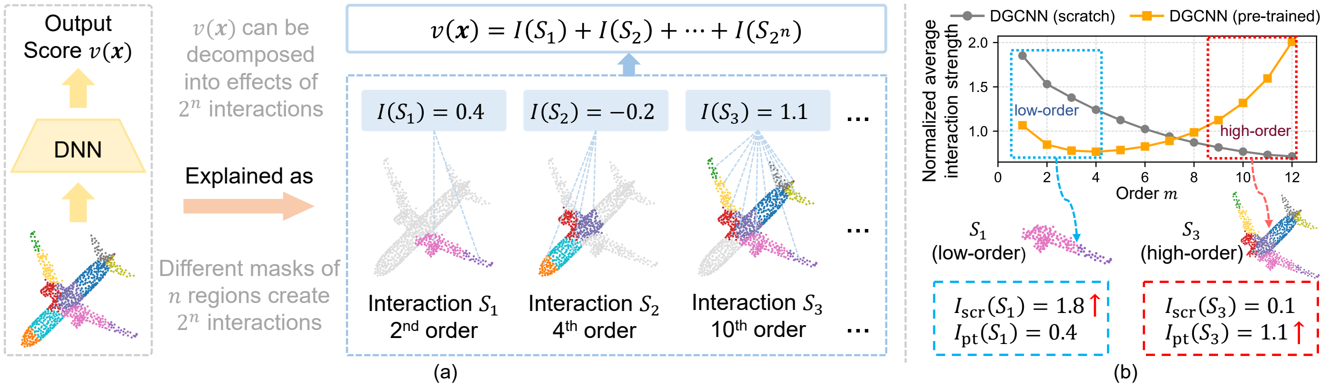
Output Score $v(x)$
DNN
$v(x)$ can be decomposed into effects of 2^n interactions
Explained as
Different masks of n regions create 2^n interactions
$v(\boldsymbol{x}) = I(S_1) + I(S_2) + \cdots + I(S_{2^n})$
$I(S_1) = 0.4$
$I(S_2) = -0.2$
$I(S_3) = 1.1$
...
Interaction S_1 2nd order
Interaction S_2 4th order
Interaction S_3 10th order
(a)
DGCNN (scratch)
DGCNN (pre-trained)
Normalized average interaction strength
2.0
1.5
1.0
0
2
4
6
8
10
12
Order m
low-order
high-order
S_1 (low-order)
S_3 (high-order)
$I_{\text{scr}}(S_1) = 1.8$↑
$I_{\text{pt}}(S_1) = 0.4$
$I_{\text{scr}}(S_3) = 0.1$
$I_{\text{pt}}(S_3) = 1.1$↑
(b)

Figure 1. (a) Illustration of how interactions can be used to explain a DNN. Given an input point cloud with n regions, the output score of the DNN can be decomposed into the sum of the numerical effects of 2^n interactions, where each interaction S encodes the collaborations among the point cloud regions in the set S. (b) Comparing the strength of interactions across different orders encoded by the DGCNN trained from scratch (scr) and the DGCNN using a pre-training method (pt). Results show that the pre-trained DGCNN encodes stronger high-order interactions and weaker low-order interactions than the DGCNN trained from scratch.

methods is that they enhance the strength of high-order interactions encoded by DNNs while reducing the strength of low-order interactions. This common mechanism indicates that pre-training methods enhance the DNNs' ability to capture global 3D structures, while reducing their reliance on local 3D structures.

- **Sufficient pre-training and adequate fine-tuning data for downstream tasks further reinforce the mechanism described above.** We observe that the strength of high-order interactions increases with the number of pre-training epochs while the strength of low-order interactions decreases. Additionally, increasing the amount of data for downstream tasks also amplifies this effect.
- **Pre-training methods carry a potential risk of reducing the transferability of features encoded by DNNs.** We observe that the performance of the pre-trained DNNs may decrease on unseen test datasets, possibly due to pre-training methods causing the DNNs to encode high-order interactions with excessively high strength.

Building on the common mechanism we identified, we propose a new method to directly enhance the strength of high-order interactions encoded by DNNs while reducing the strength of low-order interactions. Experimental results on classification and semantic segmentation benchmarks show that our method achieves performance comparable to pre-training methods, **without the need for pre-training on large-scale unlabeled datasets**.

2. Related work

Self-supervised learning (SSL) of 3D point clouds. Recently, 3D point cloud processing has developed rapidly [5, 11, 14, 15, 24, 27, 35, 37, 40], with many self-supervised methods proposed to learn representations from individual 3D objects [1, 9, 13, 16, 21, 23, 30, 32, 38, 43]. The goal of SSL is to design pretext tasks to help the model learn the data distribution and features in advance, preparing it for downstream tasks. In this paper, we explore the common mechanism behind the performance improvement of the following five widely used open-source pre-training methods.

- *Occlusion Completion (OcCo)* [32]. OcCo masks occluded points from a camera view and trains an encoder-decoder model to reconstruct these missing points.
- *Jigsaw* [21]. Jigsaw trains a model to reconstruct point clouds with parts rearranged in random order.
- *Implicit Auto-encoder (IAE)* [38]. IAE trains the model as an encoder to map the point clouds to a high-dimensional space and uses a decoder to reconstruct the encoder's outputs back into 3D geometry.
- *Spatio-Temporal Representation Learning (STRL)* [9]. STRL captures spatio-temporal information from 3D sequences by using two temporally correlated frames to learn invariant representations.
- *CrossPoint* [1]. CrossPoint learns transferable representations by maximizing the agreement between 3D point clouds and corresponding 2D images.

Using game-theoretical interactions to explain DNNs. Game-theoretical interactions provide a solid theoretical foundation for explaining DNNs. Ren *et al.* [17] proposed a mathematical formulation for the concepts encoded by a DNN, while Ren *et al.* [18] further leveraged these concepts to define optimal baseline values for Shapley values. Li *et al.* [10] provided a theoretical guarantee that interactions accurately capture the true concepts encoded by a DNN. At the application level, interactions have been widely used to explain the representation capacity of DNNs from various perspectives, including adversarial robustness [17, 34], adversarial transferability [33], and generalization power [41, 45]. In this paper, we use interactions to investigate the common mechanism underlying different pre-training methods for 3D point clouds.

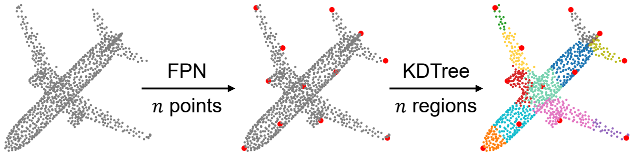

Figure 2. Process of dividing an input point cloud into n regions.

3. Interactions in 3D point cloud processing

Preliminaries: interactions. As a new explanatory metric, interaction has been used to clarify the inference logic [7], generalization power [34], and robustness of a DNN [45]. It can be viewed as a universal measure due to its close theoretical connections with other metrics. As proven by [19], the Harsanyi interaction serves as the basis for existing game-theoretic attributions and interactions, including the Shapley value [22], the Shapley interaction index [8], and the Shapley Taylor interaction index [28]. Please see the supplementary material for additional details.

Quantifying interactions for 3D point cloud processing. We extend interactions to 3D point clouds. Considering an point cloud $x \in \mathbb{R}^{P\times 3}$, we divide it into n regions, as shown in Fig. 2. First, we apply the farthest point sampling (FPS) algorithm to select n points from the point cloud as the centers of each region. Then, we use the k-dimensional tree (KDTree) algorithm to assign the remaining points to their nearest region. By doing so, we divide the input point cloud x into n regions, indexed by $N = \{1, 2, ..., n\}$.

Given a trained DNN $v : \mathbb{R}^{P\times 3} \to \mathbb{R}$, we follow [10, 20, 25] to define the DNN's output score as $v(x) = \log \frac{p}{1-p}$ to represent the classification confidence, where p is the output probability of the ground truth class. Then, the output score can be rewritten as the sum of the numerical effects of all 2^n interactions between the point regions, as follows.

$$v(x) = \sum_{S\subseteq N} I(S). \tag{1}$$

Here, $I(S)$ represents the numerical effect of the interaction among the point regions in $S \subseteq N$, defined as follows.

$$I(S) \triangleq \sum_{T\subseteq S} (-1)^{|S|-|T|} \cdot v(x_T), \tag{2}$$

where x_T represents the input point cloud with the regions in $T \subseteq N$ unchanged, while the regions in $N\setminus T$ are masked by replacing them with the centroid of the point cloud.

Understanding interactions in 3D point cloud processing. The interaction extracted from the input point cloud x encodes an **AND** relationship among the point regions in S, with the numerical effect $I(S)$ representing the combined contribution of these regions to the output score $v(x)$. As shown in Fig. 1, when the point regions in the set $S_1 = \{\textit{wingtip}, \textit{wing root}\}$ are unmasked, they form a "*wing*" pattern and contribute a numerical effect $I(S_1)$ that pushes the output score $v(x)$ towards the "*airplane*" category. Masking any region in S_1 will deactivate this AND interaction and remove $I(S_1)$ from $v(x)$. In fact, Tang *et al.* [29] has proven that interaction satisfies the universal matching property, which states that **the DNN's inference score $v(x_T)$ can always be faithfully explained as the sum of the numerical effects of all activated interactions, regardless of how the point cloud regions are masked**.

Theorem 1 (Universal matching property, proven by [29]). *Given an input sample x with n variables indexed by $N = \{1, 2, ..., n\}$, we can generate 2^n masked samples x_T where $T \subseteq N$. Let us construct the following surrogate logical model $\phi(\cdot)$ to use interactions for inference, which are extracted from the DNN $v(\cdot)$ on the sample x. Then, the output of the surrogate logical model $\phi(\cdot)$ can always match the output of the DNN $v(\cdot)$, regardless of how the input sample is masked.*

$$\begin{aligned} &\forall\, T \subseteq N,\ \phi(x_T) = v(x_T), \\ &\qquad \phi(x_T) = v(x_\emptyset) + \sum_{S\subseteq N} I(S)\cdot \mathbb{1}\begin{pmatrix} x_T \text{ triggers} \\ \textbf{AND} \text{ relation } S \end{pmatrix} \\ &\qquad\qquad\;\; = v(x_\emptyset) + \sum\nolimits_{\emptyset\neq S\subseteq T} I(S). \end{aligned} \tag{3}$$

Defining and quantifying the representation complexity of DNNs. The order of an interaction is defined as $m = |S|$, which reflects the representation complexity of DNNs. High-order interactions measure the effects of collaborations among massive point cloud regions, representing global and complex 3D structures, while low-order interactions measure the effects of collaborations between a few point regions, representing simple and local 3D structures. We introduce a new metric for measuring the representation complexity of DNNs, as follows.

$$\kappa^{(m)} \triangleq \frac{\mathbb{E}_x \mathbb{E}_{S\subseteq N, |S|=m}\left[|I(S)|\right]}{Z}, \tag{4}$$

where $\mathbb{E}$ denotes the mathematical expectation, and $Z = \mathbb{E}_x \mathbb{E}_{S\subseteq N}\left[|I(S)|\right]$ is a normalization term to ensure fair comparisons across different DNNs. Here, $\kappa^{(m)}$ measures the normalized average strength of the m-th order interactions. If the value of $\kappa^{(m)}$ of a high-order is larger than that of a low-order, the DNN's representation complexity is enough to capture global and complex 3D structures. Otherwise, the DNN's representation complexity is limited to encoding only local and simple 3D structures.

We further propose the following metrics to measure the strength of high-order interactions and low-order interactions encoded by the DNN.

$$\begin{aligned} \kappa^{\text{high}} &= \sum_{m\in\Omega^{\text{high}}} \kappa^{(m)},\ s.t.\ \Omega^{\text{high}} \overset{\text{def}}{=} \{m \mid \lceil \tfrac{2}{3} n \rceil < m \leq n\}, \\ \kappa^{\text{low}} &= \sum_{m\in\Omega^{\text{low}}} \kappa^{(m)},\ s.t.\ \Omega^{\text{low}} \overset{\text{def}}{=} \{m \mid 1 \leq m \leq \lceil \tfrac{1}{3} n \rceil\}, \end{aligned} \tag{5}$$

where Ω^{high} and Ω^{low} denote the ranges of high-order and low-order interactions, respectively.

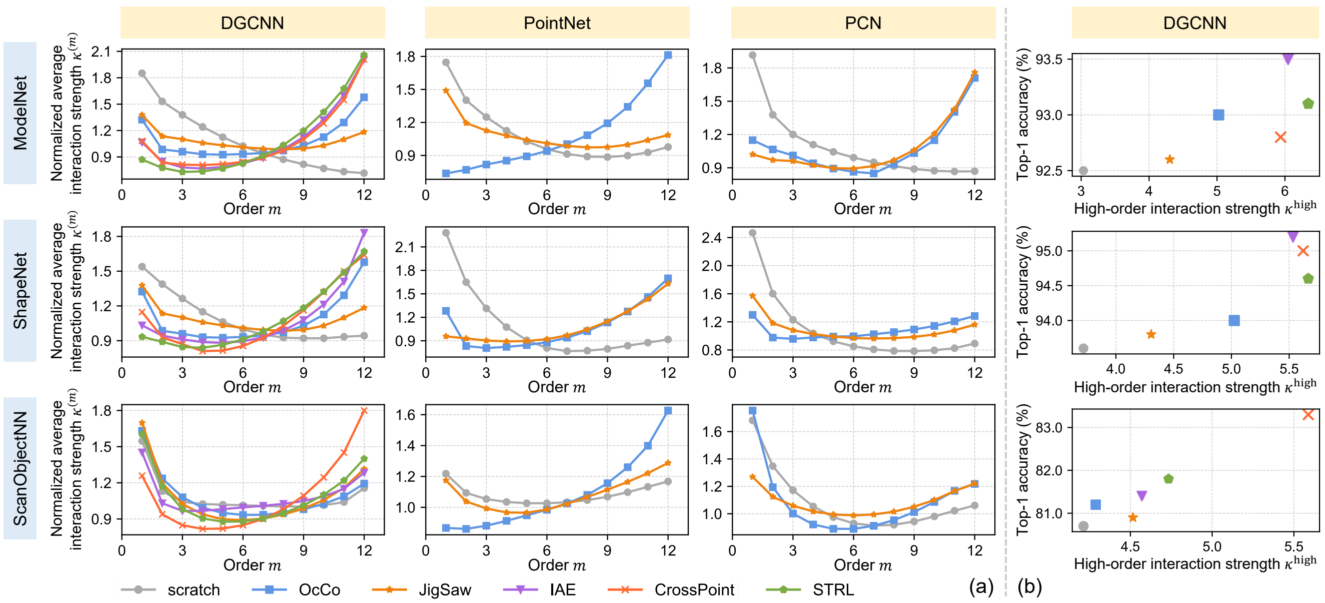

Figure 3. [Conclusion 1] (a) Comparing the normalized average strength of interactions encoded by different DNNs, including DNNs trained from scratch and DNNs trained with different pre-training methods. Results show that the DNNs using pre-training methods consistently encode stronger high-order interactions and weaker low-order interactions than the DNNs trained from scratch. (b) The relationship between the strength of high-order interactions encoded by different DNNs and their corresponding classification accuracy. Results show that DNNs encoding stronger high-order interactions tend to exhibit higher accuracy.

4. Interpreting different pre-training methods using interactions

4.1. Comparative study setup

For a given network architecture, we compare the interactions encoded by the model trained from scratch with those encoded by models trained using various pre-training methods. This comparison aims to explore whether these pre-training methods share a common underlying reason for performance improvement, which we define as the common mechanism across these methods. To provide a unified explanation for most pre-training methods, we conduct experiments on five widely used open-source pre-training methods, including IAE [38], STRL [9], CrossPoint [1], OcCo [32] and JigSaw [21], as detailed in Sec. 2.

Networks and datasets. We conduct experiments on three network architectures: DGCNN [35], PointNet [14], and PCN [40]. For DGCNN, we utilize all five pre-training methods, while for PointNet and PCN, we focus on OcCo [32] and Jigsaw [21], depending on the accessibility of open-source implementations for each pre-training method.

To compare the interactions encoded by different DNNs, we use three benchmark datasets for 3D classification task: ModelNet40 [36], ShapeNet[2] [3], and ScanObjectNN [31]. Tab. 1 shows the statistics of these datasets. We randomly select 10 samples per class from each dataset and use the method in Sec. 3 to divide each point cloud sample into n regions for quantifying the interactions encoded by DNNs.

Name	Type	# Class	# Training / Testing
ModelNet	synthesized	40	9,843 / 2,468
ShapeNet	synthesized	16	12,137 / 4,744
ScanObjectNN	real world	15	2,304 / 576

Table 1. Statistics of datasets for classification.

4.2. Exploring the common mechanism of different pre-training methods

Conclusion 1. The common mechanism across different pre-training methods is that they enhance the strength of high-order interactions encoded by DNNs, while reducing the strength of low-order interactions.

Fig. 3 (a) shows the normalized average strength of the interactions encoded by different DNNs, including DNNs trained from scratch and DNNs using different pre-training methods. Results show that the strength of high-order interactions encoded by the DNNs using pre-training methods is consistently greater than that of the DNNs trained from scratch, across all datasets and network architectures. Conversely, the DNNs using pre-training methods typically encode weaker low-order interactions than the DNNs trained from scratch. Fig. 3 (b) further illustrates the relationship between the strength of high-order interactions and the clas-

[2] The ShapeNet dataset for classification is derived from the ShapeNet part segmentation dataset, following [26].

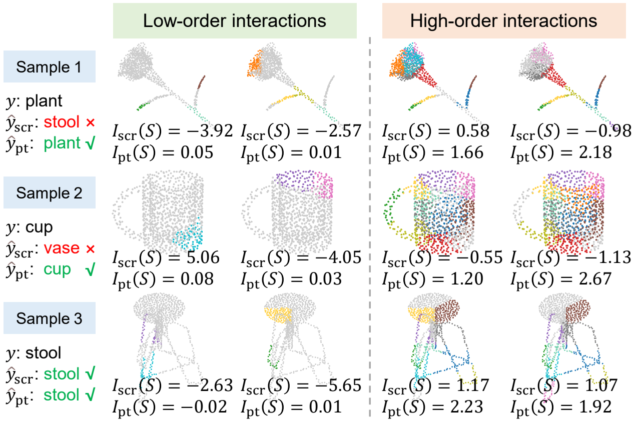
Low-order interactions
High-order interactions
Sample 1
y: plant
$\hat{y}_{\text{scr}}$: stool ×
$\hat{y}_{\text{pt}}$: plant √
$I_{\text{scr}}(S) = -3.92$ $I_{\text{pt}}(S) = 0.05$
$I_{\text{scr}}(S) = -2.57$ $I_{\text{pt}}(S) = 0.01$
$I_{\text{scr}}(S) = 0.58$ $I_{\text{pt}}(S) = 1.66$
$I_{\text{scr}}(S) = -0.98$ $I_{\text{pt}}(S) = 2.18$
Sample 2
y: cup
$\hat{y}_{\text{scr}}$: vase ×
$\hat{y}_{\text{pt}}$: cup √
$I_{\text{scr}}(S) = 5.06$ $I_{\text{pt}}(S) = 0.08$
$I_{\text{scr}}(S) = -4.05$ $I_{\text{pt}}(S) = 0.03$
$I_{\text{scr}}(S) = -0.55$ $I_{\text{pt}}(S) = 1.20$
$I_{\text{scr}}(S) = -1.13$ $I_{\text{pt}}(S) = 2.67$
Sample 3
y: stool
$\hat{y}_{\text{scr}}$: stool √
$\hat{y}_{\text{pt}}$: stool √
$I_{\text{scr}}(S) = -2.63$ $I_{\text{pt}}(S) = -0.02$
$I_{\text{scr}}(S) = -5.65$ $I_{\text{pt}}(S) = 0.01$
$I_{\text{scr}}(S) = 1.17$ $I_{\text{pt}}(S) = 2.23$
$I_{\text{scr}}(S) = 1.07$ $I_{\text{pt}}(S) = 1.92$

Figure 4. Visualization of interactions encoded by the DGCNN trained from scratch (scr) and the DGCNN pre-trained (pt) with IAE. The pre-trained DGCNN typically encodes stronger high-order interactions and weaker low-order interactions compared to the DGCNN trained from scratch.

sification accuracy across different DNNs. We observe that DNNs encoding stronger high-order interactions tend to exhibit higher accuracy. Thus, **we regard this shared phenomenon as the common mechanism behind the performance improvement of different pre-training methods, *i.e.*, different pre-training methods generally enhance the strength of high-order interactions encoded by DNNs, while reducing the strength of low-order interactions, as summarized in Conclusion 1.**

Conclusion 1 reveals that pre-training methods enhance the ability of DNNs to encode complex and global 3D structures, while reducing their reliance on simple and local 3D structures. As simple and local 3D structures (*e.g.*, a curve, a corner) can appear across different categories, they often lack sufficient classification information, so an over-reliance on them may lead to incorrect classifications. For example, as shown in Fig. 4, the DNN trained from scratch incorrectly classifies a "*plant*" sample as a "*stool*". This misclassification may occur because the local structures the DNN learns for the plant, such as the "*stem*" and the "*leaf*", are similar to some local structures of a stool, such as the "*legs*". However, the DNN still encodes a high strength for these local structures (*i.e.*, low-order interactions), which results in an incorrect classification. In contrast, pre-training methods improve the modeling of complex and global 3D structures, allowing DNNs to get a more comprehensive understanding of the input, which in turn enhances their performance. **Thus, beyond traditional accuracy metrics, interactions can help identify the potential reasons for classification errors by revealing which 3D structures modeled by the DNN have inappropriate weights, offering a new perspective for debugging.**

Comparison with transformer-based pre-training methods. We also measure interactions encoded by transformer-based models, including PointBERT [39], PointMAE [12], PointM2AE [42], and PointGPT [4]. These models integrate pre-training methods into the model architecture, making them incompatible with traditional DNNs (*e.g.*, DGCNN). Therefore, we directly compare the interactions encoded by transformer-based models with the interactions encoded by traditional DNNs, including the DNNs trained from scratch and the DNNs trained with pre-training methods. As shown in Fig. 5, transformer-based models also encode stronger high-order interactions and weaker low-order interactions than traditional DNNs trained from scratch, which exhibit a similar pattern to the interactions encoded by traditional DNNs using pre-training methods. This further supports Conclusion 1.

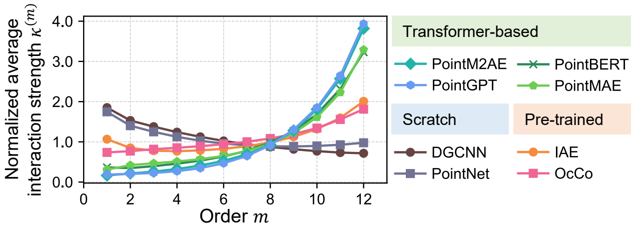

Figure 5. Comparing the normalized average strength of interactions encoded by (1) transformer-based models, (2) traditional DNNs (*e.g.*, DGCNN and PointNet) trained from scratch, and (3) traditional DNNs using pre-training methods (*e.g.*, DGCNN with IAE, and PointNet with OcCo). Results show that transformer-based models also encode stronger high-order interactions and weaker low-order interactions, exhibiting a similar pattern to traditional DNNs using pre-training methods.

4.3. Exploring the impact of different factors on the common mechanism

We further explore two factors that impact the common mechanism: (a) the extent of pre-training, and (b) the amount of fine-tuning data used for downstream tasks.

> **Conclusion 2(a).** The pre-training process progressively enhances the strength of high-order interactions encoded while weakening the strength of low-order interactions as the extent of pre-training increases.

In this subsection, we first investigate the relationship between the extent of pre-training and the strength of interactions encoded by DNNs. Here, the extent of pre-training refers to the number of pre-training epochs, *i.e.*, the range of epochs from the start of pre-training to the epoch at which pre-training converges. To this end, we conduct experiments on DGCNN with two pre-training methods, including IAE and CrossPoint. For each pre-training method, let $T_{\max}$ denote the total number of epochs at which the pre-training process of the DNN converges. We select the DNNs at training epochs $0, 0.2T_{\max}, 0.4T_{\max}, \ldots, T_{\max}$, covering six different stages of the pre-training process. Then, for all

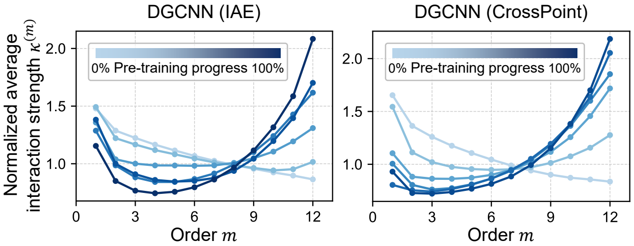

Figure 6. [Conclusion 2(a)] Comparing the normalized average strength of interactions encoded by DGCNNs pre-trained for different extents, ranging from initial pre-training (0%) to full convergence (100%). As the extent of pre-training increases, the strength of high-order interactions encoded by the DNNs typically rises, while the strength of low-order interactions generally decreases.

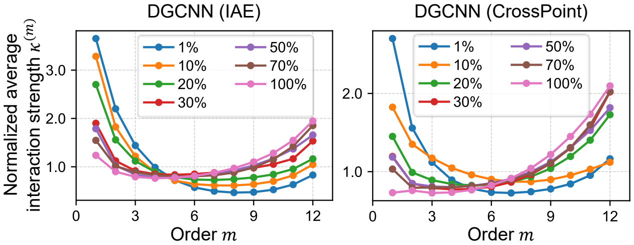

Figure 7. [Conclusion 2(b)] Comparing the normalized average strength of interactions encoded by DNNs fine-tuned with varying amounts of data. Results show that as the amount of fine-tuning data increases from 1% to 100%, the strength of high-order interactions encoded by the DNNs generally increases, while the strength of low-order interactions generally decreases.

DNNs at different pre-training extents, we fine-tune them on the same downstream task and quantify the interactions encoded by these fine-tuned DNNs.

Fig. 6 presents the experimental results. We observe that as the extent of pre-training increases, the strength of high-order interactions encoded by the DNNs generally increases, while the strength of low-order interactions typically decreases. We summarize this relationship between the extent of pre-training and the interactions encoded by DNNs in Conclusion 2(a). This conclusion suggests that sufficient pre-training enhances the model's ability to capture complex and global 3D contexts, further validating the common mechanism outlined in Conclusion 1.

> **Conclusion 2(b).** Increasing the amount of fine-tuning data further enhances the strength of high-order interactions encoded by DNNs, while weakening the strength of low-order interactions.

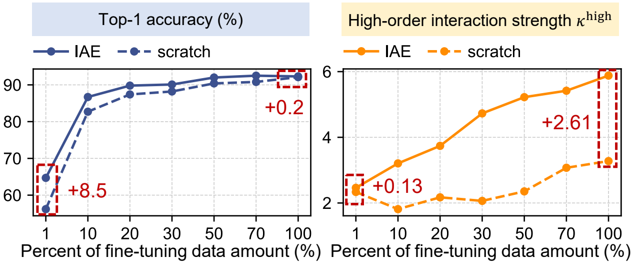

Figure 8. Comparing the classification accuracy and the strength of high-order interactions encoded by different DNNs fine-tuned with varying amounts of data. As the amount of data increases, the accuracy gap between the DNN trained from scratch and the DNN pre-trained with IAE narrows, while the gap in the strength of high-order interactions encoded by these DNNs widens.

To investigate the relationship between the amount of fine-tuning data for downstream tasks and the interactions encoded by DNNs, we construct seven training sets of varying sizes from the ModelNet40 dataset, containing 1%, 10%, 20%, 30%, 50%, 70%, and 100% of the original ModelNet40 training data, respectively. Note that we ensure at least one sample from each class is included, allowing the model to learn from all categories. We then use the different-sized training sets to fine-tune DGCNNs, including those pre-trained using the IAE method and the CrossPoint method. As shown in Fig. 7, as the amount of fine-tuning data increases, the strength of high-order interactions encoded by DNNs gradually increases, while the strength of low-order interactions decreases. We summarize this relationship between the amount of fine-tuning data and the interactions encoded by DNNs in Conclusion 2(b).

4.4. Exploring the potential risk of pre-training methods in reducing DNN's transferability

> **Conclusion 3.** Pre-training methods carry a potential risk of reducing the transferability of features encoded by DNNs.

When exploring the relationship between the amount of fine-tuning data and the interactions encoded by DNNs, we observe the following anomalous phenomenon. As shown in Fig. 8, the gap in classification accuracy between the pre-trained DNN and the DNN trained from scratch becomes marginal as the fine-tuning data increases. For example, when the fine-tuning data reaches 100%, the accuracy gap is only 0.2%. However, the gap in the strength of high-order interactions between the two DNNs gradually increases, indicating that high-order interactions with excessively high strength are not necessary for performance improvement.

Since high-order interactions generally carry a greater risk of overfitting [41], we investigate the potential risk of pre-training methods in reducing the transferability of features encoded by DNNs. Here, the transferability of features refers to the generalization ability of the features. For example, if the features learned from one dataset (*e.g.*, the features of the *airplane* class in ModelNet) can be applied to another unseen dataset (*e.g.*, identifying the *airplane* class in ShapeNet), we consider these features to have high transferability. To this end, we use ShapeNet as the unseen

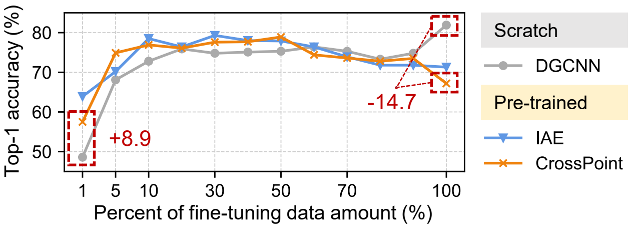

Figure 9. [Conclusion 3] Comparing the zero-shot classification accuracy of DNNs with and without pre-training, followed by fine-tuning with varying amounts of data. Results show that the zero-shot accuracy of the pre-trained DNN initially exceeds that of the DNN trained from scratch when the fine-tuning data is limited (*e.g.*, 1%), but falls below that of the DNN trained from scratch as the fine-tuning data becomes sufficient (*e.g.*, 100%).

dataset and compare the classification accuracy of different DNNs, including DGCNNs trained with varying amounts of data from ModelNet, as well as DGCNNs pre-trained and then fine-tuned with varying amounts of data. Since the category labels in the two datasets do not completely align, we identify eight common categories. Please see the supplementary material for more implementation details.

Fig. 9 shows the results. We find that when the amount of fine-tuning data is limited (*e.g.*, 1%), pre-trained DNNs, such as the DNN pre-trained with CrossPoint, achieve higher zero-shot accuracy (+8.9%) compared to the DNN trained from scratch. In contrast, when the fine-tuning data is sufficient (*e.g.*, 100%), the accuracy of the DNN pre-trained with CrossPoint significantly lags behind that of the DNN trained from scratch (-14.7%). We attribute this to pre-training methods causing DNNs to encode high-order interactions with excessively high strength, which in turn reduces the transferability of the features encoded by the DNNs. Note that we are not criticizing the use of pre-training methods to enhance the strength of high-order interactions encoded by DNNs as inherently negative. Rather, we are proposing this potential risk and offering new insights for the design of pre-training methods.

5. Guiding the training process using the common mechanism

Traditional pre-training methods, while improving performance, inevitably require extensive pre-training on large-scale unlabeled datasets, which demands considerable time and computational resources. As discussed above, we find that the common mechanism underlying different pre-training methods is that they universally enhance the strength of high-order interactions encoded by DNNs while reducing the strength of low-order interactions. Building on this insight, we propose a new method that directly enhances the strength of high-order interactions encoded by DNNs while reducing the strength of low-order interactions. In this way, **our method achieves performance comparable to traditional pre-training methods while avoiding the need for pre-training on large-scale unlabeled datasets**. Specifically, we introduce a new heuristic loss term defined as follows.

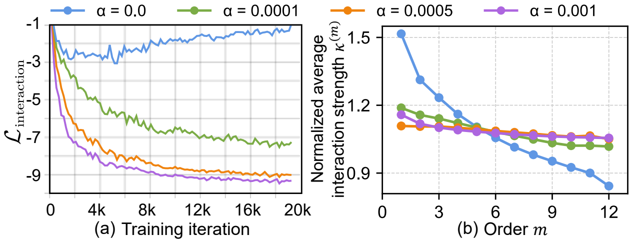

Figure 10. (a) Curves showing the values of the proposed loss term $\mathcal{L}_{\text{interaction}}$ for different values of α throughout the training process. (b) Comparison of the normalized average strength of interactions encoded by DNNs for various α values in the loss term.

$$\mathcal{L}_{\text{interaction}} = \mathbb{E}_x \left[\mathbb{E}_{|S|\in\Omega^{\text{low}}} \left[|I(S)|\right] - \mathbb{E}_{|S|\in\Omega^{\text{high}}} \left[|I(S)|\right]\right], \quad (6)$$

where Ω^{high} and Ω^{low} define the ranges of high-order and low-order interactions, as detailed in Sec. 3. Minimizing the loss term $\mathcal{L}_{\text{interaction}}$ forces the DNN to weaken the strength of low-order interactions, *i.e.*, decreasing $\mathbb{E}_x\mathbb{E}_{|S|\in\Omega^{\text{low}}} [|I(S)|]$, while enhancing the strength of high-order interactions, *i.e.*, increasing $\mathbb{E}_x\mathbb{E}_{|S|\in\Omega^{\text{high}}} [|I(S)|]$.

However, computing Eq. (6) is NP-hard. To overcome this challenge, we approximate $\mathcal{L}_{\text{interaction}}$ using a sampling-based approach. Specifically, given a point cloud x with n regions indexed by $N = \{1, 2, ..., n\}$, we sample three disjoint subsets $S_1, S_2, S_3 \subseteq N$ where the orders of the subsets $|S_1|, |S_2|, |S_3| \in \Omega^{\text{low}}$, with each subset representing a low-order interaction encoded by the DNN. We consider the union $S_{\text{union}} = S_1 \cup S_2 \cup S_3$ as a relatively high-order interaction. Then, we can approximate the interaction loss $\mathcal{L}_{\text{interaction}}$ as follows.

$$\mathcal{L}'_{\text{interaction}} = \mathbb{E}_{S_1,S_2,S_3\subseteq N} \left[\mathbb{E}_{i\in\{1,2,3\}} \left[|I(S_i)|\right] - |I(S_{\text{union}})|\right]. \quad (7)$$

Given a traditional DNN, we incorporate the interaction loss into the training process using the following loss function for the classification task, **without the need for additional pre-training on large-scale unlabeled datasets**.

$$\mathcal{L} = \mathcal{L}_{\text{classification}} + \alpha\mathcal{L}_{\text{interaction}}, \quad (8)$$

where $\mathcal{L}_{\text{classification}}$ denotes the standard classification loss function (*e.g.*, cross-entropy loss), and $\alpha > 0$ is the hyper-parameter controlling the strength of the interaction loss. Please see Tab. 4 for the effects of varying α. As shown in Fig. 10 (b), the strength of high-order interactions encoded by the DNN with $\alpha > 0$ is generally higher than the result when $\alpha = 0$, but it does not increase indefinitely as α grows. This shows the effectiveness of our interaction loss.

Experiments and results analysis. To evaluate the effectiveness of the proposed loss term, we conduct experi-

Method	ModelNet40	ScanObjectNN	No Pre-train
PointNet	89.2	68.0	✓
PointNet + JigSaw	89.6	-	✗
PointNet + OcCo	**90.1**	-	✗
PointNet + $\mathcal{L}_{\text{interaction}}$ (Ours)	**90.1**	**69.0**	✓
DGCNN	92.5	78.1	✓
DGCNN + JigSaw	92.6	83.5	✗
DGCNN + OcCo	93.0	84.3	✗
DGCNN + STRL	93.1	-	✗
DGCNN + CrossPoint	92.8	-	✗
DGCNN + IAE	**94.2**	**85.6**	✗
DGCNN + $\mathcal{L}_{\text{interaction}}$ (Ours)	93.3	79.4	✓
CurveNet	92.8	79.2	✓
CurveNet + $\mathcal{L}_{\text{interaction}}$ (Ours)	**93.1**	**82.0**	✓
GDANet	92.3	78.7	✓
GDANet + $\mathcal{L}_{\text{interaction}}$ (Ours)	**92.8**	**80.0**	✓

Table 2. Classification accuracy (%) on ModelNet40 and ScanObjectNN datasets. The best results are shown in bold and the second-best results are underlined. Our method achieves results comparable to pre-training methods, while not requiring pre-training on large-scale datasets.

ments on 3D point cloud classification and semantic segmentation tasks. For the classification task, we use the ModelNet40 and ScanObjectNN datasets, as described in Sec. 4.1. Specifically, for the ScanObjectNN dataset, we conduct experiments using the PB_T50_RS variant, which is the most challenging variant. We train PointNet and DGCNN using the proposed loss term and set α to 0.0005. As shown in Tab. 2, our proposed loss term consistently improves the performance of PointNet, DGCNN, CurveNet [11], and GDANet [37] on both the ModelNet40 and the ScanObjectNN testing splits, compared to their original versions. Moreover, our method demonstrates performance comparable to pre-training methods, without the need for pre-training on large-scale datasets.

For the semantic segmentation task, we conduct experiments on the Stanford Large-Scale 3D Indoor Spaces (S3DIS) dataset [2]. The S3DIS consists of 3D point clouds collected from six distinct large-scale indoor environments, with each point cloud annotated with per-point categorical labels. We randomly subsample 4,096 points from the original point cloud and apply 6-fold cross-validation during fine-tuning. Since the proposed interaction loss is specifically designed for 3D classification, it cannot be directly applied to segmentation tasks. Instead, we adopt a two-stage training approach: first, we train a DNN on the classification task with our interaction loss, and then fine-tune the model on the semantic segmentation task. As shown in Tab. 3, the DGCNN using the proposed loss term achieves 86.8% overall accuracy and 59.0% mIoU, outperforming the majority of pre-training methods. Additionally, our loss term also improves the performance of the PointNet.

Method	S3DIS 6-Fold	
	OA	mIoU
PointNet	78.5	47.6
PointNet + $\mathcal{L}_{\text{interaction}}$ (Ours)	**82.1**	**50.8**
DGCNN	84.1	56.1
DGCNN + JigSaw	84.4	56.6
DGCNN + OcCo	85.1	58.5
DGCNN + STRL	84.2	57.1
DGCNN + IAE	85.9	**60.7**
DGCNN + $\mathcal{L}_{\text{interaction}}$ (Ours)	**86.8**	59.0

Table 3. Semantic segmentation on S3DIS. We report Overall Accuracy (OA) and mean Intersection over Union (mIoU) across six folds. Our method surpasses most pre-training methods.

α	ModelNet40	ScanObjectNN
0.0	92.5	78.1
0.0001	93.0	79.0
0.0005	**93.3**	**79.4**
0.001	91.3	78.1

Table 4. Classification accuracy (%) for DGCNNs trained with varying hyper-parameters α for the interaction loss.

Effects of the hyper-parameter α. We train the DGCNN with various interaction loss weights α and evaluate the testing accuracy, as shown in Tab. 4. The accuracy initially increases and then decreases as α rises. We attribute this to the loss term enhancing the strength of high-order interactions encoded by the DNN. At lower α values, the interaction loss improves the DNN's modeling of global 3D structures. However, excessively high values of α lead to excessively high strength of high-order interactions, increasing the risk of overfitting, as discussed in Conclusion 3. With an appropriately chosen α, the interaction loss effectively enhances the training process, further supporting the common mechanism outlined in Conclusion 1.

6. Conclusion

In this paper, we use interactions to investigate the common mechanism underlying the effectiveness of different pre-training methods for 3D point clouds. Specifically, these methods generally enhance the strength of high-order interactions encoded by DNNs, while reducing the strength of low-order interactions. We then explore the impact of various factors on the mechanism and find that sufficient pre-training and adequate fine-tuning data further reinforce this mechanism. Additionally, we identify a potential risk that pre-training may reduce the transferability of DNNs. Based on the common mechanism, we propose a new method that directly enhances the strength of high-order interactions encoded by DNNs while weakening the strength of low-order interactions. Experiments show that our method achieves performance comparable to pre-training methods, without the need for pre-training on large-scale datasets.

Acknowledgments

This work is partially supported by the National Nature Science Foundation of China (No. 62206170, 62376199).

References

[1] Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9902–9912, 2022. 1, 2, 4

[2] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1534–1543, 2016. 8

[3] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 4

[4] Guangyan Chen, Meiling Wang, Yi Yang, Kai Yu, Li Yuan, and Yufeng Yue. Pointgpt: Auto-regressively generative pre-training from point clouds. *Advances in Neural Information Processing Systems*, 36, 2024. 5

[5] Jiajing Chen, Burak Kakillioglu, Huantao Ren, and Senem Velipasalar. Why discard if you can recycle?: A recycling max pooling module for 3d point cloud analysis. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 559–567, 2022. 2

[6] Lu Chen, Siyu Lou, Benhao Huang, and Quanshi Zhang. Defining and extracting generalizable interaction primitives from dnns. *arXiv preprint arXiv:2401.16318*, 2024. 1

[7] Xu Cheng, Chuntung Chu, Yi Zheng, Jie Ren, and Quanshi Zhang. A game-theoretic taxonomy of visual concepts in dnns. *arXiv preprint arXiv:2106.10938*, 2021. 3

[8] Michel Grabisch and Marc Roubens. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of game theory*, 28:547–565, 1999. 3, 1

[9] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6535–6545, 2021. 1, 2, 4

[10] Mingjie Li and Quanshi Zhang. Does a neural network really encode symbolic concepts? In *International conference on machine learning*, pages 20452–20469. PMLR, 2023. 2, 3

[11] AAM Muzahid, Wanggen Wan, Ferdous Sohel, Lianyao Wu, and Li Hou. Curvenet: Curvature-based multitask learning deep networks for 3d object recognition. *IEEE/CAA Journal of Automatica Sinica*, 8(6):1177–1187, 2020. 2, 8

[12] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *European conference on computer vision*, pages 604–621. Springer, 2022. 5

[13] Omid Poursaeed, Tianxing Jiang, Han Qiao, Nayun Xu, and Vladimir G Kim. Self-supervised learning of point clouds via orientation estimation. In *2020 International Conference on 3D Vision (3DV)*, pages 1018–1028. IEEE, 2020. 1, 2

[14] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2, 4

[15] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in neural information processing systems*, 35:23192–23204, 2022. 2

[16] Yongming Rao, Jiwen Lu, and Jie Zhou. Global-local bidirectional reasoning for unsupervised representation learning of 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5376–5385, 2020. 1, 2

[17] Jie Ren, Mingjie Li, Qirui Chen, Huiqi Deng, and Quanshi Zhang. Towards axiomatic, hierarchical, and symbolic explanation for deep models. *arXiv preprint arXiv:2111.06206v5*, 2021. 2, 1

[18] Jie Ren, Zhanpeng Zhou, Qirui Chen, and Quanshi Zhang. Can we faithfully represent masked states to compute shapley values on a dnn? *arXiv preprint arXiv:2105.10719*, 2021. 2

[19] Jie Ren, Mingjie Li, Qirui Chen, Huiqi Deng, and Quanshi Zhang. Defining and quantifying the emergence of sparse concepts in dnns. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20280–20289, 2023. 3

[20] Qihan Ren, Yang Xu, Junpeng Zhang, Yue Xin, Dongrui Liu, and Quanshi Zhang. Towards the dynamics of a dnn learning symbolic interactions. *arXiv preprint arXiv:2407.19198*, 2024. 1, 3

[21] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 2, 4

[22] Lloyd S Shapley. A value for n-person games. *Contribution to the Theory of Games*, 2, 1953. 3, 1

[23] Charu Sharma and Manohar Kaul. Self-supervised few-shot learning on point clouds. *Advances in Neural Information Processing Systems*, 33:7212–7221, 2020. 1, 2

[24] Wen Shen, Binbin Zhang, Shikun Huang, Zhihua Wei, and Quanshi Zhang. 3d-rotation-equivariant quaternion neural networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 531–547. Springer, 2020. 2

[25] Wen Shen, Qihan Ren, Dongrui Liu, and Quanshi Zhang. Interpreting representation quality of dnns for 3d point cloud processing. *Advances in Neural Information Processing Systems*, 34:8857–8870, 2021. 1, 3

[26] Wen Shen, Zhihua Wei, Shikun Huang, Binbin Zhang, Panyue Chen, Ping Zhao, and Quanshi Zhang. Verifiability and predictability: Interpreting utilities of network architectures for point cloud processing. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 10703–10712, 2021. 1, 4

[27] Wen Shen, Zhihua Wei, Qihan Ren, Binbin Zhang, Shikun Huang, Jiaqi Fan, and Quanshi Zhang. Interpretable rotation-equivariant quaternion neural networks for 3d point cloud processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3290–3304, 2024. 2

[28] Mukund Sundararajan, Kedar Dhamdhere, and Ashish Agarwal. The shapley taylor interaction index. In *International conference on machine learning*, pages 9259–9268. PMLR, 2020. 3, 1

[29] Ling Tang, Wen Shen, Zhanpeng Zhou, Yuefeng Chen, and Quanshi Zhang. Defects of convolutional decoder networks in frequency representation. *arXiv preprint arXiv:2210.09020*, 2022. 3

[30] Ali Thabet, Humam Alwassel, and Bernard Ghanem. Self-supervised learning of local features in 3d point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 938–939, 2020. 1, 2

[31] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1588–1597, 2019. 4

[32] Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J Kusner. Unsupervised point cloud pre-training via occlusion completion. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9782–9792, 2021. 1, 2, 4

[33] Xin Wang, Jie Ren, Shuyun Lin, Xiangming Zhu, Yisen Wang, and Quanshi Zhang. A unified approach to interpreting and boosting adversarial transferability. *arXiv preprint arXiv:2010.04055*, 2020. 2

[34] Xin Wang, Shuyun Lin, Hao Zhang, Yufei Zhu, and Quanshi Zhang. Interpreting attributions and interactions of adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1095–1104, 2021. 2, 3

[35] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019. 2, 4

[36] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 4

[37] Mutian Xu, Junhao Zhang, Zhipeng Zhou, Mingye Xu, Xiaojuan Qi, and Yu Qiao. Learning geometry-disentangled representation for complementary understanding of 3d object point cloud. In *Proceedings of the AAAI conference on artificial intelligence*, pages 3056–3064, 2021. 2, 8

[38] Siming Yan, Zhenpei Yang, Haoxiang Li, Chen Song, Li Guan, Hao Kang, Gang Hua, and Qixing Huang. Implicit autoencoder for point-cloud self-supervised representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14530–14542, 2023. 1, 2, 4

[39] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19313–19322, 2022. 5

[40] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 international conference on 3D vision (3DV)*, pages 728–737. IEEE, 2018. 2, 4

[41] Hao Zhang, Sen Li, Yinchao Ma, Mingjie Li, Yichen Xie, and Quanshi Zhang. Interpreting and boosting dropout from a game-theoretic view. *arXiv preprint arXiv:2009.11729*, 2020. 1, 2, 6

[42] Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. *Advances in neural information processing systems*, 35:27061–27074, 2022. 5

[43] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10252–10263, 2021. 1, 2

[44] Huilin Zhou, Huijie Tang, Mingjie Li, Hao Zhang, Zhenyu Liu, and Quanshi Zhang. Explaining how a neural network play the go game and let people learn. *arXiv preprint arXiv:2310.09838*, 2023. 1

[45] Huilin Zhou, Hao Zhang, Huiqi Deng, Dongrui Liu, Wen Shen, Shih-Han Chan, and Quanshi Zhang. Concept-level explanation for the generalization of a dnn. *arXiv preprint arXiv:2302.13091*, 2023. 2, 3