Improved MLP Point Cloud Processing with High-Dimensional Positional Encoding

Yanmei Zou[1], Hongshan Yu[1*], Zhengeng Yang[2*], Zechuan Li[1], Naveed Akhtar[3]

[1]College of Electrical and Information Engineering, Quanzhou Innovation Institute, Hunan University, Changsha, China
[2]College of Engineering and Design, Hunan Normal University, Changsha, China
[3]School of Computing and Information Systems, The University of Melbourne, 3052 Victoria, Australia
zouyanmei@hnu.edu.cn, yuhongshancn@hotmail.com, yzg050215@163.com, lizechuan@hnu.edu.cn, naveed.akhtar1@unimelb.edu.au

Abstract

Multi-Layer Perceptron (MLP) models are the bedrock of contemporary point cloud processing. However, their complex network architectures obscure the source of their strength. We first develop an "abstraction and refinement" (ABS-REF) view for the neural modeling of point clouds. This view elucidates that whereas the early models focused on the ABS stage, the more recent techniques devise sophisticated REF stages to attain performance advantage in point cloud processing. We then borrow the concept of "positional encoding" from transformer literature, and propose a High-dimensional Positional Encoding (HPE) module, which can be readily deployed to MLP based architectures. We leverage our module to develop a suite of HPENet, which are MLP networks that follow ABS-REF paradigm, albeit with a sophisticated HPE based REF stage. The developed technique is extensively evaluated for 3D object classification, object part segmentation, semantic segmentation and object detection. We establish new state-of-the-art results of 87.6 mAcc on ScanObjectNN for object classification, and 85.5 class mIoU on ShapeNetPart for object part segmentation, and 72.7 and 78.7 mIoU on Area-5 and 6-fold experiments with S3DIS for semantic segmentation. The source code for this work is available at https://github.com/zouyanmei/HPENet.

Introduction

The increasing popularity of 3D sensors is currently fueling a wide use of 3D point clouds in numerous application domains, such as autonomous driving (Zheng et al. 2021; Shi et al. 2022), robotics (Li et al. 2022) and geological surveying (Kong, Wu, and Saroglou 2020). Unlike digital images with regular 2D grid structures, 3D points in a typical point cloud are irregularly located in 3D space. This intrinsic irregularity causes considerable challenges in processing point clouds with neural networks.

Existing neural network based point cloud processing methods can be categorized into two broad categories: voxel based (Huang and You 2016; Choy, Gwak, and Savarese 2019) and point based methods (Zhao et al. 2021; Qian et al. 2022; Qi et al. 2017a). The former discretize the underlying 3D space into volumetric units before processing the point cloud. This generally helps in making the methods computationally efficient. However, the discretization process also results in a noticeable loss of fine-grained geometric information. The seminal work of PointNet (Qi et al. 2017a) originally demonstrated the possibility of directly processing point clouds with Multi-Layer Perceptron (MLP) based neural models. Since PointNet, numerous point based methods have surfaced, e.g., PointNet++ (Qi et al. 2017b), PointConv (Wu, Qi, and Fuxin 2019), PointNeXt (Qian et al. 2022). A key attribute of such methods is that they employ sophisticated local feature aggregation schemes to encode strong representations of the point clouds. For instance, PointNet++ uses a hierarchical network structure for that purpose, whereas PointConv employs a density-aware discrete convolution for high-quality local feature aggregation. The more recent PointNeXt proposes an inverted residual bottleneck module to improve PointNet++ scalability.

In recent years, the success of transformers in the natural language processing (Vaswani et al. 2017; Devlin et al. 2018) and computer vision domains (Dosovitskiy et al. 2020; Liu et al. 2021a) has also motivated transformer based neural models for directly processing 3D point clouds. To that end, Point Transformer (Zhao et al. 2021) and other recent methods, e.g., (Lai et al. 2022; Zhang et al. 2022), use transformer architectures for an even more sophisticated feature aggregation. These efforts are emerging in parallel to the MLP networks for point clouds (Choe et al. 2022; Ma et al. 2022; Qian et al. 2022). One of the intended contributions of this paper is to show that the key feature extraction modules used by the conventional MLP based methods and the emerging transformer based techniques essentially follow the same two-stage "abstraction and refinement" (ABS-REF) paradigm. We discuss this unified view of the latest techniques in detail in the *Proposed Method* Section.

Under our ABS-REF perspective, it becomes clear that whereas the early works, e.g., PointNet++ (Qi et al. 2017b) and PointConv (Wu, Qi, and Fuxin 2019), employ sophisticated local feature aggregation strategies at the ABS stage, they generally lack the REF stage. As compared to them, success of the more recent techniques can be attributed to the REF stage, which enables an increased receptive field of the network and a greater extent of context information considerations. These factors are crucial for discriminative

*Corresponding author.

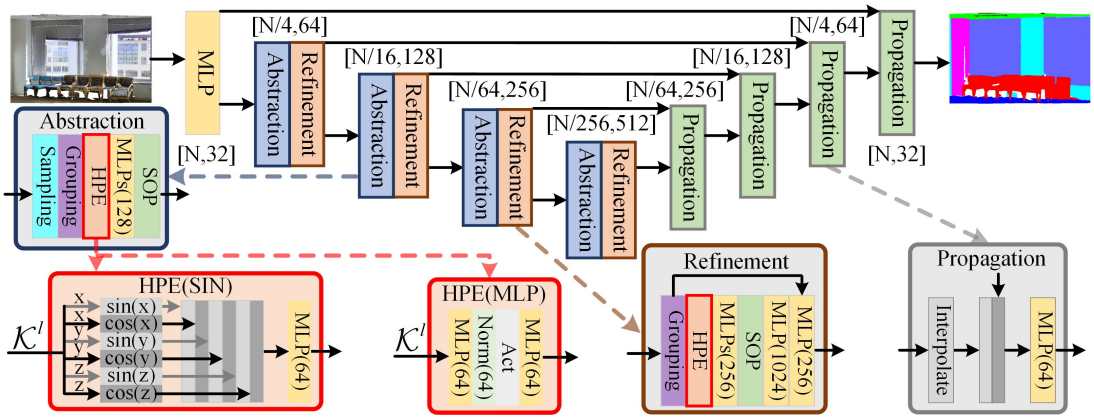

Figure 1: HPENet architecture for semantic segmentation. The network delineates between Abstraction (ABS) and Refinement (REF) stages of feature extraction, and uses the proposed High-dimensional Positional Encoding (HPE) module in both stages.

feature learning, which leads to better performance.

Positional information is the key intrinsic property of point clouds. However, point based methods often treat the point positions as an added information by concatenating other features and relative point positions, e.g., PointNet++ (Qi et al. 2017b). Though useful, this strategy lacks in giving the point positional information its due attention. Fortunately, the notion of positional encoding, which originated in the transformer literature (Vaswani et al. 2017), potentially provides an algorithmic solution to this problem by enabling positional information embedding in a feature space. Inspired, we propose positional encoding for MLP based point cloud modeling, thereby allowing explicit incorporation of the positional information along with the relative local point relations in the models.

Indeed, we can find existing instances of leveraging positional encoding in point based models. However, those approaches are either transformer (not MLP) based (Zhao et al. 2021) or they use non-learnable encodings which is not adaptive, e.g., Position Pooling in (Liu et al. 2020). Our technique enables exploiting adaptive positional encoding in MLP based architectures. Due to a low-dimensional representation, the relative geometric relationships in a point cloud are often not sufficiently encoded by point coordinates for the modeling purpose. Hence, we enrich the geometric relationship representation by first projecting the point coordinates onto a high-dimensional space. We allow this in both data-driven and parameter-free manners. The enrichment is followed by an MLP to align the high-dimensional vectors to their corresponding feature space. This process is packed in a High-dimensional Positional Encoding (HPE) module. This module is used to devise our HPENets, see Fig. 1. Our key contributions are summarised as follows.

- We identify a unified "abstraction and refinement" paradigm underpinning the current high-performing point cloud modeling techniques, which allows an intuitive delineation of the key strengths of the methods.
- We propose a High-dimensional Positional Encoding (HPE) scheme for effective point cloud geometric representation with positional information. The HPE scheme can be generically used to enhance MLP architectures.
- We propose HPENets, which are ABS-REF stage inspired MLP networks that leverages our HPE modules in both ABS and REF stages.
- With an extensive evaluation of our technique, we establish state-of-the-art (SOTA) results[1] of 87.6 mAcc on ScanObjectNN for object classification, 85.5 class mIoU on ShapeNetPart for object part segmentation, and 72.7 and 78.7 mIoU on Area-5 and 6-fold experiments with S3DIS for semantic segmentation.

Related Work

Due to the intrinsic limitations of voxel based methods (Choy, Gwak, and Savarese 2019; Thomas et al. 2019), point based methods (Qi et al. 2017a; Wu, Qi, and Fuxin 2019; Hu et al. 2020) have attracted considerable attention of the research community in recent years for point cloud processing. Existing point based methods can be broadly categorized into four groups, namely; MLP based (Tolstikhin et al. 2021; Lian et al. 2021; Tang et al. 2022; Wang et al. 2022), convolution based (Engelmann, Kontogianni, and Leibe 2020; Xu et al. 2021), attention based (Zhao et al. 2021; Lai et al. 2022) and graph based (Shen et al. 2018; Wang et al. 2019) methods. Key contributions along these categories are discussed below.

MLP based methods apply MLPs to extract pointwise features and then use a symmetric operation such as max-pooling or average-pooling on the point groups to obtain high-level features. After the pioneering work of PointNet (Qi et al. 2017a), numerous MLP-based techniques have emerged. Most of them focus on devising sophisticated modules to extract the local geometric structure (Qi et al. 2017b). Inspired by the widely used SIFT descriptor (Lowe 2004), PointSIFT (Jiang et al. 2018) develops a 3D SIFT descriptor that considers eight crucial orientations and scales for local scale-invariant feature transform. To improve the generalisation and performance of MLP-based networks,

[1]Our claim is limited to the techniques that, similar to our approach, do not benefit from pre-training, voting or ensembling.

PointMLP (Ma et al. 2022) proposes a local geometric affine module to transform point features in local regions adaptively. More recently, PointNeXt (Qian et al. 2022) proposes an inverted residual MLP module for improved scalability.

Convolution based methods focus on designing a local convolution kernel suitable for point cloud processing. For instance, PointConv (Wu, Qi, and Fuxin 2019) proposes a density-aware discrete convolution kernel which comprises weight and density functions, whereas KPConv (Thomas et al. 2019) presents a kernel point convolution which uses any number of kernel points to process various point clouds. Engelmann, Kontogianni, and Leibe (2020) employ a dilated point convolution to increase the receptive field size of point convolutional networks. Lei, Akhtar, and Mian (2019) proposed a spherical kernel that uses an Octree-guided CNN for point cloud process. Their method is further enhanced in (Lei, Akhtar, and Mian 2020) for graph convolution.

Attention based methods exploit attention mechanisms to model long-range dependency between point pairs in a set. These methods are mainly inspired by the attention mechanism (Vaswani et al. 2017), which was first introduced in natural language processing. In order to efficiently process large-scale point clouds, RandLA-Net (Hu et al. 2020) uses random point sampling to guarantee efficiency and attention-based local feature aggregation for better performance. Both in natural language processing (Vaswani et al. 2017; Devlin et al. 2018) and computer vision domains (Dosovitskiy et al. 2020; Liu et al. 2021a), attention mechanism is currently causing a paradigm shift. Since Point Transformer (Zhao et al. 2021), point cloud processing has also started to benefit from this mechanism considerably using the transformer architectures (Park et al. 2022; Lai et al. 2022; Yu et al. 2022; Guo et al. 2021).

Graph based methods employ graph structure to extract features, which generally treat points as nodes and feature relations as edges. Landrieu and Simonovsky (2018) proposed the superpoint graph to deal with large-scale 3D semantic segmentation tasks. CurveNet (Xiang et al. 2021) pays attention to graph structure and employs a shape descriptor, termed "curves", using guided walks in point clouds. Other examples of graph-based methods also include (Lei, Akhtar, and Mian 2020).

Proposed Method

Image processing models are currently experiencing and paradigm shift at the hands of transformers (Dosovitskiy et al. 2020; Liu et al. 2021a). Following the suite, many recent works are directly importing the transformer architectures to point cloud modeling (Zhao et al. 2021; Lai et al. 2022). However, point cloud data has its peculiar nature. We envisage that a more systematic delineation of the strengths of the existing point cloud techniques can better guide the adoption of relevant concepts of transformers in the point cloud domain. Hence, we first provide a new perspective on the existing point-based methods, and then propose a high-dimensional positional encoding enhancement for MLP-based methods. Below, we first briefly introduce the mathematical notions used in the remaining paper.

A point cloud with N points can be considered comprising two sets of distinct elements, namely; the point set $\mathcal{P} = \left\{p_m \in \mathbb{R}^{1\times 3}\right\}_{m=1}^{N}$ and the feature set $\mathcal{F} = \left\{f_m \in \mathbb{R}^{1\times c}\right\}_{m=1}^{N}$, where p_m is the position of the m-th point and f_m is the corresponding feature with c channels. In a typical neural model, after a sampling layer, a smaller point cloud is generated with N^{l+1} points, such that $N^{l+1} < N^l$. Here, l is the index of the sampling layer. By using a grouping operation to group k points neighboring a sampled point in a local region, we get grouped point sets $\mathcal{K} = \left\{k_m \in \mathbb{R}^{k\times 3}\right\}_{m=1}^{N}$ and the corresponding feature sets $\mathcal{D} = \left\{d_m \in \mathbb{R}^{k\times c}\right\}_{m=1}^{N}$.

Abstraction and Refinement View

In Fig. 2, we illustrate the two-stage "abstraction and refinement" (ABS-REF) view of the major existing and the proposed technique. This view is largely inspired by the intuitions behind the subsampling and convolution blocks in the image processing domain. We find that point cloud literature currently generally lacks in a clear delineation between the adopted abstraction and refinement processes, which adversely contributes to developing effective techniques.

Abstraction (ABS) stage: Analogous to the subsampling operation performed in the image processing networks, we can identify an abstraction (ABS) stage for the point cloud networks. Effectively, this stage eventually abstracts features from input point cloud and produces a new point cloud with fewer points. The stage can be composed of multiple operations, including a sampling operation (Eq. 1), a grouping operation (Eq. 2), and an intra-set feature aggregation operation (Eq. 3). Commonly, the sampling operation selects a new point set with fewer elements using Farthest Point Sampling (FPS), which leverages the centroids of local regions for subsampling. The grouping operation generally selects neighboring points around the centroids to define local region sets using, e.g., k-Nearest Neighbors (KNNs). Since the aggregation operation in ABS stage abstracts local context information from a set to the corresponding centroid, we call it intra-set operation. Concretely, given a point set $\mathcal{P}^l$ and its corresponding feature set $\mathcal{F}^l$, we get the point set $\mathcal{P}^{l+1}$, grouped point sets $\mathcal{K}_{ABS}^{l+1}$, and feature sets $\mathcal{D}_{ABS}^{l+1}$ after the sampling and grouping operations. We use the subscript ABS to emphasize the ABStraction stage. In this stage, the intra-set feature aggregation operation h_{ABS} encodes local region patterns into the feature vectors and aggregates local context information intra set. Overall, the abstraction stage can be mathematically expressed as

$$\mathcal{P}^{l+1} = \text{FPS}\left(\mathcal{P}^l\right), p_m^{l+1} \in \mathcal{P}^{l+1}, \tag{1}$$

$$\mathcal{D}_{ABS}^{l+1}(p_m^{l+1}), \mathcal{K}_{ABS}^{l+1}(p_m^{l+1}) = \text{KNN}(p_m^{l+1}, \mathcal{P}^l, \mathcal{F}^l), \tag{2}$$

$$f_m^{l+1} = h_{ABS}\left(\mathcal{D}_{ABS}^{l+1}\left(p_m^{l+1}\right), \mathcal{K}_{ABS}^{l+1}\left(p_m^{l+1}\right)\right), \tag{3}$$

where $\mathcal{D}_{ABS}^{l+1}(p_m^{l+1})$ and $\mathcal{K}_{ABS}^{l+1}(p_m^{l+1})$ are the neighbor feature and point sets of the centroid p_m^{l+1}, respectively.

Refinement (REF) stage: Inspired by the underlying objective of the convolution block in image processing networks, we can identify a refinement (REF) stage in point

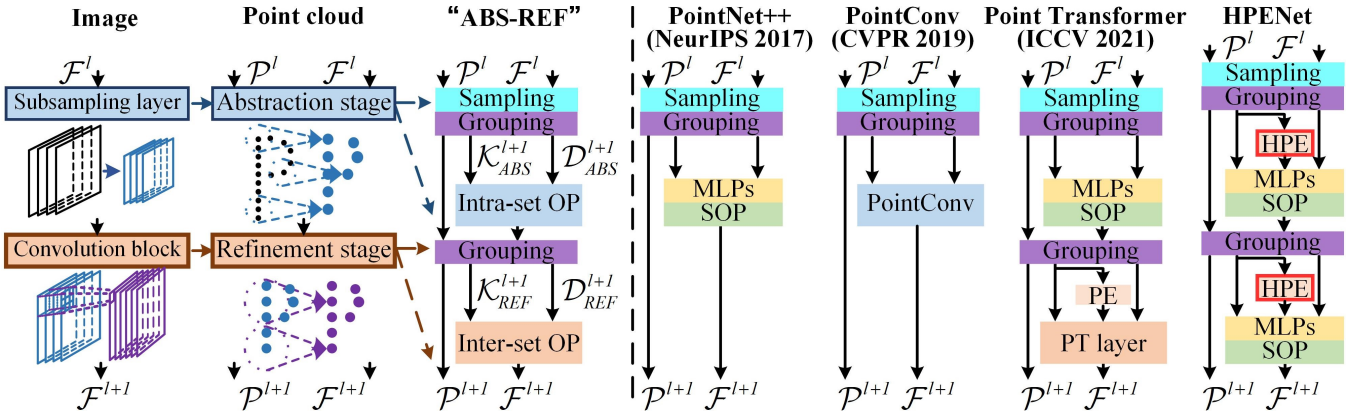

Figure 2: "Abstraction and Refinement" (ABS-REF) perspective. *Left:* The proposed ABS-REF view of point cloud models is analogous to subsampling and convolution block view in image models. The shown "ABS-REF" column expands abstraction and refinement stages. *Right:* Representative instantiations of the ABS-REF framework. Whereas early methods, e.g., PointNet++, PointConv, ignore REF stage, more recent techniques, e.g., Point Transformer, achieve higher performance by accounting for REF stage in point cloud models. Abbreviations include SOP: Symmetric OPeration, OP: aggregation OPeration, PT: Point Transformer, HPE: proposed High-dimensional Positional Encoding.

cloud networks. This stage aims to refine the centroid features by gathering local context information. Specifically, the REF stage further processes the point set $\mathcal{P}^{l+1}$ and features set $\mathcal{F}^{l+1}_{ABS}$ generated by the ABS stage. In Fig. 2 (left), we illustrate a simplified architecture of the refinement stage in the adopted "ABS-REF" view of the techniques. In the refinement stage, a grouping operation (Eq. 4) is first used to group the local sets in centroid point cloud. Later, an inter-set feature aggregation operation h_{REF} is employed to extract and aggregate the inter-set context information. Mathematically, the REF stage can be expressed as

$$\mathcal{D}^{l+1}_{REF}(p^{l+1}_m), \mathcal{K}^{l+1}_{REF}(p^{l+1}_m) = \text{KNN}(p^{l+1}_m, \mathcal{P}^{l+1}, \mathcal{F}^{l+1}_{ABS}), \tag{4}$$

$$f^{l+1}_m = h_{REF}\left(\mathcal{D}^{l+1}_{REF}\left(p^{l+1}_m\right), \mathcal{K}^{l+1}_{REF}\left(p^{l+1}_m\right)\right), \tag{5}$$

where $\mathcal{D}^{l+1}_{REF}(p^{l+1}_m)$ and $\mathcal{K}^{l+1}_{REF}(p^{l+1}_m)$ are the neighbor feature set and point set of the centroid p^{l+1}_m, respectively.

The benefits of joint application of ABS and REF stages in a network are two-fold. First, the effective receptive field of the network gains from the REF stage. Ideally, a centroid's receptive field is k_{ABS} in the ABS stage, while $k_{ABS} \times k_{REF}$ in the REF stage, where k_{ABS} and k_{REF} are the number of neighbor points of a set in the ABS and REF stages. Second, the REF stage helps improving the scalability by increasing the network depth by stacking REF stages, similar to stacking convolutional blocks for images.

Instantiation of ABS-REF framework: To exemplify systematic understanding of point cloud models under our ABS-REF perspective, we provide representative examples in Fig. 2 (right). It can be seen that PointNet++ (Qi et al. 2017b) and PointConv (Wu, Qi, and Fuxin 2019) only have the ABS stage. Although the two models use different intra-set operations for local feature aggregation, both are single stage models under our perspective. PointNet++ employs MLPs, while PointConv uses the density-aware discrete convolution. Nevertheless, both models are essentially void of the REF stage. More recently, Point Transformer (Zhao et al. 2021) has reported impressive results. Incidentally, we can easily identify an additional REF stage in Point Transformer.

In what follows, we first develop High-dimensional Positional Encoding (HPE), which is beneficial for both ABS and REF stages. Thereafter, we leverage HPE to develop HPENets, which are conveniently designed suite of networks for MLP based point cloud processing. Particularly unique to our models is the inter-set OPeration sub-stage in the REF component, which also distinguishes our technique from the transformer based methods that employ a REF stage, e.g., Point Transformer (Zhao et al. 2021).

High-Dimensional Positional Encoding

Positional information is *the* most important feature of points clouds. It encodes robust geometric details of a scene. Hence, we propose to leverage it fully in both ABS and REF stages of point cloud modeling using explicit positional encoding (PE). The notion of PE originated in the transformer literature (Vaswani et al. 2017). In the point cloud context, PE can encode a point coordinate $p_m = [p^x_m, p^y_m, p^z_m] \in \mathbb{R}^{1\times 3}$ into the space of corresponding feature $f_m \in \mathbb{R}^{1\times c}$ to embed geometric information. For a transformer based neural architecture for 3D modeling, sinusoidal PE (PE_{SIN}) and learnable PE (PE_{MLP}) can be formulated as below.

$$PE_{SIN}\begin{cases} (p_m, 6i+0) = \sin\left(100 p^x_m / 1000^{6i/c}\right) \\ (p_m, 6i+1) = \cos\left(100 p^x_m / 1000^{6i/c}\right) \\ (p_m, 6i+2) = \sin\left(100 p^y_m / 1000^{6i/c}\right) \\ (p_m, 6i+3) = \cos\left(100 p^y_m / 1000^{6i/c}\right) \\ (p_m, 6i+4) = \sin\left(100 p^z_m / 1000^{6i/c}\right) \\ (p_m, 6i+5) = \cos\left(100 p^z_m / 1000^{6i/c}\right) \end{cases}, \tag{6}$$

$$PE_{MLP}\left(p_m\right) = \theta_{3,c}\left(Norm\left(\delta_{3,3}\left(p_m\right)\right)\right). \tag{7}$$

In the above equations, $i = c/6$ is the index of the subgroup PE vector. The θ and δ denote MLP-based transformations, with subscripts denoting the channel dimensions

of their input and output. $Norm$ denotes the normalization, e.g., batch/layer normalization for restricting the PE to [0, 1]. The sine and cosine functions in PE_{SIN} inherently restrict the values in [-1, 1]. Though potentially useful, both PE_{SIN} and PE_{MLP} provide low-dimensional encodings, which is inadequate to effectively capture the complex geometric relations among the points of unstructured point clouds. Moreover, PE_{SIN} is not adaptive.

To overcome the inadequacy, we propose a High-dimensional Positional Encoding (HPE) module. Our module first transforms the point coordinates to a high-dimension space for a more comprehensive encoding of geometric details. Then, it employs an MLP to align the high-dimensional encoding with the feature space, which also makes its use flexible. We propose methods for generating the high-dimensional codes using sinusoidal and learnable encoding, termed HPE_{SIN} and HPE_{MLP}.

Our HPE_{SIN} uses sine and cosine functions to extend the channel dimensions from 3 to $(\lfloor c/6 \rfloor \times 6)$ to get a high-dimensional vector, followed by an MLP to align the vector to the feature space. Following the notational conventions from above, HPE_{SIN} can be formulated as

$$HPE_{SIN}\left(p_m\right) = \theta_{(\lfloor c/6 \rfloor \times 6),c}\left(PE_{SIN}\left(p_m\right)\right). \quad (8)$$

Our HPE_{MLP} generates the high-dimensional vector in a data-driven manner. Specifically, it uses an MLP to extend channel dimensions from 3 to c and then uses an MLP to transform the high-dimensional vector, formulated as

$$HPE_{MLP}\left(p_m\right) = \theta_{c,c}\left(Norm\left(\delta_{3,c}\left(p_m\right)\right)\right). \quad (9)$$

The channel dimensions of the high-dimensional vectors in our encoding can be any suitable value. In our approach, we pack the encoding scheme in HPE(SIN) and HPE(MLP) modules, as shown in Fig. 1. These module are readily usable for the ABS and REF stages of MLP based networks.

HPENets for Point Cloud Processing

Based on our "ABS-REF" view and HPE module, we develop MLP point cloud processing networks, termed HPENets. To explain, we focus on the more comprehensive encoder-decoder architecture for the semantic segmentation task, as shown in Fig. 1. Other networks are easily deduced from this explanation. In Fig. 1, the encoder consists of a single point embedding layer and four blocks that follow "ABS-REF" view, while incorporating the proposed HPE modules. The point embedding layer is used to enrich the input representation. We denote the channels of point embedding layer as C_e, which can be varied. The number of REF layers can also vary in the ABS-REF blocks for different tasks. We denote these numbers by a set B that consists of four elements. To exemplify, our HPENet applied for the segmentation task on S3DIS (Armeni et al. 2016) can use B = [3,6,3,3], which means the number of REF layers in the four "ABS-REF" blocks are 3, 6, 3 and 3, respectively. The value in B can decrease to 0 to degenerate HPENet into a single-stage method, e.g., for object classification.

As shown in Fig. 2, in the ABS stage, we introduce our HPE module after the grouping layer, which uses the grouped points set $\mathcal{K}_{ABS}^{l+1}$ as the inputs. We first use the grouped feature set $\mathcal{D}_{ABS}^{l+1}$ to add high-dimensional positional encodings and then follow it by a concatenation of grouped points set as the input of MLPs. We opt a similar strategy in the REF stage. As illustrated in Fig. 1, the obvious difference between the ABS and REF stages is the existence of the sampling layer and the design of local aggregation operation (MLPs). In ABS, the MLPs are used before the Symmetric OPeration (SOP), as they aim to aggregate the local features. In contrast, the SOP is embedded between the MLPs in the REF stage. Specifically, the MLP before the SOP pays attention to capturing inter-set context information, while the MLPs following SOP focus on refining the point-wise features. By varying the hyper-parameters B and C_e, we conveniently construct a range of "HPENets" with different model sizes to match the training data scales. We develop HPENets with the following configurations in our experiments.

- ScanObjectNN: C_e = 32, B = [0, 0, 0 ,0].
- ModelNet40: C_e = 64, B = [0, 0, 0 ,0].
- ShapeNetPart: C_e = 160, B = [0, 0, 0 ,0].
- S3DIS: C_e = 64, B = [3, 6, 3 ,3].
- ScanNet V2: C_e = 64, B =[5, 8, 5 ,5].

Method	Params.(M)	OA(%)	mAcc(%)
PointNet (Qi et al. 2017a)	3.5	68.2	63.4
PointNet++ (Qi et al. 2017b)	1.5	77.9	75.4
DGCNN (Wang et al. 2019)	1.8	78.1	73.6
PointMLP (Ma et al. 2022)	12.6	85.7	84.4
PointNeXt (Qian et al. 2022)	1.4	88.2	86.8
PointMetaBase (Lin et al. 2023)	1.4	88.2	86.8
HPENet(SIN)	1.7	<u>88.4</u>	<u>86.9</u>
HPENet(MLP)	1.7	**88.9**	**87.6**

Table 1: 3D object classification on ScanObjectNN (Uy et al. 2019). The best and second-best results are boldfaced and underlined, respectively.

Experiments

Our technique is extensively evaluated on five datasets for four different tasks of object classification, object part segmentation, semantic segmentation and object detection.

3D Object Classification

ScanObjectNN (Uy et al. 2019) collects real-world objects from 700 unique scenes of the SOTA mesh datasets SceneNN (Hua et al. 2016) and ScanNet (Dai et al. 2017). It contains about 15,000 real scanned objects, categorized into 15 classes with 2,902 unique object instances. Because of occlusions and noise, ScanObjectNN is a highly challenging dataset for the current methods. Following Ma et al. (2022), we evaluate HPENet on PB_T50_RS, the hardest and most commonly used variant of ScanObjectNN, using the standard metrics of mean accuracy (mAcc) and overall accuracy (OA). As reported in Tab. 1, HPENet outperforms the existing techniques and HPENet(MLP) achieves the SOTA performance with 88.9% OA and 87.6% mAcc. The HPENet(MLP) outperforms the existing best MLP-based method PointNeXt (Qian et al. 2022) by 0.7% OA and

Method	S3DIS Area-5			S3DIS 6-fold			ScanNet V2
	OA	mAcc	mIoU	OA	mAcc	mIoU	Val mIoU
PointNet (Qi et al. 2017a)	-	49.0	41.1	78.5	66.2	47.6	-
PointNet++ (Qi et al. 2017b)	83.0	-	53.5	81.0	-	54.5	53.5
KPConv (Thomas et al. 2019)	-	72.8	67.1	-	79.1	70.6	69.2
Point Transformer (Zhao et al. 2021)	90.8	76.5	70.4	90.2	81.9	73.5	70.6
Stratified Transformer (Lai et al. 2022)	**91.5**	78.1	72.0	-	-	-	**74.3***
PointNeXt (Qian et al. 2022)	91.0	77.2	71.1	90.3	83.0	74.9	71.5
PointMetaBase (Lin et al. 2023)	91.3	78.0	72.3	91.3	-	77.0	72.8
HPENet(SIN)	91.0	**78.9**	72.4	91.7	86.1	78.2	72.5
HPENet(MLP)	**91.5**	78.5	**72.7**	**91.9**	**86.2**	**78.7**	74.0*

Table 2: 3D semantic segmentation results on S3DIS and ScanNet V2. For ScanNet V2 results are on validation set. *Stratified Transformer requires 211 hours training while requiring 120,000 point input to achieve the results. Our HPENet(MLP) needs only 82 hours training and uses 64,000 point input.

Method	Cls. mIoU	Ins. mIoU
Point Transformer (Zhao et al. 2021)	83.7	86.6
Stratified Transformer (Lai et al. 2022)	85.1	86.6
PointNeXt-S (Qian et al. 2022)	85.2	87.0
HPENet(SIN)	**85.5**	**87.1**
HPENet(MLP)	85.3	87.0

Table 3: 3D object part segmentation on ShapeNetPart.

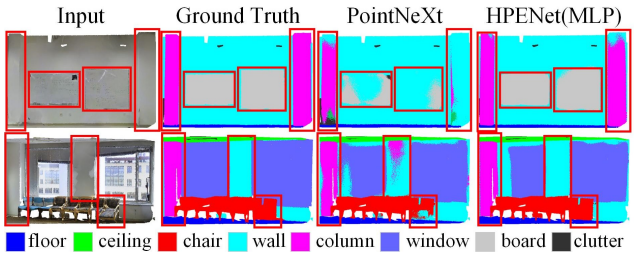

Figure 3: Representative qualitative results of HPENet (MLP) and the strong MLP-based method PointNeXt (Qian et al. 2022) on S3DIS Area-5.

0.8% mAcc, which indicated that HPE is effective for MLP-based point cloud processing. It is emphasized that we do not employ any pre-training or voting strategies to outperform the current SOTA methods. In the table, we also report the model sizes as parameters in millions. It is notable that our model sizes are also on the lower side of the spectrum.

ModelNet40 (Wu et al. 2015) is a widely popular dataset for synthetic object classification with standard evaluation protocols. Our HPENet(SIN) variant equals the SOTA performance of 91.3 mAcc on this dataset with PointMLP (Ma et al. 2022). Moreover, our model achieves these results with only 5.9M parameters as compared to 12.6M parameters of PointMLP.

3D Object Part Segmentation

ShapeNetPart (Yi et al. 2016) is an object-level dataset for object part segmentation, consisting of 16,881 objects with 16 shape categories belonging to 50 parts labels. Following Qi et al. (2017b), we randomly select 2,048 points as input and use class mean IoU (Cls. mIoU) and instance mean IoU (Ins. mIoU) for evaluation. In Tab. 3, we report the results of the top performing approaches. Our method outperforms the SOTA method on this dataset as well. Notably, HPENet also outperforms the strong transformer-based method Stratified Transformer (Lai et al. 2022).

3D Semantic Segmentation

Semantic segmentation aims to assign a semantic label to each point in scene point clouds. In general, this task is much more challenging than object classification. We evaluate HPENet on two popular large-scale datasets, S3DIS (Armeni et al. 2016) and ScanNet (Dai et al. 2017). The results are summarised in Tab. 2. We discuss them below.

S3DIS (Armeni et al. 2016) comprises 6 large-scale indoor areas and 271 rooms, which are captured from 3 different buildings. In total, 273 million points are annotated and classified into 13 semantic categories. Following PointNeXt (Qian et al. 2022), we use two evaluation protocols. The first uses Area-5 as the test scene and all other scenes for training and the second strategy is the standard 6-fold cross-validation. For evaluation, we use the popular metrics of mean IoU (mIoU), mAcc, and OA. From Tab. 2, it can be observed that HPENet establishes new state-of-the-art performances of 72.7% mIoU on S3DIS Area-5 and 78.7% mIoU on S3DIS (6-fold cross-validation). Again, we do not use any pre-training or voting strategies to gain performance boost in our results.

Despite being an MLP-based approach, HPENet performs at par or better than transformer methods. Our HPENet outperforms PointNext - the strong MLP-based method - by absolute gains of 0.5%, 1.7%, and 1.6% in terms of OA, mAcc, and mIoU on the Area-5 test; and by 1.6%, 3.2%, and 3.8% in term of OA, mAcc, and mIoU for the 6-fold experiments, respectively. We provide a representative example of qualitative results for our method on S3DIS in Fig. 3 along with the strong MLP based method, PointNeXt.

ScanNet V2 (Dai et al. 2017) consists of 3D indoor scenes with 2.5 million RGB-D frames in more than 1,500 scans, annotated with 20 semantic classes. We follow the standard training and validation splits of 1,201 and 312 scenes, respectively. As shown in the last column of Tab. 2, HPENet

Method	mAP@0.25	mAP@0.5
VoteNet (Qi et al. 2019)	63.8	44.2
3DETR (Misra, Girdhar, and Joulin 2021)	65.0	47.0
GroupFree3D (Liu et al. 2021b)	68.2	52.6
VoteNet + HPE(MLP)	65.0	45.6
GroupFree3D + HPE(MLP)	**69.1**	**53.0**

Table 4: 3D object detection on ScanNet V2. VoteNet and GroupFree3D use MMDetection3D (Contributors 2020).

Networks(C_e, B)	HPE	Param.	mIoU	△	TP
HPENet_dv(32,[0,0,0,0])		0.8M	64.2	-	232
HPENet_dv(32,[0,0,0,0])	SIN	0.9M	65.0	+0.8	199
HPENet_dv(32,[0,0,0,0])	MLP	0.9M	65.3	+1.1	205
HPENet_dv(32,[1,1,1,1])		3.7M	66.7	+2.5	161
HPENet_dv(32,[1,1,1,1])	SIN	4.1M	69.3	+5.1	115
HPENet_dv(32,[1,1,1,1])	MLP	4.1M	69.9	+5.7	125
HPENet_dv*		3.7M	63.7	-0.4	228

Table 5: Ablation study on S3DIS Area-5 demonstrating efficacy of ABS-REF view, and contribution of HPE modules. △ is increment from previous row. TP denotes throughput in instance/second.

achieves highly competitive performance of 74.0% mIoU which outperforms PointNeXt by 2.5% mIoU. According to the released files of the best transformer-based method Stratified Transformer, HPENet uses less than half of the training time (211 hours vs 82 hours) but still achieves comparable performance. The reported performance of HPENet is achieved with 64,000 points, whereas the Stratified Transformer requires 120,000 points to achieve these results.

3D Object Detection

The key building block of HPENet, i.e., HPE module is inherently compatible to MLP based backbones. To demonstrate its flexibility, we also extend the competitive techniques of VoteNet (Qi et al. 2019) and GroupFree3D (Liu et al. 2021b) with HPE(MLP). In Tab. 4, we summarize the results of our extension following the standard evaluation protocols on ScanNet V2 dataset (Dai et al. 2017). A consistent across the board gain is achieved with our HPE(MLP) extension.

Ablation & Further Discussion

ABS-REF efficacy: In Tab. 5, we establish the contribution of REF stage in our HPENet that follows ABS-REF paradigm. By removing the REF stage, HPENet degenerates to a single-stage method. We call this degenerated version HPENet-dv in the table. We chose HPENet-dv as the baseline and expanded it by adding a REF behind each ABS. This obtained 2.5% mIoU performance gain. Further using our HPE schemes, we eventually achieve a performance of 69.9% mIoU, which is already comparable to 70.4% mIoU of Point Transformer (PT). Due to the simple local aggregation strategy used in REF, the size of our model is much smaller (4.1M vs 7.8M) than that of PT. Moreover, our model has 3.6 times better Through Put (TP) than PT. To verify the impact of parameters, we remove the grouping operation in the REF stage of HPENet-dv(32,[1,1,1,1]) to get a model with only the ABS stage and the same number of parameters, termed HPENet-dv*. However, HPENet-dv* only achieves 63.7% mIoU. These results validate that the REF stage is an important component under our ABS-REF view and our HPE effectively supports this view.

Type	ABS	REF	OA	mAcc	$\triangle_{mAcc}$	mIoU	$\triangle_{mIoU}$
HPE_{SIN}			90.7	76.0	-	70.2	-
	√		90.8	77.2	+1.2	71.2	+1.0
		√	90.9	77.4	+1.4	71.4	+1.2
	√	√	91.0	**78.9**	+2.8	72.4	+2.2
HPE_{MLP}	√		90.8	77.7	+1.7	71.4	+1.2
		√	91.1	77.6	+1.6	70.8	+0.6
	√	√	**91.5**	78.5	+2.5	**72.7**	+2.5
PE_{MLP}	√	√	90.9	76.8	+0.8	71.2	+1.0
$HPE_{SIN}(mul)$	√	√	90.8	76.9	+0.9	70.8	+0.6
$HPE_{SIN}(abs)$	√	√	89.7	75.3	-0.7	69.1	-1.1

Table 6: Ablation study for positional encoding on S3DIS Area-5 justifying HPE use in both ABS and REF stages.

Dimension	3	c//8	c//4	c//2	c
mIoU	71.2	71.6	71.9	72.0	72.7

Table 7: Ablation study on dimension of HPE(MLP) on S3DIS Area-5. 'c' denotes feature channel number.

More on positional encoding: In Tab. 6, we evaluate the influence of different positional encodings in different stages of HPENet on S3DIS Area-5. We use the high-dimensional positional encoding (HPE_{MLP} and HPE_{SIN}) and learnable positional encoding (PE_{MLP}). In the experiments, we also study the effect of absolute positional encoding by replacing the input of HPE_{SIN} with absolute point coordinates, named $HPE_{SIN}(abs)$. Moreover, we replace the regular element-wise addition with element-wise multiplication $HPE_{SIN}(mul)$, which treats the positional encoding as a dynamic feature weight. These results clearly justify the proposed HPE_{MLP} and HPE_{SIN}. Moreover, these results support our unique idea that both ABS and REF should use positional encoding. In Tab. 7, we analyze the effects of dimension variation of high-dimensional projected space with HPE(MLP) on S3DIS Area-5. The results indicate that high-dimensional representation is crucial for position encoding.

Conclusion

Inspired by the distinct subsampling and convolution stages in image processing models, we provide a two-stage "abstraction and refinement" (ABS-REF) view for point cloud neural processing. This view allows an intuitive delineation of the key strengths of the existing methods. We also propose a high-dimensional positional encoding (HPE) scheme that is compatible with the "ABS-REF" paradigm. Based on ABS-REF view and HPE, we devise a suite of HPENets that leverage HPE for MLP based modeling for object classification, object part segmentation, semantic segmentation and object detection, mostly improving SOTA performance across the board.

Acknowledgments

This work was supported by the NSFC (61973106, U2013203,62103137, U1913202, U21A20487); the Natural Science Fund of Hunan Province (2021JJ10024, 2022JJ30024, 2022JJ40100); the Key Research and Development Project of Science and the Technology Plan of Hunan Province(2022GK2014).

References

Armeni, I.; Sener, O.; Zamir, A. R.; Jiang, H.; Brilakis, I.; Fischer, M.; and Savarese, S. 2016. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1534–1543.

Choe, J.; Park, C.; Rameau, F.; Park, J.; and Kweon, I. S. 2022. Pointmixer: Mlp-mixer for point cloud understanding. In *European Conference on Computer Vision*, 620–640. Springer.

Choy, C.; Gwak, J.; and Savarese, S. 2019. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3075–3084.

Contributors, M. 2020. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d. Accessed: 2023-04-07.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5828–5839.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Engelmann, F.; Kontogianni, T.; and Leibe, B. 2020. Dilated point convolutions: On the receptive field size of point convolutions on 3d point clouds. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 9463–9469. IEEE.

Guo, M.-H.; Cai, J.-X.; Liu, Z.-N.; Mu, T.-J.; Martin, R. R.; and Hu, S.-M. 2021. Pct: Point cloud transformer. *Computational Visual Media*, 7(2): 187–199.

Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; and Markham, A. 2020. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11108–11117.

Hua, B.-S.; Pham, Q.-H.; Nguyen, D. T.; Tran, M.-K.; Yu, L.-F.; and Yeung, S.-K. 2016. Scenenn: A scene meshes dataset with annotations. In *2016 fourth international conference on 3D vision (3DV)*, 92–101. Ieee.

Huang, J.; and You, S. 2016. Point cloud labeling using 3d convolutional neural network. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, 2670–2675. IEEE.

Jiang, M.; Wu, Y.; Zhao, T.; Zhao, Z.; and Lu, C. 2018. Pointsift: A sift-like network module for 3d point cloud semantic segmentation. *arXiv preprint arXiv:1807.00652*.

Kong, D.; Wu, F.; and Saroglou, C. 2020. Automatic identification and characterization of discontinuities in rock masses from 3D point clouds. *Engineering Geology*, 265: 105442.

Lai, X.; Liu, J.; Jiang, L.; Wang, L.; Zhao, H.; Liu, S.; Qi, X.; and Jia, J. 2022. Stratified Transformer for 3D Point Cloud Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8500–8509.

Landrieu, L.; and Simonovsky, M. 2018. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4558–4567.

Lei, H.; Akhtar, N.; and Mian, A. 2019. Octree guided cnn with spherical kernels for 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9631–9640.

Lei, H.; Akhtar, N.; and Mian, A. 2020. Spherical kernel for efficient graph convolution on 3d point clouds. *IEEE transactions on pattern analysis and machine intelligence*, 43(10): 3664–3680.

Li, R.; Li, J.; Wang, J.; Wu, Q.; and Liu, X. 2022. Dual-view 3D object recognition and detection via Lidar point cloud and camera image. *Robotics and Autonomous Systems*, (150-): 150.

Lian, D.; Yu, Z.; Sun, X.; and Gao, S. 2021. As-mlp: An axial shifted mlp architecture for vision. *arXiv preprint arXiv:2107.08391*.

Lin, H.; Zheng, X.; Li, L.; Chao, F.; Wang, S.; Wang, Y.; Tian, Y.; and Ji, R. 2023. Meta Architecture for Point Cloud Analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17682–17691.

Liu, Z.; Hu, H.; Cao, Y.; Zhang, Z.; and Tong, X. 2020. A closer look at local aggregation operators in point cloud analysis. In *European Conference on Computer Vision*, 326–342. Springer.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021a. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022.

Liu, Z.; Zhang, Z.; Cao, Y.; Hu, H.; and Tong, X. 2021b. Group-free 3d object detection via transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2949–2958.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60: 91–110.

Ma, X.; Qin, C.; You, H.; Ran, H.; and Fu, Y. 2022. Rethinking network design and local geometry in point

cloud: A simple residual mlp framework. *arXiv preprint arXiv:2202.07123*.

Misra, I.; Girdhar, R.; and Joulin, A. 2021. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2906–2917.

Park, C.; Jeong, Y.; Cho, M.; and Park, J. 2022. Fast Point Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16949–16958.

Qi, C. R.; Litany, O.; He, K.; and Guibas, L. J. 2019. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, 9277–9286.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

Qian, G.; Li, Y.; Peng, H.; Mai, J.; Hammoud, H.; Elhoseiny, M.; and Ghanem, B. 2022. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 35: 23192–23204.

Shen, Y.; Feng, C.; Yang, Y.; and Tian, D. 2018. Mining point cloud local structures by kernel correlation and graph pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4548–4557.

Shi, H.; Wei, J.; Li, R.; Liu, F.; and Lin, G. 2022. Weakly supervised segmentation on outdoor 4d point clouds with temporal matching and spatial graph propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11840–11849.

Tang, Y.; Han, K.; Guo, J.; Xu, C.; Li, Y.; Xu, C.; and Wang, Y. 2022. An image patch is a wave: Phase-aware vision mlp. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10935–10944.

Thomas, H.; Qi, C. R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; and Guibas, L. J. 2019. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6411–6420.

Tolstikhin, I. O.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. 2021. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34: 24261–24272.

Uy, M. A.; Pham, Q.-H.; Hua, B.-S.; Nguyen, T.; and Yeung, S.-K. 2019. 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*, 1588–1597.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5): 1–12.

Wang, Z.; Jiang, W.; Zhu, Y. M.; Yuan, L.; Song, Y.; and Liu, W. 2022. Dynamixer: a vision mlp architecture with dynamic mixing. In *International Conference on Machine Learning*, 22691–22701. PMLR.

Wu, W.; Qi, Z.; and Fuxin, L. 2019. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9621–9630.

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1912–1920.

Xiang, T.; Zhang, C.; Song, Y.; Yu, J.; and Cai, W. 2021. Walk in the cloud: Learning curves for point clouds shape analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 915–924.

Xu, M.; Ding, R.; Zhao, H.; and Qi, X. 2021. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3173–3182.

Yi, L.; Kim, V. G.; Ceylan, D.; Shen, I.-C.; Yan, M.; Su, H.; Lu, C.; Huang, Q.; Sheffer, A.; and Guibas, L. 2016. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6): 1–12.

Yu, X.; Tang, L.; Rao, Y.; Huang, T.; Zhou, J.; and Lu, J. 2022. 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*, 19313–19322.

Zhang, C.; Wan, H.; Shen, X.; and Wu, Z. 2022. PatchFormer: An Efficient Point Transformer With Patch Attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11799–11808.

Zhao, H.; Jiang, L.; Jia, J.; Torr, P. H.; and Koltun, V. 2021. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16259–16268.

Zheng, W.; Tang, W.; Jiang, L.; and Fu, C.-W. 2021. SE-SSD: Self-ensembling single-stage object detector from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14494–14503.