PointRWKV: Efficient RWKV-Like Model for Hierarchical Point Cloud Learning

Qingdong He[1*], Jiangning Zhang[1*], Jinlong Peng[1†], Haoyang He[2], Xiangtai Li[3], Yabiao Wang[1], Chengjie Wang[1]

[1]Youtu Lab, Tencent
[2]Zhejiang University
[2]Nanyang Technological University
{yingcaihe, vtzhang, jeromepeng, caseywang, jasoncjwang}@tencent.com, haoyanghe@zju.edu.cn,xiangtai94@gmail.com

Abstract

Transformers have revolutionized the point cloud learning task, but the quadratic complexity hinders its extension to long sequences. This puts a burden on limited computational resources. The recent advent of RWKV, a fresh breed of deep sequence models, has shown immense potential for sequence modeling in NLP tasks. In this work, we present PointRWKV, a new model of linear complexity derived from the RWKV model in the NLP field with the necessary adaptation for 3D point cloud learning tasks. Specifically, taking the embedded point patches as input, we first propose to explore the global processing capabilities within PointRWKV blocks using modified multi-headed matrix-valued states and a dynamic attention recurrence mechanism. To extract local geometric features simultaneously, we design a parallel branch to encode the point cloud efficiently in a fixed radius near-neighbors graph with a graph stabilizer. Furthermore, we design PointRWKV as a multi-scale framework for hierarchical feature learning of 3D point clouds, facilitating various downstream tasks. Extensive experiments on different point cloud learning tasks show our proposed PointRWKV outperforms the transformer- and mamba-based counterparts, while significantly saving about 42% FLOPs, demonstrating the potential option for constructing foundational 3D models.

Code — https://hithqd.github.io/projects/PointRWKV/

Introduction

Point cloud analysis is a fundamental vision task that has a wide range of applications, including autonomous driving (Mao et al. 2022), virtual reality, and robotics (Shafiullah et al. 2022), *etc*. Unlike 2D images, the intrinsic irregularity and sparsity of point clouds make it a challenging task to conduct point cloud representation learning. Balancing accuracy and complexity simultaneously remains an enduring problem.

Pioneered by PointNet (Qi et al. 2017a) and PointNet++ (Qi et al. 2017b), a series of works (Atzmon, Maron, and Lipman 2018; Li et al. 2018; Komarichev, Zhong, and Hua 2019; Qian et al. 2022; Thomas et al. 2019; Wu, Qi, and Fuxin 2019; Ma et al. 2022; Wang et al. 2019) have emerged to follow the MLP architecture to extract geometric features (Figure 1 (a)). Recently, transformers (Dosovitskiy et al. 2020; Liu et al. 2021) have surfaced as a promising approach for point cloud analysis, demonstrating significant potential with a key component of the attention mechanism to effectively capture the relationships among a set of points. Focusing on enhancing transformers for point clouds, Point Transformers (Zhao et al. 2021; Wu et al. 2022, 2023b) apply vector attention and enhance the performance and efficiency for different tasks. Further, by extending self-supervised learning mechanism, some works (Yu et al. 2022; Pang et al. 2022; Zhang et al. 2022; Qi et al. 2023) have obtained decent performance. Among them, PointMAE (Pang et al. 2022), as depicted in Figure 1 (b), introduces a standard Transformer architecture and a shifting mask tokens operation at

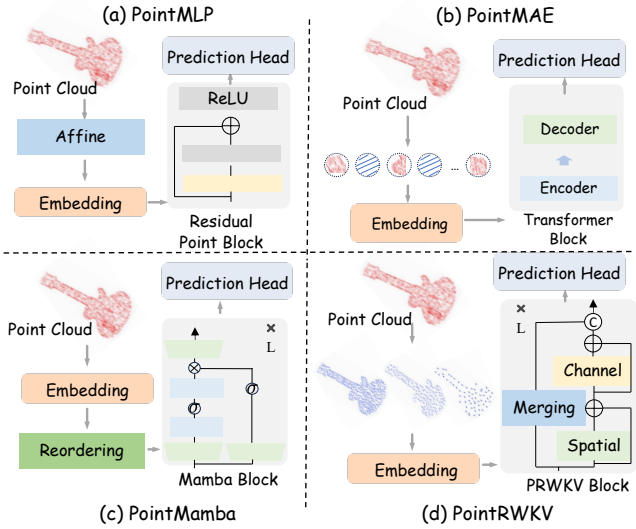

Figure 1: **Architecture comparison with different methods:** (a) MLP-based PointMLP (Ma et al. 2022), (b) transformer-based PointMAE (Pang et al. 2022), (c) mamba-based PointMamba (Liang et al. 2024) and (d) ours PointRWKV with linear complexity is capable of integrating the advantages of both global and local modeling, and multi-scale features endow it with more refined prediction accuracy.

*These authors contributed equally.
†Corresponding author.

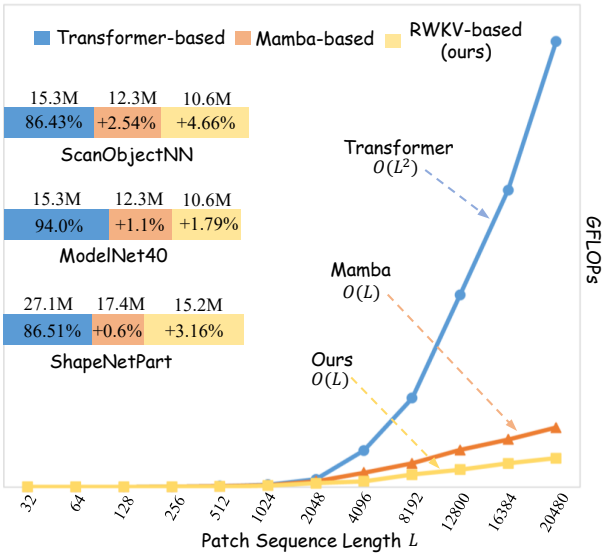

Figure 2: **Accuracy-speed tradeoff.** (Left) Overall accuracy acquired by different methods with relative parameters, (Right) FLOPs increase with sequence length.

a high ratio to reach better accuracy. However, directly applying exhaustive attention mechanisms to extended point tokens increases the demand for computational resources. This is because of the quadratic complexity inherent in attention computations, impacting both computation and memory. Furthermore, the recent state space models (SSMs) (Gu and Dao 2023; Liu et al. 2024b) introduce an alternative approach for point sequence modeling. Based on the structured SSMs, several works (Liang et al. 2024; Zhang et al. 2024; Liu et al. 2024a; Han et al. 2024) have attempted to enable the model to causally capture the point cloud structure with consistent traverse serialization mechanism or reordering strategy, as illustrated in Figure 1 (c). Albeit effective, the inherent property of unidirectional modeling of the vanilla SSM hinders them from achieving superior performance for the unordered point cloud data. Considering this, one essential question arises: *How to design a method that runs with linear complexity and achieves a high performance simultaneously?*

We notice the recent advance of RWKV (Peng et al. 2023) model within natural language processing (NLP), serving as a promising solution for attaining enhanced efficiency and handling extensive textual data. This paper pioneers the application of RWKV into the 3D point cloud learning domain, introducing PointRWKV, as illustrated in Figure 1(d). PointRWKV maintains the core structure and benefits of RWKV while preserving the linear complexity and processing multi-scale point input. Significantly, it exhibits a reduced parameter count and computational requirement, rendering it aptly suitable for deep point cloud learning.

Specifically, PointRWKV utilizes RWKV-like architecture to model the global features of point clouds with a series of stacked PRWKV blocks. Considering the unique properties of the strong geometric and semantic dependence between local parts and the overall 3D shapes of point cloud data, we modify the architecture into multi-stage hierarchies for progressively encoding multi-scale features of point clouds akin to an asymmetric U-Net (Ronneberger, Fischer, and Brox 2015). Within each PRWKV block, we devise a parallel strategy for hierarchical point cloud feature representation, enabling local and global feature aggregation. For the original RWKV branch, we present the bidirectional quadratic expansion function with multi-headed matrix-valued states and adapt the original causal attention mechanism into a dynamic bidirectional global attention mechanism. The former can broaden the semantic scope of individual tokens, while the latter facilitates the computation of global attention within linear computation complexity. For another branch, the graph stabilizer mechanism is proposed to conduct local graph-based merging for learning the compact representation of the point cloud. These designs boost the model's capacity while maintaining scalability and stability. In this way, our PointRWKV inherits the efficiency of RWKV in processing global information and sparse inputs while also being capable of modeling the local concept of point clouds.

By employing multi-scale pre-training, our PointRWKV can encode point clouds from local-to-global hierarchies, learning robust 3D representations and demonstrating exceptional transfer capabilities. We conduct extensive experiments on various point cloud learning tasks (*e.g.*, classification, part segmentation, and few-shot learning) to demonstrate the effectiveness of our method. As shown in Figure 2, after self-supervised pre-training on ShapeNet (Chang et al. 2015), PointRWKV achieves 93.63% (+4.66%) overall accuracy on the ScanObjectNN (Uy et al. 2019) and 96.89% (+1.79%) accuracy on ModelNet40 (Wu et al. 2015) for shape classification, 90.26% (+3.16%) instance mIoU on ShapeNetPart (Yi et al. 2016) for part segmentation, setting new state-of-the-art (SoTA) among pre-trained models. Meanwhile, PointRWKV reduces 13% in parameters and 42% in FLOPs compared to the transformer- and mamba-based counterparts, demonstrating the potential of RWKV in the 3D vision tasks. Our contributions are summarized as follows:

- We propose PointRWKV, which innovatively applies the RWKV framework to address point cloud learning tasks. This approach reaches a better balance among the parameters, computation complexity, and efficiency, making it an alternative for 3D vision tasks.
- To learn the hierarchical representations of point clouds, we introduce the multi-stage hierarchies architecture to encode multi-scale point features. Within each block in this architecture, we design the parallel strategy for hierarchical local and global feature aggregation, equipped with the bidirectional quadratic expansion, the dynamic bidirectional global attention mechanism, and the graph stabilizer mechanism.
- Our PointRWKV exhibits state-of-the-art performance on various downstream tasks, outperforming transformer- and mamba-based counterparts in both efficiency and computation complexity, which indicates

our approach to be a powerful representation learner for 3D point clouds.

Related Work

MLP/CNN-based Models For Point Cloud. With the progressive evolution of deep neural networks, point cloud learning has gained increasing interest, leading to the emergence of numerous deep architectures and models in recent years (Qi et al. 2017a; Huang et al. 2022b; Deng et al. 2023; Ma et al. 2022; Huang et al. 2022a). PointNet (Qi et al. 2017a) and PointNet++ (Qi et al. 2017b) are the two pioneering approaches that use MLPs directly to process point clouds. Following these, a series of works (Atzmon, Maron, and Lipman 2018; Choy, Gwak, and Savarese 2019; Deng et al. 2023; Li et al. 2018; Zhao et al. 2019; Komarichev, Zhong, and Hua 2019; Qian et al. 2022; Thomas et al. 2019; Wu, Qi, and Fuxin 2019; Ma et al. 2022) attempt to design various deep architectures to better capture local and global geometric features. Meanwhile, several works (He et al. 2022; Landrieu and Simonovsky 2018; Wang et al. 2019) explore to utilize graph neural networks to construct the 3D geometric relationships between the points.

Transformer-based Models For Point Cloud. Equipped with attention mechanism, transformer (Vaswani et al. 2017) has evolved not only in NLP (Devlin et al. 2018; Radford et al. 2018) but also in 2D (Dosovitskiy et al. 2020; Liu et al. 2021) and 3D vision (Misra, Girdhar, and Joulin 2021; Mao et al. 2021; Hatamizadeh et al. 2022), including the point cloud learning (Guo et al. 2021; Zhao et al. 2021; Yu et al. 2021; Park et al. 2022). The Point Transformer series (Zhao et al. 2021; Wu et al. 2022, 2023b) are pioneering in introducing self-attention layers specifically tailored for point clouds. Significant advancements are marked by the implementation of grouped vector attention and partition-based pooling techniques for enhanced point cloud analysis. Recently, a surge of research (Yu et al. 2022; Pang et al. 2022; Zhang et al. 2022) efforts have been directed towards enhancing point cloud data representation through self-supervised learning, particularly by leveraging masked point modeling. This self-supervised approach enables models to learn the intrinsic structure and features of point clouds in the absence of explicit labels, fostering a novel paradigm in the field.

Mamba-based Models. The Mamba model, a deep learning architecture grounded in state-space models (SSMs) (Gu and Dao 2023; Mehta et al. 2022; Fu et al. 2022; Smith, Warrington, and Linderman 2022), exhibits enhanced processing speed and scalability by enabling selective memory retention or discarding of inputs. In recent studies, the Mamba model has demonstrated its efficacy across a diverse range of applications, such as NLP (Pióro et al. 2024; Anthony et al. 2024), 2D image analysis (Liu et al. 2024b; He et al. 2024a,b), and video processing (Yang, Xing, and Zhu 2024,?), showcasing its versatility and adaptability. Recent researches (Liang et al. 2024; Zhang et al. 2024; Liu et al. 2024a; Han et al. 2024) have successfully employed the Mamba model in 3D point cloud tasks, where point cloud data is represented as state vectors, allowing the model to capture the dynamic characteristics of the data, resulting in a commendable performance.

Receptance Weighted Key Value (RWKV) Models. Transformer's memory and computational complexity scale quadratically with sequence length, while RNNs exhibit linear growth in these requirements, albeit limited by parallelization and scalability. Recently, the innovative Receptance Weighted Key Value (RWKV) (Peng et al. 2023, 2024) model is introduced, seamlessly combining Transformer's efficient parallel training with RNN's efficient inference. Empirical results demonstrate that RWKV exhibits linear time complexity and holds promise for outperforming Transformers in long-sequence reasoning tasks. Vision-RWKV (Duan et al. 2024) is an advanced adaptation of the RWKV mechanism tailored for visual tasks, leveraging its linear computational complexity to deliver exceptional efficiency in processing high-resolution images. The following works (Yuan et al. 2024; Yang et al. 2024) also explore RWKV on dense prediction tasks. In this paper, we develop RWKV to exploit its modeling capacity and linear computational efficiency for multi-class unsupervised, making it an attractive solution for 3D point cloud learning applications.

Method

The overall pipeline of PointRWKV is shown in Figure 3(a), where we encode the point cloud by a hierarchical network architecture. Given an input point cloud, we first employ the multi-scale masking strategy (Zhang et al. 2022) to sample different point numbers at various scales. Then a lightweight PointNet (Qi et al. 2017a) is applied to embed point patches and generate token embeddings. These point tokens are consumed by the block-stacked encoder, namely the PRWKV block, where each block consists of two parallel branches for hierarchical local and global feature aggregation.

Hierarchical Point Cloud Learning

Taking an input point cloud $\mathbf{P} \in \mathbb{R}^{N\times 3}$, we apply the multi-scale masking to obtain the M scales point clouds. Starting with the point cloud as the initial M-th scale, the process involves iterative downsampling and grouping using Furthest Point Sampling (FPS) and k Nearest-Neighbour (k-NN). This results in a series of scales, each with a decreasing number of points. Then, a significant proportion of seed points at the final scale are randomly masked, leaving a subset of visible points. These visible points are then back-projected across all scales to ensure consistency. This involves retrieving the k nearest neighbors from the subsequent scale's indices to serve as the visible positions for the current scale, with the remaining positions masked. The process yields visible and masked positions for all scales, providing a comprehensive multi-scale representation of the original point cloud. Finally, we employ a mini-PointNet to extract features for each local patch of different scales and add the positional encoding to get the final sequence, serving as the initial token embeddings $\mathbf{E_0} \in \mathbb{R}^{T\times C}$.

PRWKV Block. After obtaining the token embeddings, we feed them into the encoder, containing a series of PRWKV blocks with long skip connections between shallow and deep

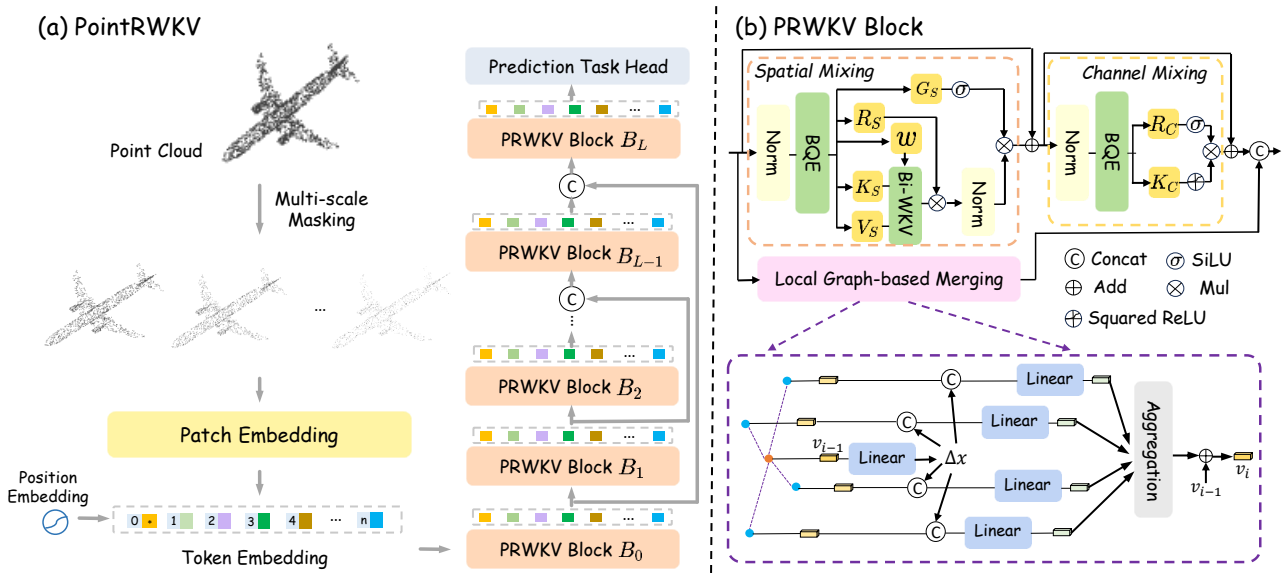

Figure 3: **Overview of the proposed PointRWKV**, which employs a hierarchical architecture to encode multi-scale point cloud features. The whole framework is composed of a series of PRWKV blocks which include the integrative feature modulation branch and the local graph-based merging branch to form the parallel feature learning strategy.

layers. Specifically, for each PRWKV block, as shown in 3(b), the two parallel branches processing strategy is employed to aggregate local and global features. The upper is the *integrative feature modulation (IFM)* flow with spatial-mixing and channel-mixing, and the below is the *local graph-based merging (LGM)*. Finally, the concatenation of the two branches is used as the output of each block.

Integrative Feature Modulation (IFM)

The integrative feature modulation branch consists of a spatial-mixing module and a channel-mixing module. The spatial-mixing module operates as an attention mechanism, executing global attention computations of linear complexity, whereas the channel-mixing module functions as a feed-forward network (FFN), facilitating feature fusion along the channel dimension.

Spatial-Mixing. After one pre-LayerNorm, the tokens are first shifted by the bidirectional quadratic expansion (BQE) function and then fed into four parallel linear layers to obtain the multi-headed vectors R_S, K_S,V_S, G_S:

$$\sqcap_S = BQE_{\sqcap}(X)\mathbf{W}^{\mathbf{S}}_{\sqcap}, \sqcap \in \{R, K, V, G\}, \tag{1}$$

where BQE is formulated as :

$$\begin{aligned} BQE_{\mho}(X) &= X + (1 - \mu_{\mho})X^{\star} \\ X^{\star} &= \mathbf{Concat}(X_1, X_2, X_3, X_4), \end{aligned} \tag{2}$$

where $\mho \in \{R, K, V, G, w\}$, $\mu_{\mho}$ is the learnable vector and the $X^{\star}$ is the slicing vector of X, *i.e.*, $X_1 = [b-1, n, 0 : C/4]$, $X_2 = [b+1, n, C/4 : C/2]$, $X_3 = [b, n-1, C/2 : 3C/4]$, $X_4 = [b, n+1, 3C/4 : C]$, b and n are the dimension index of X. The BQE function empowers the attention mechanism to inherently focus on proximate tokens across diverse channels without requiring a considerable increment in FLOPs. This procedure also broadens the receptive field of each token, thereby significantly boosting the token's coverage in the ensuing layers. Further, a new time-varying decay w is calculated by:

$$\begin{aligned} \nu(c) &= \lambda_x + \tanh(c\mathbf{A_x})\mathbf{B_x} \\ w\_BQE(X) &= X + (1 - \nu(BQE_w(X)))X^{\star} \\ d &= \nu(w\_BQE(X)) \\ w &= \exp(-\exp(d), \end{aligned} \tag{3}$$

where λ_x is a trainable vector and $\mathbf{A_x}$, $\mathbf{B_x}$ are both trainable weight matrices.

Then, K_S and V_S are passed to calculate the global attention result wkv with the new decay w. Here, we introduce the linear complexity bidirectional attention mechanism with two modifications: (1) the decay parameter varies independently, which is data-dependent in a dynamic manner, and (2) the upper limit of original RWKV attention is extended from the current token t to the last token $T-1$ in the summation formula to ensure that all tokens are mutually visible in the calculation of each result. For the t-th token, the attention result is calculated by the following formula:

$$\begin{aligned} wkv_t &= diag(u_1, \ldots, u_n) \cdot K^{\mathsf{T}}_{S,t} \cdot V_{S,t} \\ &+ \sum_{i=1}^{T-1} diag(\varpi^1_{i,j}, \ldots, \varpi^n_{i,j}) \cdot K_{S,i} \cdot V_{S,i} \end{aligned} \tag{4}$$

where u is a per-channel learned boost and $\varpi_{i,j} = \prod_{j=1}^{i-1} w_j$ is a dynamic decay. The final probability output O_S is :

$$O_S = \mathbf{Concat}(\sigma(G_S) \otimes LayerNorm(R_S \otimes wkv))\mathbf{W}^{\mathbf{S}}_{\mathbf{O}} \tag{5}$$

where σ is the SiLU activation function and $\otimes$ is the element-wise multiplication.

Channel-Mixing. The tokens from the spatial-mixing module are further passed into the channel-mixing module. Similarly, the pre-LayerNorm is utilized and R_C, K_C are obtained after the BQE operation:

$$\cap_C = BQE_{\cap}(X)\mathbf{W}^{\mathbf{C}}_{\cap}, \cap \in \{R, K\}. \quad (6)$$

Afterward, a linear projection and a gate mechanism are performed, respectively. And the final output O_C is formulated as:

$$O_C = (\sigma(R_C) \otimes SquaredReLU(K_C)\mathbf{W}^{\mathbf{C}}_{\mathbf{V}})\mathbf{W}^{\mathbf{C}}_{\mathbf{O}}. \quad (7)$$

Local Graph-based Merging (LGM)

Local geometric features have been proven to be crucial for point cloud feature learning, but RWKV's global receptive field cannot comprehensively capture local point geometry, limiting its ability to learn fine-grained features. We encode the point cloud directly into a graph, using the points as the vertices of the graph. The edges of the graph connect neighboring points that fall within a set radius, enabling the transfer of feature information between these points. This graph representation can adapt to the structure of a point cloud without the necessity to regularize it. Furthermore, to minimize the translation variance in the local graph, we incorporate a graph stabilizer mechanism. This mechanism permits points to align their coordinates based on their unique features, improving the overall effectiveness of the network.

Graph Construction. Given the point cloud $\mathbf{P} = (p_1, p_2, ..., p_N)$, we construct a graph $\mathbf{G} = (\mathbf{P}, \mathbf{E})$ by using $\mathbf{P}$ as the vertices and connecting a point to its neighbors within a fixed radius r using the well-known fixed radius near-neighbors search algorithm (Bentley, Stanat, and Williams Jr 1977), where $\mathbf{E} = \{(p_i, p_j) | \|x_i - x_j\|_2 < r\}$. Notably, given a cut-off distance, we utilize a cell list to find point pairs with a runtime complexity of $O(cN)$ where c is the max number of neighbors within the radius, which means this process will not increase much computation complexity.

Graph Stabilizer Mechanism. Typically, we can refine the vertex features by aggregating features along the edges within a graph neural network. In our scenario, we aim to include a vertex's local information about the object where the vertex belongs. So in the $(t+1)$-th iteration, we use the relative coordinates of the neighbors for the edge feature extraction, which can be denoted as:

$$v_i^{t+1} = g^t(\varrho(f^t(x_j - x_i, v_j^t) | (i,j) \in \mathbf{E}), v_i^t), \quad (8)$$

where v^t is the vertex feature from the t-th iteration, $f^t(\cdot)$ computes the edge feature between two vertices, $g^t(\cdot)$ updates the vertex features, and $\varrho(\cdot)$ is a set function that is used to accumulate the edge features for each vertex.

Despite its effectiveness, it remains susceptible to translation within the local vicinity. A slight translation added to a vertex does not significantly alter the local structure of its neighboring vertices. However, it modifies the relative coordinates of these neighbors, thereby escalating the input variance to function $f^t(\cdot)$. To diminish this translation variance, we propose the alignment of neighboring coordinates based on their structural features rather than relying on the central vertex coordinates. Since the central vertex already contains some structural features from the previous iteration, it can be used to estimate an alignment offset, prompting us to design a graph stabilizer mechanism. The Equation 8 can be rewritten as:

$$\begin{aligned} \Delta x_i^t &= h^t(v_i^t) \\ v_i^{t+1} &= g^t(\varrho(f^t(x_j - x_i + \Delta x_i^t, v_j^t) | (i,j) \in \mathbf{E}), v_i^t), \end{aligned} \quad (9)$$

where $h^t(\cdot)$ calculates the offset using the center vertex feature value from the last iteration and Δx_i^t is the coordination offset for the vertices to stabilize their coordinates. As shown in Figure 3(b), we model $f^t(\cdot)$, $g^t(\cdot)$ and $h^t(\cdot)$ by multi-layer perception (MLP) and add a residual connection in $g^t(\cdot)$.

Experiments

To validate the effectiveness of our proposed PointRWKV, we conduct the following experiments: a) pre-training our model on ShapeNet (Chang et al. 2015) training set, b) evaluating our pre-trained model on various downstream tasks, including 3D object classification, 3D part segmentation, and few-shot learning, c) ablating on various strategies.

Comparison on Downstream Tasks

Real-World Object Classification on ScanObjectNN Dataset. ScanObjectNN (Uy et al. 2019), a highly challenging 3D dataset, encompasses approximately 15,000 objects distributed across 15 distinct categories. These objects, derived from real-world indoor environments characterized by cluttered backgrounds, contribute to the heightened complexity inherent in the task of object classification. As shown in Table 1, we conduct experiments on its three variants: OBJ-BG, OBJ-ONLY, and PB-T50-RS. After pre-training, our proposed PointRWKV consistently outperforms transformer- and mamba-based models. Specifically, PointRWKV outperforms previous sota Mamba3D by 2.34%, 3.98% and 4.66% on three variants while reducing 37.2% parameters and 46.2% FLOPs. When compared to the transformer-based methods like Point-BERT, Point-MAE and Point-M2AE, PointRWKV demonstrates a substantial improvement, achieving 10.56%, 8.45% and 7.2% higher accuracy on the challenging PB-T50-RS subset, respectively. Even without the voting strategy (Liu et al. 2019), our PointRWKV still performs better than other methods that are tested with voting strategy.

Synthetic Object Classification on ModelNet40 Dataset. ModelNet40 (Wu et al. 2015) is a CAD 3D object classification dataset, which includes over 12K synthetic 3D CAD models cover 40 categories in total. As shown in Table 1, we report overall accuracy (OA) and mAcc results. PointRWKV maintains its dominance with an OA of 96.89%, which is 1.79% higher than Mamba3D's 95.1%. PointRWKV also shows a 2.89% improvement over Point-M2AE, which has an OA of 94.0%. Similarly, our PointRWKV still outperforms other methods even without the voting strategy (Liu et al. 2019). Overall, these results highlight PointRWKV's superiority over existing transformer- or mamba-based models, achieving multiple SoTA and demonstrating its strength across various settings.

Method	ScanObjectNN			ModelNet40		Params(M) ↓	FLOPs(G) ↓
	OBJ-BG ↑	OBJ-ONLY ↑	PB-T50-RS ↑	OA (%) ↑	mAcc (%) ↑		
MLP/CNN-based							
PointNet (Qi et al. 2017a)	73.3	79.2	68.0	89.2	86.2	3.5	0.5
PointNet++ (Qi et al. 2017b)	82.3	84.3	77.9	90.7	-	1.5	1.7
PointCNN (Li et al. 2018)	86.1	85.5	78.5	92.2	88.1	0.6	0.9
DGCNN (Wang et al. 2019)	82.8	86.2	78.1	92.9	-	1.8	2.4
MVTN (Hamdi, Giancola, and Ghanem 2021)	92.6	92.3	82.8	93.8	-	11.2	43.7
PointMLP (Ma et al. 2022)	-	-	85.4±0.3	94.5	91.4	12.6	31.4
PointNeXt (Qian et al. 2022)	-	-	87.7±0.4	93.2±0.1	90.8±0.2	1.4	1.6
Transformer-based							
Transformer (Vaswani et al. 2017)	79.86	80.55	77.24	91.4	-	22.1	4.8
PointTransformer (Zhao et al. 2021)	-	-	-	93.7	90.6	22.1	4.8
Point-BERT (Yu et al. 2022)	87.43	88.12	83.07	93.2	-	22.1	4.8
Point-MAE (Pang et al. 2022)	90.02	88.29	85.18	93.8	-	22.1	4.8
Point-M2AE (Zhang et al. 2022)	91.22	88.81	86.43	94.0	-	15.3	3.6
Mamba-based							
PCM (Zhang et al. 2024)	-	-	88.10±0.3	93.4±0.2	-	34.2	45.0
PointMamba (Liang et al. 2024)	90.71	88.47	84.87	93.6	-	12.3	3.6
Mamba3D (Han et al. 2024)	95.18	92.60	88.97	95.1	-	16.9	3.9
RWKV-based							
PointRWKV w/o vot.	**96.01**	**95.62**	**93.05**	**96.16**	**92.25**	**10.6**	**2.1**
PointRWKV w/ vot.	**97.52**	**96.58**	**93.63**	**96.89**	**93.08**	**10.6**	**2.1**

Table 1: **Object classification on the ScanObjectNN and ModelNet40 datasets.** We report the overall accuracy (OA, %) of ScanObjectNN on its three variants: OBJ-BG, OBJ-ONLY, and PB-T50-RS, the OA and mAcc of ModelNet40 and the overall params(M) and FLOPs(G).

Method	5-way		10-way	
	10-shot ↑	20-shot ↑	10-shot ↑	20-shot ↑
DGCNN (Wang et al. 2019)	91.8 ± 3.7	93.4 ± 3.2	86.3 ± 6.2	90.9 ± 5.1
DGCNN + OcCo (Wang et al. 2021)	91.9 ± 3.3	93.9 ± 3.1	86.4 ± 5.4	91.3 ± 4.6
Transformer (Yu et al. 2022)	87.8 ± 5.2	93.3 ± 4.3	84.6 ± 5.5	89.4 ± 6.3
Transformer + OcCo (Yu et al. 2022)	94.0 ± 3.6	95.9 ± 2.3	89.4 ± 5.1	92.4 ± 4.6
Point-BERT (Yu et al. 2022)	94.6 ± 3.1	96.3 ± 2.7	91.0 ± 5.4	92.7 ± 5.1
Point-M2AE (Zhang et al. 2022)	96.8 ± 1.8	98.3 ± 1.4	92.3 ± 4.5	95.0 ± 3.0
PointMamba (Liang et al. 2024)	95.0 ± 2.3	97.3 ± 1.8	91.4 ± 4.4	92.8 ± 4.0
Mamba3D (Han et al. 2024)	96.4 ± 2.2	98.2 ± 1.2	92.4 ± 4.1	95.2 ± 2.9
PointRWKV(ours)	**97.9 ± 1.2**	**99.2 ± 0.6**	**94.8 ± 2.8**	**96.7 ± 2.6**
Improvement	+1.1	+0.9	+2.5	+1.5

Table 2: **Few-shot classification on ModelNet40**. We report the average accuracy (%) and standard deviation (%) of 10 independent experiments without voting.

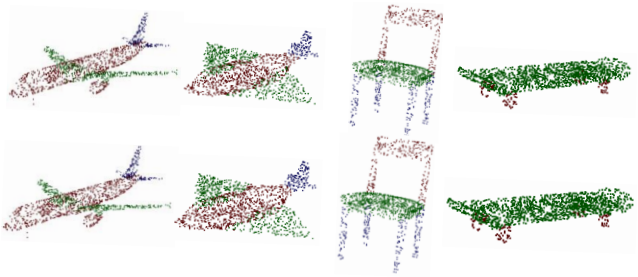

Figure 4: **Part segmentation results on ShapeNetPart.** Top row is ground truth and bottom row is our prediction.

Part Segmentation on ShapeNetPart Dataset. We conduct part segmentation on the challenging ShapeNetPart (Yi et al. 2016) dataset to predict more detailed class labels for each point within a sample. It comprises 16880 models with 16 different shape categories and 50 part labels. Experimental results on the ShapeNetPart dataset are shown in Table 3. We report mean IoU (mIoU) for all classes (Cls.) and all instances (Inst.). Our PointRWKV still obtains 3.16% and 4.27% improvements in terms of Inst. mIoU and Cls. mIoU while significantly reducing parameters and FLOPs. Moreover, we present qualitative results of part segmentation in Figure 4. It can be found that PointRWKV achieves highly competitive results compared to the ground truth.

Few-shot Learning. To further evaluate the performance of PointRWKV with limited fine-tuning data, we conduct experiments for few-shot classification on ModelNet40 with an n-way, m-shot setting, where n is the number of classes randomly sampled from the dataset, and m denotes the number of samples randomly drawn from each class. As shown in Table 2, we experiment with $n \in \{5, 10\}$ and $m \in \{10, 20\}$. PointRWKV surpasses previous methods by 1.1%, 0.9%, 2.5%, and 1.5%, respectively, for all four settings with much smaller deviations. This illustrates PointRWKV's competence in learning semantic information and its potent ability to transfer knowledge to downstream tasks.

Structural Efficiency. To fully explore the the efficiency of our method, we gradually increase the sequence length to test the ability of processing the long point sequence. As shown as Figure 2, when the sequence continues to increase, we reduce the GFLOPs by 15.4× and 2.1× compared with the transformer-based method and the most convincing mamba-based method, achieving true linear complexity. Furthermore, as illustrated in the last two columns of

Method	Backbone	Cls. mIoU (%) ↑	Inst. mIoU (%) ↑	Params(M) ↓	FLOPs(G) ↓
PointNet (Qi et al. 2017a)	MLP-based	80.39	83.7	3.6	4.9
PointNet++ (Qi et al. 2017b)	MLP-based	81.85	85.1	1.0	4.9
PointCNN (Li et al. 2018)	CNN-based	84.6	86.1	-	-
DGCNN (Wang et al. 2019)	CNN-based	82.3	85.2	1.3	12.4
APES (Wu et al. 2023a)	MLP-based	83.67	85.8	-	-
PointMLP (Ma et al. 2022)	MLP-based	84.6	86.1	-	-
PointNeXt (Qian et al. 2022)	MLP-based	85.2±0.1	87.0±0.1	22.5	110.2
Transformer (Vaswani et al. 2017)	Transformer-based	83.4	85.1	27.1	15.5
PointTransformer (Zhao et al. 2021)	Transformer-based	83.7	86.6	-	-
Point-BERT (Yu et al. 2022)	Transformer-based	84.1	85.6	27.1	10.6
Point-MAE (Pang et al. 2022)	Transformer-based	84.2	86.1	27.1	15.5
Point-M2AE (Zhang et al. 2022)	Transformer-based	84.86	86.51	-	-
PCM (Zhang et al. 2024)	Mamba-based	85.6	87.1	34.2	45.0
PointMamba (Liang et al. 2024)	Mamba-based	84.42	86.0	17.4	14.3
Mamba3D (Han et al. 2024)	Mamba-based	84.1	85.7	21.9	9.5
PointRWKV(ours)	RWKV-based	**89.87**	**90.26**	**15.2**	**6.5**

Table 3: **Part segmentation on ShapeNetPart dataset.** The class mIoU and the instance mIoU are reported, with model parameters (M) and FLOPs (G).

Table 1 and Table 3, our PointRWKV reduces 3.1%-13.8% in parameters and 31.6%-41.7% in FLOPs when compared with transformer- and mamba-based counterparts.

Variant	OA	mAcc
base	89.6	88.3
+ BQE	90.4	89.0
+ bi-attention	91.6	90.2
+ Multi-scale	92.8	90.6
+ LGM w/o GS	93.0	91.1
+ LGM w/ GS	93.6	91.7

Table 4: Ablation of main components.

Number	OA	mAcc
0	91.2	90.5
1	92.1	91.2
2	92.4	91.4
3	92.5	91.7
4	92.3	91.5

Table 5: Ablation of graph iteration.

Ablation Study

Analysis of Different Components. We illustrate the importance of components of our network by adding different parts. The baseline is to only use single scale of 1024 points as input and remove the BQE, bidirectional attention mechanism, the LGM and graph stabilizer (GS) mechanism. As shown in Table 4, we add the shift by bidirectional quadratic expansion (BQE) function and the modified bidirectional attention mechanism, enhancing OA almost equally, respectively. We observe that applying the hierarchical multi-scale point cloud learning and local graph-based merging can lift the performance by a large margin, which demonstrates the importance of refined feature learning. And the absence of graph stabilizer hurts the performance, which proves the necessity of local graph adjustments.

Ablations of Different Number of Graph Iteration. In local graph-based merging branch,we refine the vertex states iteratively. In Table 5, we study the impact of the number of iterations on the accuracy. The initial vertex state alone achieves the lowest accuracy since it only has a small receptive field around the vertex. As the number of iterations increases, the corresponding accuracy value also becomes higher compared to no iteration. This largely illustrates the effectiveness of the iterative graph stabilizer.

Scales	OA	mAcc
512	90.3	88.9
1024	89.6	88.3
2048	90.2	89.5
4096	89.7	88.8
512-1024-2048	92.8	91.6
1024-2048-4096	91.5	90.1

Table 6: Ablation of different point scales.

Ablations of Different Point Scales. In Table 6, we experiment with different scales of points as input to encode the token embeddings. Apart from the original single 1024 points input, we also sample different points and combine adjacent scale to form the multi-scales embeddings. It can be observed that single-scale input fluctuates in accuracy, but multi-scale input has more outstanding final performance.

Conclusion

In this paper, we introduce PointRWKV, a novel RWKV-based architecture for point cloud learning. With a hierarchical architecture, PointRWKV learns to produce powerful 3D representations by encoding multi-scale point clouds. To facilitate local and global feature aggregation, we design the parallel feature merging strategy. Experimental results show that PointRWKV exhibits superior performance over transformer- and mamba-based counterparts on different point cloud learning datasets while significantly reducing parameters and FLOPs. With its linear complexity capability, we hope PointRWKV will serve as an efficient and cost-effective baseline for more 3D tasks.

References

Anthony, Q.; Tokpanov, Y.; Glorioso, P.; and Millidge, B. 2024. BlackMamba: Mixture of Experts for State-Space Models. *arXiv preprint arXiv:2402.01771*.

Atzmon, M.; Maron, H.; and Lipman, Y. 2018. Point convolutional neural networks by extension operators. *arXiv preprint arXiv:1803.10091*.

Bentley, J. L.; Stanat, D. F.; and Williams Jr, E. H. 1977. The complexity of finding fixed-radius near neighbors. *Information processing letters*, 6(6): 209–212.

Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

Choy, C.; Gwak, J.; and Savarese, S. 2019. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, 3075–3084.

Deng, X.; Zhang, W.; Ding, Q.; and Zhang, X. 2023. Pointvector: a vector representation in point cloud analysis. In *CVPR*, 9455–9465.

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*.

Duan, Y.; Wang, W.; Chen, Z.; Zhu, X.; Lu, L.; Lu, T.; Qiao, Y.; Li, H.; Dai, J.; and Wang, W. 2024. Vision-rwkv: Efficient and scalable visual perception with rwkv-like architectures. *arXiv preprint arXiv:2403.02308*.

Fu, D. Y.; Dao, T.; Saab, K. K.; Thomas, A. W.; Rudra, A.; and Ré, C. 2022. Hungry hungry hippos: Towards language modeling with state space models. In *ICLR*.

Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

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: 187–199.

Hamdi, A.; Giancola, S.; and Ghanem, B. 2021. Mvtn: Multi-view transformation network for 3d shape recognition. In *CVPR*, 1–11.

Han, X.; Tang, Y.; Wang, Z.; and Li, X. 2024. Mamba3d: Enhancing local features for 3d point cloud analysis via state space model. *arXiv preprint arXiv:2404.14966*.

Hatamizadeh, A.; Tang, Y.; Nath, V.; Yang, D.; Myronenko, A.; Landman, B.; Roth, H. R.; and Xu, D. 2022. Unetr: Transformers for 3d medical image segmentation. In *WiCV*, 574–584.

He, H.; Bai, Y.; Zhang, J.; He, Q.; Chen, H.; Gan, Z.; Wang, C.; Li, X.; Tian, G.; and Xie, L. 2024a. Mambaad: Exploring state space models for multi-class unsupervised anomaly detection. *arXiv preprint arXiv:2404.06564*.

He, H.; Zhang, J.; Cai, Y.; Chen, H.; Hu, X.; Gan, Z.; Wang, Y.; Wang, C.; Wu, Y.; and Xie, L. 2024b. MobileMamba: Lightweight Multi-Receptive Visual Mamba Network. *arXiv preprint arXiv:2411.15941*.

He, Q.; Wang, Z.; Zeng, H.; Zeng, Y.; and Liu, Y. 2022. Svga-net: Sparse voxel-graph attention network for 3d object detection from point clouds. In *AAAI*, volume 36, 870–878.

Huang, T.; Zhang, J.; Chen, J.; Ding, Z.; Tai, Y.; Zhang, Z.; Wang, C.; and Liu, Y. 2022a. 3QNet: 3D Point Cloud Geometry Quantization Compression Network. *ACM TOG*.

Huang, T.; Zou, H.; Cui, J.; Zhang, J.; Yang, X.; Li, L.; and Liu, Y. 2022b. Adaptive recurrent forward network for dense point cloud completion. *TMM*.

Komarichev, A.; Zhong, Z.; and Hua, J. 2019. A-cnn: Annularly convolutional neural networks on point clouds. In *CVPR*, 7421–7430.

Landrieu, L.; and Simonovsky, M. 2018. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*, 4558–4567.

Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; and Chen, B. 2018. Pointcnn: Convolution on x-transformed points. *NeurIPS*, 31.

Liang, D.; Zhou, X.; Wang, X.; Zhu, X.; Xu, W.; Zou, Z.; Ye, X.; and Bai, X. 2024. PointMamba: A Simple State Space Model for Point Cloud Analysis. *arXiv preprint arXiv:2402.10739*.

Liu, J.; Yu, R.; Wang, Y.; Zheng, Y.; Deng, T.; Ye, W.; and Wang, H. 2024a. Point mamba: A novel point cloud backbone based on state space model with octree-based ordering strategy. *arXiv preprint arXiv:2403.06467*.

Liu, Y.; Fan, B.; Xiang, S.; and Pan, C. 2019. Relation-shape convolutional neural network for point cloud analysis. In *CVPR*, 8895–8904.

Liu, Y.; Tian, Y.; Zhao, Y.; Yu, H.; Xie, L.; Wang, Y.; Ye, Q.; and Liu, Y. 2024b. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 10012–10022.

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*.

Mao, J.; Shi, S.; Wang, X.; and Li, H. 2022. 3d object detection for autonomous driving: A review and new outlooks. *arXiv preprint arXiv:2206.09474*, 1.

Mao, J.; Xue, Y.; Niu, M.; Bai, H.; Feng, J.; Liang, X.; Xu, H.; and Xu, C. 2021. Voxel transformer for 3d object detection. In *ICCV*, 3164–3173.

Mehta, H.; Gupta, A.; Cutkosky, A.; and Neyshabur, B. 2022. Long range language modeling via gated state spaces. In *ICLR*.

Misra, I.; Girdhar, R.; and Joulin, A. 2021. An end-to-end transformer model for 3d object detection. In *ICCV*, 2906–2917.

Pang, Y.; Wang, W.; Tay, F. E.; Liu, W.; Tian, Y.; and Yuan, L. 2022. Masked autoencoders for point cloud self-supervised learning. In *ECCV*, 604–621. Springer.

Park, C.; Jeong, Y.; Cho, M.; and Park, J. 2022. Fast point transformer. In *CVPR*, 16949–16958.

Peng, B.; Alcaide, E.; Anthony, Q.; Albalak, A.; Arcadinho, S.; Cao, H.; Cheng, X.; Chung, M.; Grella, M.; GV, K. K.; et al. 2023. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*.

Peng, B.; Goldstein, D.; Anthony, Q.; Albalak, A.; Alcaide, E.; Biderman, S.; Cheah, E.; Ferdinan, T.; Hou, H.; Kazienko, P.; et al. 2024. Eagle and Finch: RWKV with matrix-valued states and dynamic recurrence. *arXiv preprint arXiv:2404.05892*.

Pióro, M.; Ciebiera, K.; Król, K.; Ludziejewski, J.; and Jaszczur, S. 2024. Moe-mamba: Efficient selective state space models with mixture of experts. *arXiv preprint arXiv:2401.04081*.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 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. *NeurIPS*, 30.

Qi, Z.; Dong, R.; Fan, G.; Ge, Z.; Zhang, X.; Ma, K.; and Yi, L. 2023. Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. In *ICML*, 28223–28243. PMLR.

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. *NeurIPS*, 35: 23192–23204.

Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. 2018. Improving language understanding by generative pre-training.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, 234–241. Springer.

Shafiullah, N. M. M.; Paxton, C.; Pinto, L.; Chintala, S.; and Szlam, A. 2022. Clip-fields: Weakly supervised semantic fields for robotic memory. *arXiv preprint arXiv:2210.05663*.

Smith, J. T.; Warrington, A.; and Linderman, S. W. 2022. Simplified state space layers for sequence modeling. In *ICLR*.

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 *ICCV*, 6411–6420.

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 *ICCV*, 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. *NeurIPS*, 30.

Wang, H.; Liu, Q.; Yue, X.; Lasenby, J.; and Kusner, M. J. 2021. Unsupervised point cloud pre-training via occlusion completion. In *ICCV*, 9782–9792.

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 TOG*, 38(5): 1–12.

Wu, C.; Zheng, J.; Pfrommer, J.; and Beyerer, J. 2023a. Attention-based point cloud edge sampling. In *CVPR*, 5333–5343.

Wu, W.; Qi, Z.; and Fuxin, L. 2019. Pointconv: Deep convolutional networks on 3d point clouds. In *CVPR*, 9621–9630.

Wu, X.; Jiang, L.; Wang, P.-S.; Liu, Z.; Liu, X.; Qiao, Y.; Ouyang, W.; He, T.; and Zhao, H. 2023b. Point transformer v3: Simpler, faster, stronger. *arXiv preprint arXiv:2312.10035*.

Wu, X.; Lao, Y.; Jiang, L.; Liu, X.; and Zhao, H. 2022. Point transformer v2: Grouped vector attention and partition-based pooling. *NeurIPS*, 35: 33330–33342.

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 *CVPR*, 1912–1920.

Yang, Y.; Xing, Z.; and Zhu, L. 2024. Vivim: a video vision mamba for medical video object segmentation. *arXiv preprint arXiv:2401.14168*.

Yang, Z.; Zhang, H.; Zhao, D.; Wei, B.; and Xu, Y. 2024. Restore-RWKV: Efficient and Effective Medical Image Restoration with RWKV. arXiv:2407.11087.

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 ToG*, 35(6): 1–12.

Yu, X.; Rao, Y.; Wang, Z.; Liu, Z.; Lu, J.; and Zhou, J. 2021. Pointr: Diverse point cloud completion with geometry-aware transformers. In *ICCV*, 12498–12507.

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 *CVPR*, 19313–19322.

Yuan, H.; Li, X.; Qi, L.; Zhang, T.; Yang, M.-H.; Yan, S.; and Loy, C. C. 2024. Mamba or RWKV: Exploring High-Quality and High-Efficiency Segment Anything Model. *arXiv preprint arXiv:2406.19369*.

Zhang, R.; Guo, Z.; Gao, P.; Fang, R.; Zhao, B.; Wang, D.; Qiao, Y.; and Li, H. 2022. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. *NeurIPS*, 35: 27061–27074.

Zhang, T.; Li, X.; Yuan, H.; Ji, S.; and Yan, S. 2024. Point Could Mamba: Point Cloud Learning via State Space Model. *arXiv preprint arXiv:2403.00762*.

Zhao, H.; Jiang, L.; Fu, C.-W.; and Jia, J. 2019. Pointweb: Enhancing local neighborhood features for point cloud processing. In *CVPR*, 5565–5573.

Zhao, H.; Jiang, L.; Jia, J.; Torr, P. H.; and Koltun, V. 2021. Point transformer. In *ICCV*, 16259–16268.