Improved MLP Point Cloud Processing with High-Dimensional Positional Encoding

Yanmei Zou¹, Hongshan Yu¹*, Zhengeng Yang²*, Zechuan Li¹, Naveed Akhtar³
¹College of Electrical and Information Engineering, Quanzhou Innovation Institute, Hunan University, Changsha, China
²College of Engineering and Design, Hunan Normal University, Changsha, China
³School of Computing and Information Systems, The University of Melbourne, 3052 Victoria, Australia

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 Saragoglou 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...
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. 2022; Dai 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 $\mathcal{P} = \{p_m \in \mathbb{R}^{1 \times 3} \}_{m=1}^N$ and the feature set $\mathcal{F} = \{f_m \in \mathbb{R}^{1 \times c} \}_{m=1}^N$, 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 $\mathcal{K} = \{k_m \in \mathbb{R}^{k \times 3} \}_{m=1}^N$ and the corresponding feature sets $\mathcal{D} = \{d_m \in \mathbb{R}^{k \times c} \}_{m=1}^N$.

**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 $\mathcal{P}$ and its corresponding feature set $\mathcal{F}$, we get the point set $\mathcal{P}^{l+1}$, grouped point sets $\mathcal{K}^{l+1}$, and feature sets $\mathcal{D}^{l+1,ABS}$ 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

$$\mathcal{P}^{l+1} = \text{FPS} (\mathcal{P}^l), \quad \mathcal{P}^{l+1}_m \in \mathcal{P}^{l+1},$$

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

$$f^{l+1}_m = h_{ABS}(\mathcal{D}^{l+1,ABS}(\mathcal{P}^{l+1}_m), \mathcal{K}^{l+1}(\mathcal{P}^{l+1}_m)),$$

where $\mathcal{D}^{l+1,ABS}(\mathcal{P}^{l+1})_m$ and $\mathcal{K}^{l+1}(\mathcal{P}^{l+1})_m$ are the neighbor feature and point sets of the centroid $\mathcal{P}^{l+1}_m$, 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
cloud networks. This stage aims to refine the centroid features by gathering local context information. Specifically, the REF stage further processes the point set $F^{l+1}$ and features set $F_{ABS}^{l+1}$ 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

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

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

where $D^{l+1}_{REF}(p^{l+1}_m)$ and $K^{l+1}_{REF}(p^{l+1}_m)$ 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 HPE-Net, 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 ($P_{ESIN}$) and learnable PE ($P_{EMLP}$) can be formulated as below.

$$P_{ESIN} = \begin{pmatrix}
(p_m, 6i + 0) = \sin (100p^x_m/1000) \\
(p_m, 6i + 1) = \cos (100p^x_m/1000) \\
(p_m, 6i + 2) = \sin (100p^y_m/1000) \\
(p_m, 6i + 3) = \cos (100p^y_m/1000) \\
(p_m, 6i + 4) = \sin (100p^z_m/1000) \\
(p_m, 6i + 5) = \cos (100p^z_m/1000)
\end{pmatrix},$$ (6)

$$P_{EMLP}(p_m) = \theta_{3c}(\text{Norm}(\delta_{3c}(p_m))).$$ (7)
of their input and output. \textit{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-dimensional 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 \(\lceil c/6 \rceil \times 6\) 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

\[
HPE_{SIN}(p_m) = \theta(\lceil c/6 \rceil \times 6,c \cdot (PE_{SIN}(p_m))) . \tag{8}
\]

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

\[
HPE_{MLP}(p_m) = \theta_{c,c}(Norm(\delta_{3,c}(p_m))) . \tag{9}
\]

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.

**HPE Nets for Point Cloud Processing**

Based on our “ABS-REF” view and HPE module, we develop MLP point cloud processing networks, termed HPE Nets. 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 \(K_{ABS}^{i+1}\) as the inputs. We first use the

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

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

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

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

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP@0.25</th>
<th>mAP@0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>VoteNet (Qi et al. 2019)</td>
<td>63.8</td>
<td>44.2</td>
</tr>
<tr>
<td>3DETR (Misra, Girdhar, and Joulin 2021)</td>
<td>65.0</td>
<td>47.0</td>
</tr>
<tr>
<td>GroupFree3D (Liu et al. 2021b)</td>
<td>68.2</td>
<td>52.6</td>
</tr>
<tr>
<td>VoteNet + HPE(MLP)</td>
<td>65.0</td>
<td>45.6</td>
</tr>
<tr>
<td>GroupFree3D + HPE(MLP)</td>
<td>69.1</td>
<td>53.0</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Dimension</th>
<th>3</th>
<th>c/8</th>
<th>c/4</th>
<th>c/2</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>mIoU</td>
<td>71.2</td>
<td>71.6</td>
<td>71.9</td>
<td>72.0</td>
<td>72.7</td>
</tr>
</tbody>
</table>

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

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.

**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, U2013202, U21A10487); the Natural Science Fund of Hunan Province (2021JJ10024, 2021JJ30024, 2022JJ40100); the Key Research and Development Project of Science and the Technology Plan of Hunan Province (2022GK2014).

References


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


