PointListNet: Deep Learning on 3D Point Lists

Hehe Fan[1,2] Linchao Zhu[1] Yi Yang[1] Mohan Kankanhalli[2]

[1]Zhejiang University [2]National University of Singapore

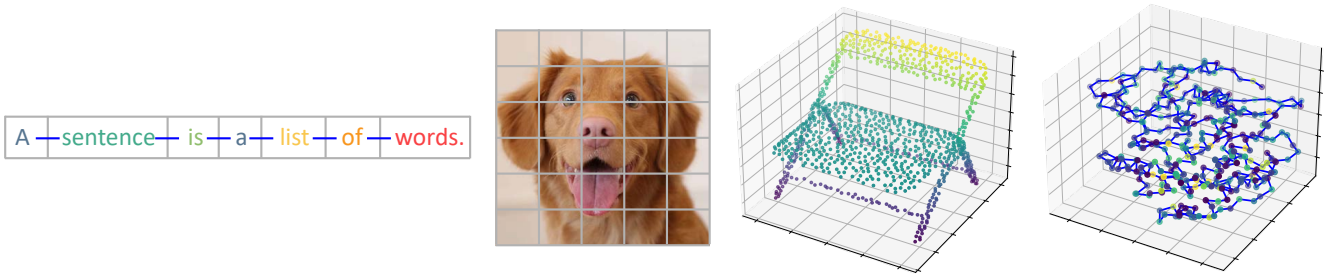

Figure 1. Data structure comparison of text, image, point cloud and protein. **(a)** Texts are regular 1D lists of words. The position is the word's sequential order in the text and the feature is the word itself. **(b)** Images are regular 2D grids of pixels. The position is the row and column where the pixel is located and the feature is the color. **(c)** Point clouds are irregular 3D point sets. The position is the 3D coordinate and the feature is the point attributes. **(d)** Proteins can be seen as 3D point lists. The position of an amino acid involves a regular 1D sequential order and an irregular 3D coordinate. The feature is the amino acid (residue) type.

Abstract

*Deep neural networks on regular 1D lists (*e.g.*, natural languages) and irregular 3D sets (*e.g.*, point clouds) have made tremendous achievements. The key to natural language processing is to model words and their regular order dependency in texts. For point cloud understanding, the challenge is to understand the geometry via irregular point coordinates, in which point-feeding orders do not matter. However, there are a few kinds of data that exhibit both regular 1D list and irregular 3D set structures, such as proteins and non-coding RNAs. In this paper, we refer to them as 3D point lists and propose a Transformer-style PointListNet to model them. First, PointListNet employs non-parametric distance-based attention because we find sometimes it is the distance, instead of the feature or type, that mainly determines how much two points, e.g., amino acids, are correlated in the micro world. Second, different from the vanilla Transformer that directly performs a simple linear transformation on inputs to generate values and does not explicitly model relative relations, our PointListNet integrates the 1D order and 3D Euclidean displacements into values. We conduct experiments on protein fold classification and enzyme reaction classification. Experimental results show the effectiveness of the proposed PointListNet.*

1. Introduction

The essence of deep learning is to capture the structure of a certain kind of data via artificial neural networks. Usually, an element of data includes a position part and a feature part. According to the type of element position, data exhibit different structures. Various deep neural networks are proposed to model those structures and have made tremendous achievements.

For example, texts are 1D lists of words. As shown in Fig. 1(a). The position of a word is its order in the text and the feature is the word itself. To capture the structure of texts or the dependency of words, 1D convolutional neural networks (CNNs) [3, 30, 58], recurrent neural networks (RNNs) [9, 26, 39] and Transformers [13, 49] are widely used. A digital image can be seen as a 2D rectangular grid or matrix of pixels, as shown in Fig. 1(b). Each pixel has a 2D position and is associated with a feature of color or other attributes. In this case, 2D CNNs are usually used to model image structure [23, 33, 46]. Recently, Transformers are also employed for image understanding [15].

Recently, 3D point cloud/set processing is attracting

more and more attention from the deep learning community. Different from texts or images, in which the orders of words or the positions of pixels are regular (words or pixels are distributed uniformly in texts or images), the 3D coordinates of points are irregular (points are distributed unevenly in 3D Euclidean space), as shown in Fig. 1(c). To capture the irregular structure of point clouds, deep neural networks, such as multilayer perceptrons (MLPs) [42, 43, 45], convolutions [48, 56] and Transformers [22, 62], need to not only effectively exploit 3D coordinates for geometry understanding but also be invariant to permutations of the input set in point-feeding order.

Besides regular 1D lists of words, 2D grids of pixels and irregular 3D point sets, data may exhibit hybrid structures. For example, proteins are made up of amino acids. As shown in Fig. 1(d), those amino acids are linked by peptide bonds and form a chain. Therefore, proteins include a 1D list data structure. Because amino acids are arranged uniformly in the chains, the list structure is regular. In addition to the 1D sequential order in the peptide chain, each amino acid is with a 3D coordinate, which specifies its spatial position in the protein. Those 3D coordinates describe a geometry structure. Similar to point clouds, the geometry structure of proteins exhibits irregularity. Therefore, the data structure of proteins involves a regular 1D list and an irregular 3D set. In this paper, we refer to this data structure as 3D point list. Point lists also exist in other polymers, such as non-coding RNAs. Because the function of proteins or non-coding RNAs is based on their structures, modeling 3D point lists can facilitate a mechanistic understanding of their function to life.

In this paper, we propose a Transformer-style network, named PointListNet, to capture the structure of 3D point lists. First, different from the vanilla Transformer [15, 49], which calculates self-attention by performing computationally expensive matrix multiplication on inputs, our PointListNet employs a simple non-parametric distance-based attention mechanism because we find sometimes it is mainly the distance, instead of the feature or type, that determines how much two elements, *e.g.*, amino acids, are correlated in the micro world. Second, because structures are relative, which is independent of the absolute sequential order or the absolute Euclidean coordinate, our PointListNet integrates the 1D order and 3D Euclidean displacements into values. This is substantially different from the vanilla Transformer that directly performs a simple linear transformation on absolute positional embeddings and input features to generate values, which does not explicitly model relative distance or direction. To evaluate PointListNet, we conduct experiments on protein fold classification and enzyme reaction classification and achieve new state-of-the-art accuracy. The contributions of this paper are fivefold:

- Among the early efforts, we investigate a range of point cloud methods for protein modeling.
- We propose a Transformer-style network, *i.e.*, PointListNet, for 3D point list modeling.
- We replace self-attention with non-parametric distance-based attention, which is more efficient and effective to achieve the correlation among microparticles in some cases.
- We integrate relative structure modeling into Transformer and employ regular and irregular methods to capture the sequence and geometry structures, respectively.
- We conduct extensive experiments on two protein tasks and the proposed method significantly outperforms existing methods.

2. Related Work

Deep Learning on 3D Point Sets. Deep learning on point sets/clouds has been widely investigated in several problems, including shape classification, object part segmentation, scene semantic segmentation, reconstruction and object detection [8, 10, 17, 19, 22, 36–38, 41–43, 48, 53, 56, 59]. Most recent works aim at directly manipulating 3D points without transforming coordinates into regular voxel grids. Since a point cloud is essentially a set of unordered points and invariant to permutations of its points, deep learning on point clouds mainly focuses on designing effective operations that do not rely on point orders. Because point cloud methods do not involve sequence modeling, directly applying them to 3D point lists, *e.g.*, proteins, may lead to inferior accuracy.

Deep Learning on Proteins. Proteins exhibit multi-level structures. Deep-learning-based methods for protein representation learning mainly focus on the 1D primary and the 3D tertiary structures understanding. The primary structure refers to the sequence of amino acids in the polypeptide chain. The tertiary structure refers to the three-dimensional structure created by a single protein molecule (a single polypeptide chain). For the primary structure, because acids in polypeptide chains can be seen as words in sentences, approaches for natural language processing can be used for sequence-based protein representation learning [1, 5, 27, 34, 35, 44, 44, 47]. For the tertiary structure, the 3D geometric information of amino acids or atoms is used to enhance protein representation [2, 4, 7, 12, 21, 25, 28, 29, 54, 60]. Different from these methods, we propose a Transformer-style method to model primary and tertiary structures for proteins. Moreover, we employ different approaches to capture the 1D and 3D structures.

Transformer. Impressive progress has been made on natural language processing due to the success of Transformer

networks [11, 13, 49, 57]. In computer vision, the community has used self-attention or Transformer to model images in a non-local manner [6, 16, 32, 52, 61]. In particular, Zhao *et al*. proposed a Point Transformer [62] to model point clouds. Fan *et al*. proposed a P4Transformer [18] for point cloud video understanding. Lai *et al*. proposed a Stratified Transformer [36] for point cloud segmentation. Feng *et al*. proposed a Structure Embedding Transformer (SEFormer) [20] for 3D object detection. Wang *et al*. proposed a Relation-Enhanced Transformer [51] for text-based point cloud localization. Inspired by these methods, we propose a Transformer-style PointListNet for 3D point list modeling. Different from these methods, we replace learning-based self-attention with rule-based distance-attention, thus more efficient to achieve the correlation among microparticles. Moreover, we integrate relative structure modeling into Transformer and employ regular and irregular methods to capture the sequence and geometry structures, respectively.

3. Proposed Point List Network

In this section, because our method is inspired by Transformer, we first briefly review the vanilla Transformer and discuss its potential limitation for 3D point list modeling. Then, we present the proposed Point List Network (PointListNet) in detail. Finally, we incorporate our PointListNet into deep neural networks to address two protein recognition tasks, *i.e*., protein fold classification and enzyme reaction classification.

3.1. Vanilla Transformer

Transformer has an ability to merge related elements or regions based on their similarities, semantics or relations so that each position has a larger receptive field to collect more information from its related elements or regions. Specifically, suppose $\boldsymbol{F} \in \mathbb{R}^{N\times C}$ is the input features of N positions, where C is the number of feature channels, and $\boldsymbol{T} \in \mathbb{R}^{N\times 1}$ is their positions. As shown in Fig. 2(a), Transformer first integrates absolute positional embedding into the input features,

$$\boldsymbol{I} = \text{Embedding}(\boldsymbol{T}) + \boldsymbol{F}. \tag{1}$$

Second, it performs two individual linear transformations on $\boldsymbol{I}$ to generate queries $\boldsymbol{Q} \in \mathbb{R}^{N\times C''}$ and keys $\boldsymbol{K} \in \mathbb{R}^{N\times C''}$, where C'' is the dimension of queries and keys. Then, the softmax function is applied to the scaled dot-product attention of $\boldsymbol{Q}$ and $\boldsymbol{K}$ to generate the attention weights $\boldsymbol{A} \in \mathbb{R}^{N\times N}$,

$$\boldsymbol{Q} = \boldsymbol{I}\cdot\boldsymbol{W}_q, \quad \boldsymbol{K} = \boldsymbol{I}\cdot\boldsymbol{W}_k, \quad \boldsymbol{A} = \text{softmax}(\frac{\boldsymbol{Q}\cdot\boldsymbol{K}^T}{\sqrt{C''}}), \tag{2}$$

where $\cdot$ is matrix multiplication and $\boldsymbol{W}_q, \boldsymbol{W}_k \in \mathbb{R}^{C\times C''}$. Third, Transformer employs another linear transformation on $\boldsymbol{I}$ to generate values $\boldsymbol{V} \in \mathbb{R}^{N\times C'}$, where C' is the value dimension. Finally, the output is computed as a weighted sum of the values,

$$\boldsymbol{V} = \boldsymbol{I}\cdot\boldsymbol{W}_v, \quad \boldsymbol{O}_t = \sum_{t'=1}^{N} \alpha_{tt'}\boldsymbol{v}_{t'}, \tag{3}$$

where $\boldsymbol{W}_v \in \mathbb{R}^{C\times C'}$ and $\boldsymbol{O} \in \mathbb{R}^{N\times C'}$. The $\alpha_{tt'}$ denotes the attention weight of the t'-th position on the t-th position in $\boldsymbol{A}$ and $\boldsymbol{v}'_t$ denotes the value at the t'-th position in $\boldsymbol{V}$.

The attention weights $\boldsymbol{A}$ can indicate how much two elements are correlated in the input. Based on self-attention and global weighted sum, Transformer is able to adaptively search related elements or regions, thus being flexible to capture the structure in data. However, the vanilla Transformer does not model relative relations, such as direction or distance. Therefore, it may not properly model the 1D sequence and 3D geometry structures in point lists.

3.2. PointListNet

A 3D point list can be represented by 1D sequence orders $\boldsymbol{T} \in \mathbb{R}^{N\times 1}$, 3D geometry coordinates $\boldsymbol{P} \in \mathbb{R}^{N\times 3}$ and the associated features $\boldsymbol{F} \in \mathbb{R}^{N\times C}$. For example, a protein can be seen as a point list because each amino acid in it has a 1D sequential order $t \in \{1, \cdots, N\}$ in the peptide chain and a 3D coordinate $\boldsymbol{p}_t \in \mathbb{R}^{1\times 3}$ that specifies its spatial location and a feature, *e.g*., the amino acid type or other attributes $\boldsymbol{f}_t \in \mathbb{R}^{1\times C}$. Inspired by Transformer, we propose a PointListNet for 3D point list modeling.

3.2.1 Non-parametric Distance-based Attention

When employing Transformer for protein modeling, we find that it may be the 1D and 3D distances, instead of their features, that mainly determine amino acids' correlations. This is significantly different from the data in the macro world. In the macro world, 3D objects and their local parts have a strong and discriminative semantic pattern. For example, in a chair, there are chair legs and a chair seat. For a human body, there are the head, hands, arms, *etc*. Those semantic patterns are beneficial for networks to understand 3D structure via complicated mechanisms, *e.g*., self-attention. However, for proteins, there may not exist such semantic patterns. The main relations between amino acids are distance and direction, *i.e*., displacement. In this case, simpler attention mechanisms may be effective enough. Therefore, we replace the self-attention in the vanilla Transformer with a non-parametric distanced-based attention mechanism.

Suppose $\boldsymbol{D}^{1d} \in \mathbb{R}^{N\times N}$ is the distance matrix of 1D point orders where the order distance between the t-th point and the t'-th point is defined as $|t - t'|$ and $\boldsymbol{D}^{3d} \in \mathbb{R}^{N\times N}$

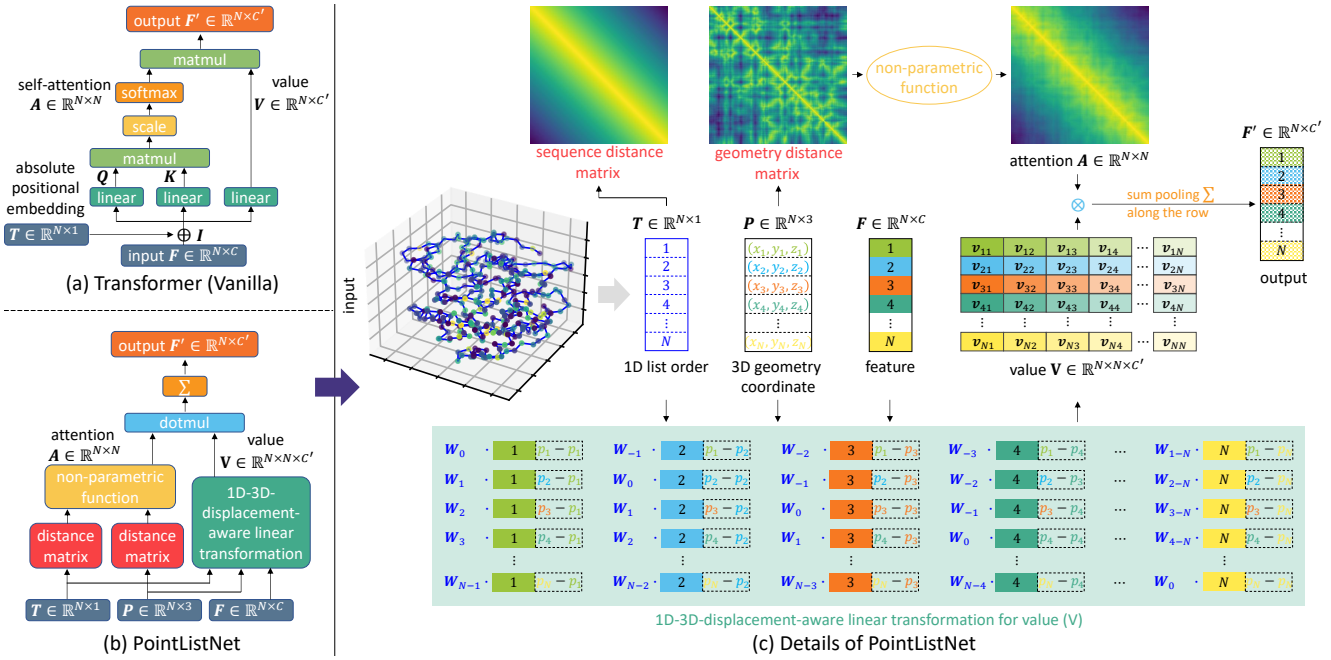

Figure 2. **(a)** Vanilla Transformer first adds the absolute positional embeddings into the input features and then performs three simple linear transformations for self-attention and value, in which the relative relations are not explicitly encoded. **(b)** Our PointListNet employs a non-paramedic distance-based attention mechanism. Moreover, we integrate the 1D and 3D displacement information into values to model the relative relations between points. **(c)** When performing 1D-3D-displacement-aware linear transformations, we use different groups of parameters (*i.e.*, $\boldsymbol{W}_{1-N}, \boldsymbol{W}_{2-N}, \cdots, \boldsymbol{W}_0, \cdots, \boldsymbol{W}_{N-2}, \boldsymbol{W}_{N-1}$) to reflect regular 1D order displacements but directly encode 3D geometry displacements (append to point features) due to their irregularity.

is the geometry distance matrix where the distance between the two points is defined as $\sqrt{\|\boldsymbol{p}_t - \boldsymbol{p}_{t'}\|_2}$. Then, to reduce the influence of different protein sizes, we normalize the two distances, *i.e.*, $\boldsymbol{D}^{1d}/L$ and $\boldsymbol{D}^{3d}/R$, where L and R are the longest sequence and geometry distances from the center to the farthest points, respectively. Third, we employ a non-parametric function g to calculate the attention,

$$\boldsymbol{A} = g(\frac{\boldsymbol{D}^{1d}}{L}, \frac{\boldsymbol{D}^{3d}}{R}). \tag{4}$$

Here, g is a decreasing function so that closer points in the sequence and geometry spaces have higher attention weights. Moreover, the output attention map $\boldsymbol{A}$ should be in $(0, 1)^{N\times N}$. In this way, $\boldsymbol{A}$ can be seen as a soft mask or a soft receptive field. The attention weight $\alpha_{tt'} \in (0, 1)$ in $\boldsymbol{A}$ can indicate how much the t-th and t'-th points are related.

There can be many methods to implement the g function. In this paper, we implement the function as $0.5 - 0.5\tanh\Big((\lambda(\boldsymbol{D}-1)+0.5)/\sqrt{\boldsymbol{D}\times(1-\boldsymbol{D})}\Big)$, where $\boldsymbol{D} = (\boldsymbol{D}^{1d}/L)\times(\boldsymbol{D}^{3d}/R)$ and $\lambda \in [3, 5]$.

3.3. 1D-3D-Displacement-Aware Value

Although the attention weight α can reflect the correlation degree of two points, as a scalar, it cannot encode more information about relative relations, such as direction, which are important for structure modeling. Therefore, we propose to integrate the 1D and 3D displacement information into values.

To do so, we follow the principle of convolution. In convolution, there is a kernel size that defines a receptive field, which can be seen as a hard or rigid attention mechanism. Points in the receptive field are with attention 1 otherwise with attention 0. Then, to encode the distance and direction information, *i.e.*, displacement, convolution performs different linear transformations on the features of neighbors based on displacements. For example, for a convolution with kernel size 3×3, there are 9 linear transformations for the displacements from the center to its neighbors. Motivated by convolution, we encode relative relations with the 1D-3D-displacement-aware linear transformations,

$$\boldsymbol{v}_{tt'} = \boldsymbol{f}_{t'} \cdot \boldsymbol{W}_{t'-t, \boldsymbol{p}_{t'}-\boldsymbol{p}_t}, \tag{5}$$

where $\boldsymbol{W}_{t'-t, \boldsymbol{p}_{t'}-\boldsymbol{p}_t} \in \mathbb{R}^{C\times C'}$ is the parameters for the linear transformation of 1D-3D-displacement $(t'-t, \boldsymbol{p}_{t'}-\boldsymbol{p}_t)$.

However, due to the continuity of point coordinates, there would be countless possible 3D geometry displacements, even in an extremely small area. In this case, we cannot assign an individual $\boldsymbol{W}$ for each geometry displace-

Table 1. Comparison with 3D point cloud methods, *i.e.*, PointNet++ [43], DGCNN [53], Point Transformer [62] and PointMLP [38], on protein fold classification and enzyme catalytic reaction classification (accuracy %). Experiments are conducted by ourselves.

Method	Modeling	Protein Fold Classification			Enzyme Reaction
		Fold	Superfamily	Family	Classification
PointNet++ [43]	$\boldsymbol{f}'_t = \underset{\|\boldsymbol{p}_{t'}-\boldsymbol{p}_t\|\le r}{\mathrm{MAX}} \mathrm{MLP}\big([\boldsymbol{f}_{t'}, \boldsymbol{p}_{t'}-\boldsymbol{p}_t]\big)$	26.0	37.7	93.8	78.4
DGCNN [53]	$\boldsymbol{f}'_t = \underset{\boldsymbol{f}_{t'}\in \mathrm{TopK}(\boldsymbol{f}_t)}{\mathrm{MAX}} \mathrm{MLP}\big([\boldsymbol{f}_{t'}, \boldsymbol{f}_{t'}-\boldsymbol{f}_t]\big)$	25.6	39.2	94.4	80.1
Point Transformer [62]	$\boldsymbol{f}'_t = \sum_{\|\boldsymbol{p}_{t'}-\boldsymbol{p}_t\|\le r} \alpha_{tt'} \times \big(\boldsymbol{W}_3 \cdot \boldsymbol{f}_{t'} + \delta_{tt'}\big)$ $\alpha_{tt'} = \mathrm{softmax}\big(\mathrm{MLP}(\boldsymbol{W}_1 \cdot \boldsymbol{f}_t - \boldsymbol{W}_2 \cdot \boldsymbol{f}_{t'} + \delta_{tt'})\big)$ $\delta_{tt'} = \mathrm{MLP}(\boldsymbol{p}_t - \boldsymbol{p}_{t'})$	26.4	40.1	92.0	81.3
PointMLP [38]	$\boldsymbol{f}'_t = \mathrm{MLP}\big(\underset{\boldsymbol{f}_{t'}\in \mathrm{TopK}(\boldsymbol{f}_t)}{\mathrm{MAX}} \mathrm{MLP}(\boldsymbol{f}_{tt'})\big)$ $\boldsymbol{f}_{tt'} = \alpha \odot \frac{\boldsymbol{f}_{t'}-\boldsymbol{f}_t}{\delta+\epsilon} + \beta$ $\delta = \sqrt{\frac{1}{K\times N\times C}\sum_{i=1}^{N}\sum_{j=1}^{K}(\boldsymbol{f}_{t'}-\boldsymbol{f}_t)^2}$	26.8	38.8	94.2	79.7
PointListNet (ours)	3D Coordinate	36.8	55.3	97.4	84.5
	1D Order & 3D Coordinate	**55.2**	**76.4**	**99.5**	**88.0**

ment. To address this problem, we follow PointNet++ [43] to treat 3D displacement as a part of feature, and concatenate point features and 3D displacements for modeling,

$$\boldsymbol{v}_{tt'} = [\boldsymbol{f}_{t'}, \boldsymbol{p}_{t'} - \boldsymbol{p}_t] \cdot \boldsymbol{W}_{t'-t}, \tag{6}$$

where $\boldsymbol{W}_{t'-t} \in \mathbb{R}^{(C+3)\times C'}$ and $[\cdot,\cdot]$ denotes concatenation. In this way, our 1D-3D-displacement-aware linear transformations take advantage of both the regularity of the 1D sequence structure and the irregularity of the 3D geometry structure.

However, the above implementation is not rotationally invariant. Rotating point lists may lead to different representations. For micro-particles, there is no definition or concept for the up, down, left and right directions. Therefore, rotationally invariant is important for modeling them. Inspired by [24], we replace the 3D displacement encoding in Eq. (6) with rotationally invariant encoding [28].

Note that because of the goal to capture the relative structure, the values in our PointListNet is a 3D tensor ($\mathbf{V} \in \mathbb{R}^{N\times N\times C'}$), instead of a 2D matrix ($\boldsymbol{V} \in \mathbb{R}^{N\times C'}$) in the vanilla Transformer. Finally, as shown in Fig. 2(c), the global weighted sum is performed on the values $\mathbf{V}$ to obtain the new feature,

$$\boldsymbol{f}'_t = \sum_{t'=1}^{N} \alpha_{tt'} \boldsymbol{v}_{tt'}, \tag{7}$$

where $\boldsymbol{f}'_t \in \mathbb{R}^{1\times C'}$. In this way, the point features are updated by collecting the information from their related points, which are donated as $\boldsymbol{F}' \in \mathbb{R}^{N\times C'}$.

3.4. Hierarchical Architecture

Although our PointListNet models 3D geometry coordinates in an irregular manner, which avoids employing countless parameters, capturing the dependency of 1D sequence orders still requires many parameters. For a point list with N points, PointListNet needs $2N-1$ groups of independent parameters, *i.e.*, $\{\boldsymbol{W}_{-(N-1)}, \boldsymbol{W}_{-(N-2)}, \cdots, \boldsymbol{W}_0, \cdots, \boldsymbol{W}_{N-2}, \boldsymbol{W}_{N-1}\}$, which is too many for most devices. Moreover, as shown in Fig. 2(c), when two points are too far away, their correlation tends to be 0. It is not necessary to capture the structure of every two points. Therefore, following Point Transformer [62], we employ the local modeling technique to limit the attention range by defining a sequence distance threshold l and a geometry distance threshold r. The two distances for the attention calculation are also normalized based on l and r, respectively.

To extract global representations for classification, we downsample points as the network deepens. In this way, we can construct a hierarchical or pyramid architecture. At last, we add a classifier for point list recognition tasks.

4. Experiments

4.1. Evaluation Tasks and Datasets

Following [24, 25, 60], we evaluate the proposed method on two recognition tasks: protein fold classification and enzyme reaction classification. Mean accuracy is used as the

Table 2. Accuracy (%) of protein fold classification and enzyme catalytic reaction classification. *Results are from [25]. †Results are from [60].

Level	Input	Method	Protein Fold Classification			Enzyme Reaction
			Fold	Superfamily	Family	Classification
Atom Level (Molecule)	3D Coordinate	GCN [31]*	16.8	21.3	82.8	67.3
		EdgePool (GNN)* [14]	12.9	16.3	72.5	57.9
		3D CNN [12]*	31.6	45.4	92.5	72.2
	3D Coordinate & Bond	IEConv [25]	45.0	69.7	98.9	87.2
Residue Level (Protein)	1D Order	1D ResNet [44]†	10.1	7.21	23.5	24.1
		DeepFS (1D ResNet)* [27]	17.0	31.0	77.0	70.9
		DeepFS (1D CNN)* [27]	40.9	50.7	76.2	-
		LSTM [44]†	6.41	4.33	18.1	11.0
		Transformer [44]†	9.22	8.81	40.4	26.6
	3D Coordinate	GAT [50]†	12.4	16.5	72.7	55.6
	1D Order & 3D Coordinate	GraphQA [4]*	23.7	32.5	84.4	60.8
		GVP [29]†	16.0	22.5	83.8	65.5
		IEConv [24]	47.6	70.2	99.2	87.2
		GearNet [60]	28.4	42.6	95.3	79.4
		GearNet-IEConv [60]	42.3	64.1	99.1	83.7
		GearNet-Edge [60]	44.0	66.7	99.1	86.6
		GearNet-Edge-IEConv [60]	48.3	70.3	**99.5**	85.3
		PointListNet (ours)	**55.2**	**76.4**	**99.5**	**88.0**

evaluation metric.

Protein Fold Classification. Protein fold classification is important in the study of the relationship between protein structure and protein evolution. The fold classes indicate protein secondary structure compositions, orientations and connection orders. We follow [25] to conduct protein fold classification on the training/validation/test splits of the SCOPe 1.75 data set of [27], which in total contains 16,712 proteins with 1,195 fold classes. The 3D coordinates of the proteins were collected from the SCOPe 1.75 database [40]. The data set provides three different evaluation scenarios. 1) Fold, in which proteins from the same superfamily are not used during training. 2) Superfamily, in which proteins from the same family are not provided during training. 3) Family, in which proteins of the same family are available during training.

Enzyme Reaction Classification. Enzyme reaction classification can be seen as a protein function classification task, which is based on the enzyme-catalyzed reaction according to all four levels of the Enzyme Commission (EC) number [55]. We use the dataset collected by [25], which includes 384 four-level EC classes and 29,215/2,562/5,651 proteins for training/validation/test, respectively.

4.2. Training Setup

The network is trained with the SGD optimizer for 500 epochs. The batch size is set to 8. The learning rate is set to 0.01 and decreases by 10% after the 300 and 400 epochs, respectively.

Table 3. Comparison between self-attention and our distance-based attention on protein fold classification and enzyme catalytic reaction classification (accuracy %).

Method		Fold Classification			Enzyme
		Fold	Superfamily	Family	Reaction
w/o Attention		52.0	73.2	99.1	87.2
Self-Attention	1 head	50.3	71.0	98.8	86.2
	2 heads	51.0	71.6	99.1	86.7
	4 heads	50.5	71.1	98.8	86.4
Distance-based Attention		**55.2**	**76.4**	**99.5**	**88.0**

4.3. Comparison with Point Cloud Methods

The residue-level 3D structure of proteins can be seen as 3D point lists. If the sequence structure is neglected, the amino acids of a protein form a point cloud. Therefore, existing methods for point cloud processing can be applied to protein modeling. However, point cloud methods are largely ignored in existing works. Most of them are based on existing graph-based methods. To fill this gap, we investigate point cloud methods for protein modeling and compare our method with them. In this paper, we consider three point cloud methods: PointNet++ [43], DGCNN [53], Point Transformer [62] and PointMLP [38]. For DGCNN and PointMLP, we follow the original paper to concatenate point coordinates and features as the input of the network.

Table 4. Efficiency Comparison between self-attention and our distance-based attention.

Efficiency	w/o Attention	Self-Attention	Distance-based
# Parameters	34.0 M	38.6M	34.0 M
Running Time	11.20 ms	13.56ms	11.23 ms

These two methods employ TopK to search K neighbors for each query point. In the formulation of PointMLP, α and β are two learnable parameters. We conduct the experiments by ourselves.

The modeling approaches of these point cloud methods and the experimental results are shown in Table 1. Even only with 3D geometry structure, PointListNet still significantly outperforms the point cloud methods. There may be two reasons leading to the superior of our methods. First, those point cloud methods do not employ effective attention mechanisms to capture point correlations. Second, they are not rotationally invariant.

4.4. Comparison with the State-of-the-Art

As a kind of large biomolecules and macromolecules, proteins can be modeled in two levels. The first one is the atom level, in which the 3D coordinates and types of atoms are used for the protein's geometry structure understanding via GNNs [14, 31], 3D CNNs [12], *etc*. In addition to the atom coordinate and type information, covalent or hydrogen bonds can also be used to enhance the representation [25], which can be seen as the topology structure. We refer to the input of such kind of methods as "3D coordinates & bond".

The second level is based on amino acids or residues, which is referred to as the residue level in this paper. At this level, we compare our method with existing "1D order", "3D coordinate" and "1D order & 3D coordinate" methods. They are as follows,

- 1D order. Because proteins are lists of amino acids, 1D CNN [44], LSTM [44] and Transformer [27,44] can be employed. The networks in [44] consist of two layers while those in [27] contain 10 layers.
- 3D coordinate. To model the irregular 3D geometry structure, graph neural networks are wildly used in existing protein-related works, *e.g*., GAT [50].
- 1D order & 3D coordinate. To model the sequence and geometry structures together, existing methods, *e.g*., GraphQA [4], GVP [29], IEConv [24] and GearNet [60], process geometric and sequential displacements together, *e.g*., via concatenation, or model the sequence structure in a similar way to geometry modeling, *i.e*., directly encoding sequential displacements, thus neglecting the regularity of the 1D structure.

Results are shown in Table 2. Note that none of the existing methods achieved state-of-the-art accuracy on both two tasks. In contrast, our method significantly outperforms all the existing methods. For example, on the superfamily fold classification, our PointListNet outperforms the previous state-of-the-art method, *i.e*., GearNet-Edge-IEConv, by 6.1%. This demonstrates the effectiveness of the proposed PointListNet that employs regular and irregular approaches for sequence and geometry modeling, respectively.

Table 5. Impact of 1D and 3D structure modeling on protein fold classification and enzyme catalytic reaction classification (accuracy %).

Structure	Fold Classification			Enzyme
	Fold	Superfamily	Family	Reaction
1D Order	13.1	18.7	86.4	70.0
3D Coordinate	36.8	55.3	97.4	84.5
1D Order & 3D Coordinate	**55.2**	**76.4**	**99.5**	**88.0**

The residue-level methods outperform the atom-level methods. This may be because proteins usually contain thousands of atoms and it is challenging to model the geometry structure based on so many points. In constant, a typical protein is usually made up of 300 amino acids. Therefore, protein modeling via amino acids is easier than via atoms.

4.5. Ablation Study

4.5.1 Impact of Distance-based Attention

One difference between our PointListNet and the vanilla Transformer is that we employ a non-parametric distance-based attention mechanism. To evaluate the effectiveness of the proposed distance-based attention, we compared it with the self-attention in the vanilla Transformer. We also compare the lower-bound method, which does not employ attention. Because we follow Point Transformer that integrates the local modeling technique into the hierarchical framework, even though without attention, the lower-bound method can also achieve satisfactory accuracy.

Results are shown in Table 3. Our distance-based attention outperforms the other two methods. In particular, self-attention achieves inferior accuracy compared to the baseline where no attention is used. That is because self-attention tries to find relevant elements in data (*e.g*., patches in images or words in texts) based on their semantic relations. However, for proteins, there are only 21 types of amino acids and there may be no strong semantic relations among amino acids. In this case, feature-based self-attention may be unstable, thus leading to inferior accuracy.

Then, we investigate the impact of our distance-based attention on efficiency. For running time, we conduct a test experiment on protein fold classification with Intel(R) Xeon(R) Silver 4214 CPU @ 2.20GHz and a single Nvidia Quadro RTX A5000. Experiments are shown in Table 4.

Table 6. Comparison between regular and irregular 1D sequence modeling on protein fold classification and enzyme catalytic reaction classification (accuracy %).

Modeling	Fold Classification			Enzyme
	Fold	Superfamily	Family	Reaction
Irregular	48.9	70.5	98.9	86.7
Regular	**55.2**	**76.4**	**99.5**	**88.0**

Compared to self-attention, our non-parametric distance-based attention does not significantly increase the number of parameters and running time.

4.5.2 Impact of 1D and 3D Structure Modeling

The structure of 3D point lists consists of a 1D list part and a 3D geometry part. In this section, we investigate the influence of each of the two parts on modeling. As shown in Table 5, spontaneously modeling the two structures significantly outperforms the 1D-order or 3D-coordinate method. Moreover, in the two 1D-order and 3D-coordinate methods, 3D-coordinate surpasses 1D-order by large margins. This indicates that 3D geometry dominates the protein structure, which is consistent with the consensus that it is mainly the 3D structure that determines proteins' function.

4.5.3 Impact of Regular Modeling on 1D List Structure in Point Lists

To simultaneously model the 1D and 3D structures in point lists, existing methods [4, 24, 29, 60] treat regular and discrete sequential orders as irregular data and model them in similar ways to 3D coordinates. Different from them, we employ different linear transformations to reflect regular 1D sequential orders and directly encode irregular 3D coordinates for modeling geometry.

In this section, we compare our regular 1D modeling method with an irregular approach. To this end, we first concatenate 1D displacement and the relatively encoded 3D displacement and then use the same linear transformation to model them. As shown in Table 6, our regular 1D modeling method effectively improves the accuracy, verifying our motivation that regular and irregular data should be processed in regular and irregular manners, respectively.

4.5.4 Impact of Relative Position Modeling

The vanilla Transformer directly integrates the embedding of absolute positions of points into the input features, which does not explicitly capture the relative structure of data. However, most amino acids do not have a strong or obvious relationship with their absolute positions. Instead, it is the relative position, *i.e.*, displacement, that determines their relations. Therefore, we integrate the relative structure modeling into the linear transformation for value generation. As shown in Table 7, relative position modeling significantly outperforms absolute structure modeling.

Table 7. Impact of relative position modeling on protein fold classification and enzyme catalytic reaction classification (accuracy %).

Psition	Fold Classification			Enzyme
	Fold	Superfamily	Family	Reaction
Absolute	31.1	48.7	94.8	82.7
Relative	**55.2**	**76.4**	**99.5**	**88.0**

Table 8. Impact of rotationally invariant displacement encoding on protein fold classification and enzyme catalytic reaction classification (accuracy %).

Modeling	Fold Classification			Enzyme
	Fold	Superfamily	Family	Reaction
Variant	44.9	60.1	95.1	87.2
Invariant	**55.2**	**76.4**	**99.5**	**88.0**

4.5.5 Impact of Rotationally Invariant Displacement Encoding

Because there is no concept or definition of direction for micro-particles, to be rotationally invariant is important for modeling them. In this section, we investigate the impact of invariant displacement encoding. As shown in Table 8, rotationally invariant displacement encoding can effectively improve the accuracy.

5. Conclusion

In this paper, we propose a Transformer-style PointListNet for 3D point list modeling. In our PointListNet, we replace self-attention with non-parametric distance-based attention, and integrate relative structure modeling into Transformer and employ regular and irregular methods to capture the sequence and geometry structures, respectively. To show the effectiveness of the proposed PointListNet for 3D point list modeling, we conduct experiments on protein fold classification and enzyme reaction classification and achieve new state-of-the-art accuracy. [1]

Acknowledgments

This work is supported by National Key R&D Program of China under Grant No. 2020AAA0108800, the Fundamental Research Funds for the Central Universities (No. 226-2022-00051) and the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (#A18A2b0046).

[1] Part of this work was done when Hehe Fan was a Research Fellow at National University of Singapore.

References

[1] Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12):1315–1322, 2019. 2

[2] Afshine Amidi, Shervine Amidi, Dimitrios Vlachakis, Vasileios Megalooikonomou, Nikos Paragios, and Evangelia I Zacharaki. Enzynet: enzyme classification using 3d convolutional neural networks on spatial representation. *PeerJ*, 6:e4750, 2018. 2

[3] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv*, 1803.01271, 2018. 1

[4] Federico Baldassarre, David Menéndez Hurtado, Arne Elofsson, and Hossein Azizpour. Graphqa: protein model quality assessment using graph convolutional networks. *Bioinform.*, 37(3):360–366, 2021. 2, 6, 7, 8

[5] Tristan Bepler and Bonnie Berger. Learning protein sequence embeddings using information from structure. In *ICLR*, 2019. 2

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020. 3

[7] Can Chen, Jingbo Zhou, Fan Wang, Xue Liu, and Dejing Dou. Structure-aware protein self-supervised learning. *arXiv*, 2204.04213, 2022. 2

[8] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *CVPR*, 2017. 2

[9] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv*, 1412.3555, 2014. 1

[10] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 2

[11] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *ACL*, pages 2978–2988, 2019. 3

[12] Georgy Derevyanko, Sergei Grudinin, Yoshua Bengio, and Guillaume Lamoureux. Deep convolutional networks for quality assessment of protein folds. *Bioinform.*, 34(23):4046–4053, 2018. 2, 6, 7

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019. 1, 3

[14] Frederik Diehl. Edge contraction pooling for graph neural networks. *arXiv*, 1905.10990, 2019. 6, 7

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 2

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3

[17] Hehe Fan, Yi Yang, and Mohan Kankanhalli. Point spatio-temporal transformer networks for point cloud video modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2

[18] Hehe Fan, Yi Yang, and Mohan S. Kankanhalli. Point 4d transformer networks for spatio-temporal modeling in point cloud videos. In *CVPR*, 2021. 3

[19] Hehe Fan, Xin Yu, Yi Yang, and Mohan Kankanhalli. Deep hierarchical representation of point cloud videos via spatio-temporal decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2

[20] Xiaoyu Feng, Heming Du, Yueqi Duan, Yongpan Liu, and Hehe Fan. Seformer: Structure embedding transformer for 3d object detection. In *AAAI*, 2023. 3

[21] Vladimir Gligorijević, P Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C Taylor, Ian M Fisk, Hera Vlamakis, et al. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, 12(1):1–14, 2021. 2

[22] Meng-Hao Guo, Junxiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. PCT: point cloud transformer. *Comput. Vis. Media*, 7(2):187–199, 2021. 2

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[24] Pedro Hermosilla and Timo Ropinski. Contrastive representation learning for 3d protein structures. *arXiv*, 2205.15675, 2022. 5, 6, 7, 8

[25] Pedro Hermosilla, Marco Schäfer, Matej Lang, Gloria Fackelmann, Pere-Pau Vázquez, Barbora Kozlíková, Michael Krone, Tobias Ritschel, and Timo Ropinski. Intrinsic-extrinsic convolution and pooling for learning on 3d protein structures. In *ICLR*, 2021. 2, 5, 6, 7

[26] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 1

[27] Jie Hou, Badri Adhikari, and Jianlin Cheng. Deepsf: Deep convolutional neural network for mapping protein sequences to folds. In *ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 2018. 2, 6, 7

[28] John Ingraham, Vikas K. Garg, Regina Barzilay, and Tommi S. Jaakkola. Generative models for graph-based protein design. In *NeurIPS*, 2019. 2, 5

[29] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron O. Dror. Learning from protein structure with geometric vector perceptrons. In *ICLR*, 2021. 2, 6, 7, 8

[30] Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP*, 2014. 1

[31] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 6, 7

[32] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *ECCV*, pages 491–507, 2020. 3

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 1

[34] Maxat Kulmanov and Robert Hoehndorf. Deepgoplus: improved protein function prediction from sequence. *Bioinform.*, 37(8):1187, 2021. 2

[35] Maxat Kulmanov, Mohammad Asif Khan, and Robert Hoehndorf. Deepgo: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinform.*, 34(4):660–668, 2018. 2

[36] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *CVPR*, 2022. 2, 3

[37] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *NeurIPS*, 2018. 2

[38] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. In *ICLR*, 2022. 2, 5, 6

[39] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and optimizing LSTM language models. In *ICLR*, 2018. 1

[40] Alexey G Murzin, Steven E Brenner, Tim Hubbard, and Cyrus Chothia. Scop: a structural classification of proteins database for the investigation of sequences and structures. *Journal of molecular biology*, 247(4):536–540, 1995. 6

[41] Charles R. Qi, Or Litany, Kaiming He, and Leonidas J. Guibas. Deep hough voting for 3d object detection in point clouds. In *ICCV*, 2019. 2

[42] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 2

[43] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 2, 5, 6

[44] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John F. Canny, Pieter Abbeel, and Yun S. Song. Evaluating protein transfer learning with TAPE. In *NeurIPS*, 2019. 2, 6, 7

[45] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *CVPR*, 2017. 2

[46] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015. 1

[47] Nils Strodthoff, Patrick Wagner, Markus Wenzel, and Wojciech Samek. Udsmprot: universal deep sequence models for protein classification. *Bioinform.*, 36(8):2401–2409, 2020. 2

[48] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*, 2019. 2

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1, 2, 3

[50] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *GAT*, 2018. 6, 7

[51] Guangzhi Wang, Hehe Fan, and Mohan S. Kankanhalli. Text to point cloud localization with relation-enhanced transformer. In *AAAI*, 2023. 3

[52] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018. 3

[53] 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 Trans. Graph.*, 38(5):146:1–146:12, 2019. 2, 5, 6

[54] Zichen Wang, Steven A Combs, Ryan Brand, Miguel Romero Calvo, Panpan Xu, George Price, Nataliya Golovach, Emannuel O Salawu, Colby J Wise, Sri Priya Ponnapalli, et al. Lm-gvp: A generalizable deep learning framework for protein property prediction from sequence and structure. *bioRxiv*, 2021. 2

[55] Edwin C Webb. *Enzyme nomenclature 1992*. Number Ed. 6. Academic Press, 1992. 6

[56] Wenxuan Wu, Zhongang Qi, and Fuxin Li. Pointconv: Deep convolutional networks on 3d point clouds. In *CVPR*, 2019. 2

[57] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, pages 5754–5764, 2019. 3

[58] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. Comparative study of CNN and RNN for natural language processing. *arXiv*, 1702.01923, 2017. 1

[59] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-net: Point cloud upsampling network. In *CVPR*, 2018. 2

[60] Zuobai Zhang, Minghao Xu, Arian R. Jamasb, Vijil Chenthamarakshan, Aurélie C. Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. *arXiv*, 2203.06125, 2022. 2, 5, 6, 7, 8

[61] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *CVPR*, pages 10073–10082, 2020. 3

[62] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *ICCV*, 2021. 2, 3, 5, 6