Point Linear Self-attention Based on Larger-scale Context with Asymmetric Online Distillation for 3D Semantic Segmentation

Qinghe Liu

May 5, 2025

To address the issues of low computational efficiency and limited receptive fields in current Transformer-based point cloud semantic segmentation models when processing large-scale point clouds, this research proposes a 3D point cloud semantic segmentation method combining a linear self-attention mechanism with asymmetric online knowledge distillation. This method aims to bridge the efficiency gap between state-of-the-art and lightweight models. First, a linear self-attention mechanism specifically for point clouds is designed. Its core lies in decoupling the computational complexity of self-attention from the receptive field size ($O(\|P\| \cdot k) \rightarrow O(\|P\|)$) through specific kernel function designs combined with point cloud positional encoding. This allows the model to significantly reduce the computational burden while expanding the receptive field to effectively fuse local geometry with Larger-scale context. Second, to address the potential performance degradation caused by linear attention, as well as the inherent disorder, sparsity, and ”geometric shortcut” problems of point cloud data, an asymmetric online knowledge distillation framework is proposed. This framework combines self-supervised learning tasks with knowledge transfer between teacher-student models. By using asymmetric inputs (different cropping ratios, added noise) and specific loss functions, the robust feature representations learned by the teacher model are effectively transferred to the lightweight linear-attention student model. This overcomes the geometric shortcut problem and enhances the student model’s feature learning capability. This research anticipates constructing an efficient point cloud semantic segmentation model that is competitive in both segmentation accuracy and computational efficiency. This provides a feasible solution for applying the technology in resource-constrained scenarios such as autonomous driving, robotics, and AR/VR, thereby promoting technological development in related fields.

1 Introduction

1.1 Background

With the development of point cloud deep learning, especially the proposal of indoor and outdoor datasets [1]–[6] and key point cloud methods [7]–[12], the application of point clouds in semantic segmentation, object detection, 3D reconstruction, and other fields has been greatly promoted. As one of the fundamental tasks in 3D computer vision, the development of point cloud semantic segmentation often influences the progress in classification, detection, reconstruction, etc., particularly in high-precision scenarios. However, according to recent research trends, the advancement of state-of-the-art models for point cloud semantic segmentation is often accompanied by a sharp increase in model parameters and computational cost, especially with the proposal and development of point cloud transformers [12]–[16]. This has led to a widening efficiency gap between advanced models and lightweight, readily deployable models. To mitigate this issue, researchers have proposed various methods, such as lightweight Encoder-Decoder architectures [11], [17], [18], model pruning [19], model quantization [20], optimization of self-attention mechanism efficiency [14], [21]–[28], and knowledge distillation [18], [29]–[31].

Optimization of self-attention mechanism efficiency has been a research direction since the introduction of self-attention [32]. It mainly focuses on improving the capture efficiency of self-attention [33]–[36] by partitioning and shrinking the receptive field, as well as reducing feature grouping, thereby decreasing the amount of data processed in parallel and enhancing the efficiency of obtaining self-attention. Concurrently, there are optimizations targeting the quadratic time complexity of self-attention [24]–[28], which employ kernel features, kernel functions, and various simple functions combined with activation functions to approximate softmax, thus reducing the time complexity to linear and significantly decreasing the model's computational cost.

Knowledge distillation, as one of the most popular and effective model compression methods currently in computer vision and natural language processing [37], has its core idea [29] in learning from a large, complex teacher model to train a small, lightweight student model. This method achieves knowledge transfer from teacher to student by comparing the intermediate layer outputs and final outputs of the teacher model, thereby pre-training the student model. The pre-trained student model is then fine-tuned on downstream tasks, ultimately enabling the student model to perform comparably to, or even better than, the teacher model. Through long-term theoretical analysis and experimental validation, this method currently has a solid theoretical basis [19], [29], [38], [39] and empirical performance [37], and has been widely applied in fields such as image classification and semantic segmentation.

Inspired by the two aforementioned research directions, this study proposes a point cloud semantic segmentation model based on linear self-attention and asymmetric knowledge distillation: On one hand, knowledge distillation is used to effectively transfer knowledge from high-performance, high-complexity traditional point cloud self-attention models at various learning stages to an efficient, lightweight linear self-attention model. On the other hand, by designing a linear self-attention mechanism based on point clouds, the computational efficiency of the model will be optimized, the receptive field for global semantic information will be expanded, and the model's computational burden will be reduced.Finally, the model will improve the model's operational efficiency while enhancing segmentation accuracy.

1.2 Problem Statement

Although the self-attention mechanism has demonstrated powerful feature learning capabilities in point cloud semantic segmentation, its use of a local self-attention mechanism often prevents the model from explicitly processing global information when dealing with large-scale point clouds; it must instead be acquired gradually through pyramid-style encoding layers. This not only leads to typically deep encoder architectures but also the time complexity of local self-attention is sensitive to the receptive field size, which limits the ability of self-attention to directly capture global semantic information and imposes high requirements on device resources for self-attention-based point cloud semantic segmentation models.

To pursue higher efficiency, the linear self-attention mechanism offers a promising direction. By decoupling the computational time complexity of the local self-attention mechanism from the receptive field size, it not only facilitates expanding the receptive field per layer and reducing model depth but also enhances computational efficiency. However, directly applying linear self-attention mechanisms to raw point clouds faces the following two challenges:

Performance Maintenance Difficulty: Compared to traditional self-attention mechanisms, linear self-attention, while simplifying computation, may have a reduced capacity to capture complex inter-point relationships and long-term dependencies, leading to a performance gap compared to state-of-the-art models. How to effectively narrow this gap, enabling linear models to achieve or even surpass the performance of complex models, has become a pressing issue.

Robustness Challenge in Feature Learning: Compared to data like images and language, point cloud data possesses characteristics of disorderliness and sparsity [7], causing its semantic segmentation effectiveness to rely on human-designed geometric relationships [31], for example, using algorithms like k-NN, serialization, voxelization, or projection to generate geometric structures. Moreover, models without sufficient design considerations can easily fall into the "geometric shortcut" trap [18]: during semantic segmentation, they excessively rely on simple geometric cues such as normals and density, while ignoring the semantic information of the point cloud scene. This further hinders the performance improvement of linear models. Under these circumstances, this research must ensure that the designed linear attention mechanism is not only computationally efficient but can also effectively integrate local geometric structures and global context, learning feature representations that are genuinely robust for the semantic segmentation task.

To ensure that the linear self-attention mechanism can approach or even exceed the performance of traditional self-attention mechanisms, and to enable it to learn robust feature representations for point clouds, we introduce the method of knowledge distillation. However, due to the inherent challenges of point clouds (disorderliness, sparsity, geometric shortcut problem), directly transferring distillation frameworks and loss functions from the image domain can lead to severe feature loss. Therefore, the design of the distillation framework and the distillation loss function becomes one of the key aspects of this research.

In summary, the core problem of this research lies in: how to design an effective mechanism such that, while maintaining the efficiency advantages of linear attention, the model can overcome the inherent challenges of point clouds and fully inherit the powerful feature representation capabilities of complex teacher models through strategies like knowledge distillation, ultimately achieving high-efficiency and high-accuracy semantic segmentation of raw point clouds.

1.3 Objectives

The objectives of this research are as follows:

1. **Design a linear self-attention mechanism for point clouds**: Construct a self-attention mechanism whose time complexity is decoupled from the receptive field size. This mechanism should establish appropriate geometric structures to capture geometric features, addressing the disorderliness and sparsity of point clouds, and effectively fuse geometric features with global context.

2. **Design an asymmetric online knowledge distillation framework**: Propose an asymmetric online distillation strategy aimed at transferring the robust feature representations learned by a teacher model based on traditional self-attention to a linear self-attention student model. This is intended to compensate for the potential performance loss caused by simplified computation and to overcome the geometric shortcut problem.

3. **Construct a high-accuracy, high-efficiency point cloud semantic segmentation model**: Fine-tune the distilled linear self-attention student model combined with an output layer on downstream semantic segmentation tasks to obtain an end-to-end deep learning model for raw point cloud semantic segmentation.

4. **Conduct a comprehensive evaluation of the linear self-attention model**: Systematically evaluate the segmentation accuracy (using OA and mIoU) and computational efficiency (parameter count, inference speed, FLOPs) of the proposed model on multiple standard 3D point cloud semantic segmentation datasets [1]–[4], and compare it with current state-of-the-art methods.

5. **Validate the effectiveness of the method**: Through ablation studies, verify the impact and contribution of the designed linear self-attention mechanism, the distillation method, and other individual modules to the overall performance.

1.4 Significance

This research focuses on the application challenges of current transformer-based point cloud semantic segmentation models in large-scale point clouds and resource-constrained scenarios. The proposed linear self-attention mechanism for raw point clouds holds the promise of decoupling time complexity from receptive field size, potentially improving both accuracy and computational efficiency, thus paving the way for further deployment of high-precision point cloud segmentation models.

Furthermore, to bridge the performance gap between linear self-attention and traditional self-attention, this study explores an asymmetric online distillation strategy to transfer the robust feature representations from complex teacher models to efficient linear self-attention student models. This not only offers the potential to close the performance gap but could even enable lightweight models to achieve or exceed the accuracy of complex models, providing a novel and effective pathway for constructing point cloud analysis models that balance high efficiency with high performance.

Moreover, this research pioneers the combination of these two advanced techniques, applying them to the fundamental and crucial task of 3D semantic segmentation. Exploring their synergistic potential within the point cloud domain holds significant exploratory value.

Finally, by achieving high-efficiency, high-accuracy point cloud semantic segmentation, the outcomes of this research can directly advance cutting-edge applications such as environmental perception in autonomous driving systems, robot navigation and interaction, and augmented/virtual reality scene understanding, demonstrating substantial application value.

2 Related Work

This part introduces the development status of self-attention mechanisms, knowledge distillation, and point cloud semantic segmentation relevant to this research, and elaborates on the value and some issues of current work.

2.1 Self-Attention Mechanism

2.1.1 Vanilla Self-Attention Mechanism

The standard transformer [32] has achieved tremendous success in the field of computer vision. By capturing long-range contextual dependencies, it has largely surpassed the effectiveness of CNNs and RNNs in visual feature capture. However, due to its quadratic time complexity $O(N^2)$, the computational cost is significant when processing high-resolution images or large-scale point clouds.

Liu *et al.*[21] proposed using local windows and window shifting to limit the transformer's field of view, thereby reducing the computational load. Wang *et al.*[22] reduced computation by using a feature pyramid with progressive reduction. Although these methods improve efficiency, they result in a reduced field of view for the transformer and do not fundamentally solve the bottleneck of the self-attention mechanism's quadratic time complexity.

2.1.2 Self-Attention on Point Clouds

Zhao *et al.*[12] first introduced the self-attention mechanism into point clouds and, combined with the inherent characteristics of point clouds, used MLPs for positional encoding, enabling self-attention to better capture local geometric information. Lai *et al.*[13], building upon the multi-head self-attention mechanism, introduced contextual positional encoding and a stratified key sampling strategy, providing inspiration for transformers to further capture geometric information at both local and global scales in point clouds. Wu *et al.*[14] addressed the overfitting problem caused by the drastic parameter increase in [12]'s vector attention as model depth and channel numbers grow, by proposing Grouped Vector Attention. By introducing an additional position encoding multiplier and Partition-based Pooling, it made point cloud-based self-attention computation more efficient and utilization of geometric information more complete. Wu *et al.*[15] further simplified the computational efficiency of self-attention for capturing geometric information through Point Cloud Serialization and improved accuracy. Most of these methods employ local self-attention mechanisms

[21] to optimize computational efficiency in large-scale point cloud scenarios. This, on one hand, limits the model's receptive field size, making it easier to overlook global semantic information. On the other hand, excessive network depth and lower computational efficiency restrict deployment in resource-constrained situations.

2.1.3 Linear Self-Attention Mechanism

Linear self-attention aims to replace the Softmax operation with kernel-based similarities, thereby reducing the complexity from $O(N^2)$ to $O(N)$. Early work, Katharopoulos *et al.*[24], proposed using linear dot products of kernelized feature maps. Shen *et al.*[40] used $ReLU(\cdot)$ as the kernel function to implement linear attention. Qin *et al.*[27] combined $ReLU(\cdot)$ with a cosine-based re-weighting mechanism, aiming to enhance the locality inductive biases of self-attention weights. Most of these methods discard negative feature maps due to $ReLU(\cdot)$, and the resulting attention weights tend to be uniform (high entropy), lacking distinctiveness. Han *et al.*[28] extended $ReLU(\cdot)$, introducing a power operation to approximate the exponential function for feature processing, aiming to maintain both non-negativity and low entropy characteristics of attention weights, but the issue of negative value loss persisted. Kacham *et al.*[41] demonstrated through Empirical Demonstration and Behavioral Similarity that polynomial kernels can practically replace softmax in terms of performance. Meng *et al.*[42] proved the existence of a class of element-wise functions $g(\cdot)$ (requiring first derivative $g'(x) > 0$ and second derivative $g''(x) > 0$) that can be applied to feature maps to reduce the entropy of the attention distribution. In practice, by explicitly handling positive-negative interactions and using learnable power functions to lower entropy, they mitigated the issues of negative value loss and high entropy. However, the work of [41], [42] might have achieved performance maintenance or even superiority only on a few specific types of image datasets, rather than being a universal method for fitting traditional self-attention across all data forms. Furthermore, solely using linear attention mechanisms to approximate traditional attention and achieve comparable or superior performance across different datasets is very difficult. Therefore, it is necessary to introduce techniques like knowledge distillation to optimize the learning of linear self-attention.

2.2 Knowledge Distillation

The concept of model compression, which involves using a smaller, faster model to mimic the functionality of a large, complex, high-performing but slow model, was proposed by Buciluă *et al.*[38] in 2006. With the rise of deep learning, the depth of neural networks increased rapidly, bringing the concept of model compression back into researchers' focus. Ba *et al.*[30] demonstrated that shallow networks can learn by matching the logits of deep teacher networks, without needing soft targets. Around the same time, Hinton *et al.*[29] formally proposed the concept of knowledge distillation. This method introduces a temperature parameter to soften the probability outputs, enabling the student model to learn the soft targets from the teacher model's output, thereby compressing the teacher model into a lightweight student model. Subsequently, knowledge distillation expanded in various directions. Romero *et al.*[43] first introduced feature-based distillation, using hints from the intermediate layers of the teacher model to further guide the student model. Zagoruyko *et al.*[44] proposed the concept of attention transfer, using attention maps derived from feature maps as knowledge. Yim *et al.*[45] first proposed relation-based distillation, utilizing the FSP Matrix as the relationship between different feature maps. Zhang *et al.*[46] first proposed Deep Mutual Learning, the concept of online distillation, where each student model learns by matching the soft targets of other students. These methods laid the foundation for modern distillation techniques.

2.2.1 Knowledge Distillation on 3D Semantic Segmentation

Knowledge distillation in the **point cloud semantic segmentation domain** has received limited research attention and primarily focuses on structured point clouds (voxels, projections). Zhang *et al.*[47] addressed the weakly supervised large-scale point cloud semantic segmentation problem with only a few labeled points by constructing a perturbed branch to apply random and learnable transformations to the input point cloud, forcing consistency with its original branch prediction, and using JS divergence for distillation. Hou *et al.*[48] combined point-level and voxel-level distillation, as well as inter-point and inter-voxel relation distillation, to significantly compress the model while maintaining competitive performance. Chen *et al.*[49] performed completely unsupervised 3D point cloud semantic segmentation through 2D-to-3D cross-modal distillation and supervoxel clustering. Liu *et al.*[50] first attempted to use 2D Vision Foundation Models for self-supervised representation learning on 3D large-scale point cloud sequences, utilizing 2D-3D correspondences through VFM-assisted superpixels and superpoints for contrastive learning and cross-modal knowledge distillation. Yu *et al.*[51] proposed Object-level Denoising Projection for 2D CLIP feature extraction and 3D feature aggregation as the teacher model, designed Multimodal Distillation Learning (MDL) with Object-centered Constraints for the student model, and used feature loss and label loss for supervision. Most of these distillation methods are characterized by overly simple distillation losses and relatively naive distillation scheme designs, which can easily lead point clouds into geometric shortcuts, resulting in the loss of semantic information. Furthermore, they often employ inflexible structured point clouds, incurring additional overhead for voxel or projection computation while causing pixel-level feature loss. Therefore, exploring semantic segmentation based on knowledge distillation for raw point clouds holds significant value.

2.3 3D Semantic Segmentation on Point Clouds

Semantic segmentation based on raw point clouds began with PointNet [7]. This work highlighted the inherent characteristics of point clouds, such as disorderliness and rotation invariance, and employed shared multi-layer perceptrons (MLPs) as the fundamental network unit to overcome the network design challenges posed by these characteristics. However, this work only mapped the high-dimensional features from the encoder output to each point using a single MLP, resulting in significant feature loss. Qi *et al.*[8] borrowed the shared MLP concept from PointNet as the basic unit and adopted a U-Net structure, constructing an Encoder-Decoder architecture. This allowed the low-resolution point clouds with high-dimensional features from the encoder output to be gradually restored to high resolution through a pyramid-shaped decoder. This work laid the foundation for point cloud semantic segmentation architectures: point clouds progressively pass through encoder layers to generate low-resolution, high-dimensional feature point clouds, and then through decoder layers to recover high-resolution, low-dimensional feature representations of the original point cloud, finally outputting segmentation probabilities through a classifier.

Subsequent developments in point cloud semantic segmentation primarily focused on optimizing the shared MLP [52]–[58]. These methods did not employ self-attention mechanisms, making it difficult to capture contextual dependencies in point clouds. After 2021, with the proposal of Point Transformer [12], research in point cloud semantic segmentation concentrated on improving the point cloud self-attention mechanism [13]–[16], [59]. Additionally, there have been efforts to optimize the PointNet++-based architecture [8] through data augmentation and designing robust point cloud data representations [60], [61]. Wu *et al.*[18]'s designed Sonata is currently the best-performing model in point cloud semantic segmentation. It innovatively introduced the concept of self-supervised learning to point cloud semantic segmentation. Through tasks like reconstructing masked and cropped point cloud features using teacher and student models, and calculating Sinkhorn-Knopp centering between the teacher's and student's outputs, it enables the student model to learn robust feature representations of point clouds. However, because these methods all utilize local self-attention mechanisms, they limit the receptive field size while also reducing computational efficiency.

3 Proposed Method

This section first elaborates on the feasibility and optimization goals of the design, then provides a detailed description of the research's design for the linear self-attention mechanism based on raw point clouds, the asymmetric knowledge distillation strategy based on raw point clouds, and the main architecture of the model.

3.1 Feasibility Analysis and Optimization Goals

Let the input point cloud be X, the student model parameters be denoted by θ_S, and the teacher model parameters by θ_T. The models are $S(X;\theta_S)$ and $T(X;\theta_T)$ respectively; the soft or hard labels for the semantic segmentation task are Y. The Universal Approximation Theorem (UAT) [62] states that, theoretically, neural networks with sufficient capacity are universal function approximators. This implies that for the function $f_T(X) = T(X;\theta_T^*)$ learned by the teacher model (θ_T^* being the trained parameters), there exists, in principle, a set of parameters θ_S such that the student model $S(X;\theta_S)$ can approximate $f_T(X)$ to arbitrary precision, provided that the capacity of model S is sufficient. However, in reality, due to the differences between linear and non-linear attention mechanisms, the capacity of S is strictly less than that of T. Therefore, as pointed out in the proof by Keles *et al.*[63], it is difficult to find or design a linear self-attention mechanism that adapts to the point cloud feature representations across the entire vector space. On the other hand, although S cannot fully approximate T, UAT indicates that S can still perfectly approximate T within a specific vector space. Furthermore, [41], [42] show through extensive experiments that for a specific task, there almost certainly exists such a special solution parameter interval $[\theta_{S_1}, \theta_{S_2}]$ such that $S(X;\theta_S) \approx f_T(X)$, and the size of the parameter solution space is negatively correlated with the size of the vector space. This suggests that it is theoretically feasible for linear self-attention to fully approximate traditional self-attention on specific tasks.

However, limited by the depth of neural networks, it is difficult to find an optimal approximation solution parameter for traditional self-attention mechanisms using only traditional supervised learning when the vector space is too large, i.e., in tasks with large-scale, high-dimensional data. Therefore, certain constraints must be introduced (in this study, we set the knowledge distillation loss function as L_{KD} as the constraint). Thus, the optimization goal of this research is as follows:

$$\theta_S^* = \arg\min_{\theta_S} L_{KD}(T(X;\theta_T^*), S(X;\theta_S)) \tag{1}$$

By using a specially designed knowledge distillation method, it is feasible to converge constrained linear self-attention to traditional self-attention, thereby achieving the effect of a high-performance, high-efficiency point cloud segmentation model.

3.2 Linear Attention

The traditional self-attention mechanism [32] can be represented by the following formula:

$$\text{attention} = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{2}$$

where $Q, K, V \in R^{N\times d}$ represent the Query, Key, and Value matrices, respectively. Its vector form is:

$$\text{attention}_t = \frac{\sum_{i=1}^{N} \exp\left(q_t k_i^T/\sqrt{d}\right)}{\sum_{j=1}^{N} \exp\left(q_t k_j^T/\sqrt{d}\right)} v_i \tag{3}$$

where q_t represents the t-th Query vector of Q, k_i represents the i-th Key vector, v_i represents the i-th Value vector, N is the sequence length, and d is the feature dimension of each attention head. Equations (2) and (3) show that calculating QK^T has a complexity of $O(N^2)$. Katharopoulos *et al.*[24] proposed the generalized self-attention formula:

$$v_t^{'} = \frac{\sum_{i=1}^{N} \text{sim}(q_t, k_i^T)}{\sum_{j=1}^{N} \text{sim}(q_t, k_j^T)} v_i \tag{4}$$

viewing the $\exp\left(\cdot/\sqrt{d}\right)$ in equations (2) and (3) as a special similarity function $\text{sim}(\cdot,\cdot)$. Subsequently, the similarity function $\text{sim}(q_t, k_i)$ is represented as a kernel function $k(q_t, k_i)$, and feature mapping $\phi(\cdot)(R^D \to R^C)$ is used such that $k(x, y) = \phi(x)^T\phi(y)$. Through this rewriting and the associative property of multiplication, it can be simplified to:

$$v_t^{'} = \frac{\phi(q_t)^T \sum_{j=1}^{N} \phi(k_j)v_j^T}{\phi(q_t)^T \sum_{j=1}^{N} \phi(k_j)} \tag{5}$$

Since it only needs to compute $\sum_{i=1}^{N} \phi(k_i)v_i^T$ and $\sum_{j=1}^{N} \phi(k_j)$ once, the computational complexity of this form is $O(N)$. This formula indicates that the main task of kernel-based linear self-attention mechanisms lies in designing appropriate kernel functions. Inspired by the work of [42], [63], and considering the current issues of negative value loss and high entropy in kernel functions, the linear self-attention mechanism in this research is based on the design of [42] as follows:

3.2.1 Solving the Problem of Negative Feature Map Loss

PolaFormer [42] proposes that q and k can be decomposed as follows:

$$q = q^+ - q^-, \quad k = k^+ - k^- \tag{6}$$

where $q_i^+ = \max(q_i, 0)$, $q_i^- = \max(-q_i, 0)$, and the decomposition for k is similar. Substituting equation (6) into the vector inner product calculation yields:

$$\langle q, k\rangle = \langle q^+, k^+\rangle + \langle q^-, k^-\rangle - \big(\langle q^+, k^-\rangle + \langle q^-, k^+\rangle\big) \tag{7}$$

Thus, the similarity function can be decomposed as:

$$\text{sim}(q, k) = \big(\phi(q^+)\phi(k^+)^T + \phi(q^-)\phi(k^-)^T\big) - \big(\phi(q^+)\phi(k^-)^T + \phi(q^-)\phi(k^+)^T\big) \tag{8}$$

where $\phi(\cdot)$ is the kernel function. This operation explicitly handles negative values, thereby preventing negative value loss. To avoid the instability of the subtraction operation, a mixed learning approach is adopted to obtain the self-attention. The final self-attention output is:

$$\text{attention}_t = \left[\frac{\phi([q_t^+; q_t^-])\sum_{i=1}^N \phi([k_i^+; k_i^-])^T v_i^s}{\phi([q_t^+; q_t^-])\sum_{j=1}^N \phi([k_j^+; k_j^-])^T} \odot G_s; \frac{\phi([q_t^+; q_t^-])\sum_{i=1}^N \phi([k_i^-; k_i^+])^T v_i^o}{\phi([q_t^+; q_t^-])\sum_{j=1}^N \phi([k_j^-; k_j^+])^T} \odot G_o\right] \tag{9}$$

where $[;]$ denotes the concatenation operation, $v_i = [v_i^s; v_i^o]$, and G_s, G_o are learnable parameters.

3.2.2 Kernel Function Design

The kernel function cannot simply be chosen as ReLU, because this would lead to a decrease in sharpness and cause the feature map distribution to become too flat, i.e., the high entropy problem [28]. Keles *et al.*[63] pointed out that the Taylor series $e^x \approx \sum_{k=0}^{p} \frac{x^k}{k!}$ can be used to approximate Softmax dot-product attention, and its p-th order approximation can be computed in $O(nd_q^p d_v)$. However, directly using this method to approximate $\exp(\cdot)$ requires a high order p to meet accuracy requirements, but typically when p is slightly larger, its computational efficiency becomes worse than the traditional self-attention mechanism. Influenced by this, this research, building upon ReLU, introduces a g-th order odd-degree learnable polynomial, similar to a Taylor polynomial, to simplify computation while ensuring the sharpness of the feature map:

$$P(x) = \sum_{i=1}^{g} w_i x^{p_i} + \text{bias} \tag{10}$$

where $x = \text{ReLU}(k^+)$ or $\text{ReLU}(-k^-)$, and similarly for q. $p_i = 2i - 1$ indicates performing a p_i-th power operation on each element of x, w_i is used to dynamically activate a specific odd power function for different elements, and bias is a vector used to dynamically disperse the element distribution when data is scarce. To ensure that at least one odd power function is active, this research introduces the following regularization term:

$$L_P = \sum_{i=1}^{g} \sum_{j=1, i\neq j}^{g} \frac{\|w_i\|\|w_j\|}{\|w_i - w_j\|^2} \tag{11}$$

3.2.3 Point Cloud Linear Attention

To overcome the inherent challenges of point clouds (disorderliness, sparsity, and the "geometric shortcut" problem), the method for obtaining Q, K, V differs from that used for images, and positional encoding must be introduced [12]. Let the point cloud set be $P, P \subset R^{n\times 3}$. For any element $p_t \in P$, its feature is $f_t \in R^C$ (where C is the feature dimension). By constructing a geometric structure, a subset $P' \subset P$ representing its geometric information is obtained. Let

$p_j \in P'$, its feature is $f_j \in R^C$. U_q, U_k, U_v are linear layers. $\delta_{mul}, \delta_{bias}, \gamma$ are multi-layer perceptrons (MLPs). Then Q, K, V can be obtained through the following formula:

$$q_t = U_q(f_t); \quad k_j = U_k(f_j); \quad v_j = U_v(f_j); \tag{12}$$

Note that q_t, k_j, v_j here are scalars, not vectors. The position encoding-based self-attention mechanism (ungrouped) from Point Transformer V2 [14] can be represented as:

$$\text{attention}_t = \text{Softmax}(\delta_{mul}(p_t - p_j) \odot \gamma(q_t, k_j) + \delta_{bias}(p_t - p_j)) \odot v_j \tag{13}$$

Let $r_{tj} = \delta_{mul}(p_t - p_j) \odot \gamma(q_t, k_j) + \delta_{bias}(p_t - p_j)$. Then, the scalar form of this attention mechanism is:

$$\text{attention}_t = \frac{\sum_{j=1}^{N} \exp\left(r_{tj}\right)}{\sum_{j=1}^{N} \exp\left(r_{tj}\right)} v_j \tag{14}$$

where $N = \|P'\|$. Combining formula (5) and the meaning of point cloud positional encoding, this research performs the following decomposition:

$$\text{attention}_t = \frac{\phi(q_t) \sum_{j=1}^{N} \left[\gamma(\phi(k_j) v_j) \odot \delta_{mul}(p_t - p_j) + \delta_{bias}(p_t - p_j)\right]}{\phi(q_t) \sum_{j=1}^{N} \left[\phi(k_j) \odot \delta_{mul}(p_t - p_j) + \delta_{bias}(p_t - p_j)\right]} \tag{15}$$

Further combining with formula (9), and letting $\eta_s^{[+;-]} = \sum_{j=1}^{N} \left[\gamma(\phi([k_j^+; k_j^-]) v_j^s) \odot \delta_{mul}(p_t - p_j) + \delta_{bias}(p_t - p_j)\right]$; $\xi^{[+;-]} = \sum_{j=1}^{N} \left[\phi([k_j^+; k_j^-]) \odot \delta_{mul}(p_t - p_j) + \delta_{bias}(p_t - p_j)\right]$, we obtain the final linear self-attention mechanism based on raw point clouds:

$$\text{attention}_t = \left[\frac{\phi([q_t^+; q_t^-]) \eta_s^{[+;-]}}{\phi([q_t^+; q_t^-]) \xi^{[+;-]}} \odot G_s; \frac{\phi([q_t^+; q_t^-]) \eta_o^{[-;+]}}{\phi([q_t^+; q_t^-]) \xi^{[-;+]}} \odot G_o\right] \tag{16}$$

This formula shows that the time complexity of linear self-attention is $O(\|P\|)$ rather than $O(\|P\| \cdot N)$, indicating that N is decoupled from the time complexity. Therefore, increasing the receptive field does not lead to a multiplicative increase in time complexity. Furthermore, adopting the polar decomposition method from PolaFormer [42] ensures that negative values are not lost while maintaining low entropy in the feature map.

3.3 Model Architecture

3.3.1 Backbone

Inspired by Sonata [18], the model does not adopt a decoder structure but replaces it with a linear output layer. This allows the model to focus more on capturing semantic information during pre-training tasks, rather than fusion. Compared to point cloud transformers like [13]–[15] which use four encoding layers as the encoder, this model proposes using two or three encoder layers, reducing the number of transformer modules included in each encoder layer, and significantly increasing the receptive field of each transformer block. Except for the first encoder layer, which consists of an embedding layer and several transformer blocks, each subsequent encoder layer consists of a downsampling layer and several transformer blocks. Furthermore, feature concatenation is performed using Sonata's "up_cast" method.

3.3.2 Point Embedding

Before entering the encoder structure, the point cloud is first arranged into a one-dimensional sequence using serialization ideas [64] through space-filling curves such as Z-order curves and Hilbert curves, preserving spatial locality to some extent. This method facilitates the computation of local self-attention. Finally, a sparse convolutional architecture maps the point cloud features and performs positional encoding, making it easier for the encoder to extract features.

3.3.3 Downsampling Layer

Adopting the voxelization idea from [14], the serialized point cloud is downsampled while performing feature fusion: the point cloud is divided into several point blocks according to a one-dimensional grid, and the features of points within each grid are aggregated by taking the maximum value along each feature dimension as the superpoint feature. Finally, a linear layer maps the features to a higher dimension.

3.3.4 Transformer Block

First, the serialization and deserialization indices of the point cloud are obtained. Then, Q, K, V are output through three linear layers. The obtained QKV are organized into sequences using the serialization index. Different from Point Transformer V3, when organizing groups, it not only uses a larger scale but also obtains a larger scale group through downsampling and feature fusion. This research believes that this method not only allows attention to acquire a larger receptive field but, more importantly, enables attention to focus more on semantic information rather than geometric shortcuts by comparing and extracting features from local and global groups. Finally, the two groups are concatenated, multi-head self-attention is computed, and added to the input features via a residual connection.

3.3.5 Output Block

The features from each level for every point are concatenated. Then, segmentation probabilities are output through a linear layer, a normalization layer, and a softmax function.

3.4 Asymmetric Knowledge Distillation Strategy

To better utilize current large-scale point cloud datasets for training, this research adopts a distillation scheme combining self-supervised learning with a teacher-student model framework, pre-training the main model through supervision and teacher-student model interaction.

3.4.1 Teacher-Student Architecture

Student Model: As the main model of this research, it uses linear self-attention and the architecture described above. Its gradients originate from two sources: one is the gradient obtained from the self-supervised learning loss L_{self}, and the other is the gradient obtained from the comparison loss with the teacher model, $L_{teacher}$. The weights of these two gradients change during the training process:

$$L = \alpha L_{self} + \beta L_{teacher} \tag{17}$$

This research proposes to use the parameters α and β such that in the early stages of training, the student model's gradient mainly comes from the teacher, while in the later stages, its gradient mainly comes from supervised learning. This approach helps the student acquire transferred knowledge from the teacher model early on and prevents overfitting to the teacher's knowledge through self-supervised learning. On the other hand, designing the self-supervised learning scheme can also enable the student model to acquire semantic information more directly.

Teacher Model: As the auxiliary model for distillation, it is proposed to use Sonata [18] as the main architecture and employ online distillation to train alongside the student model. Its gradient source comes entirely from supervised learning.

3.4.2 Self-Supervised Learning

Preprocessing Downsampling: In the dataset preprocessing part, after applying data augmentation operations such as rotation, scaling, translation, illumination enhancement, cropping, etc., to the point cloud data, a downsampling process similar to 3.3.3 is performed on the data, but without feature fusion, only randomly retaining a few points within the grid. Afterwards, the information of the sampled points is retained, the indices of the point clouds in the same grid are recorded, and these points are removed from the original point cloud.

Upsampling to Recover Point Cloud Information: The original point cloud high-dimensional features output from the teacher or student model are linearly interpolated [8] to the retained sampled points to obtain high-dimensional sampled point features. These are then fed into a pre-trained decoder (linear layer or MLP) to predict the non-coordinate information of the retained sampled points, obtaining a predicted view and the original view. Online clustering and Sinkhorn-Knopp are used to obtain the logits for the predicted view and the original view. Finally, two predictions, P_p, P_o, are output through softmax with temperature [65], and the loss is calculated using cross-entropy.

3.4.3 Knowledge Transfer

To make the student focus more on local geometric information, different cropping ratios are applied to the point cloud data for the teacher and student models. Typically, the student retains only a local point cloud view, while the teacher retains a global point cloud view. Additionally, Gaussian noise is added to each dimension of the points sampled by the student to reduce the degree of overfitting. Knowledge transfer between the teacher and student models occurs by comparing the output high-dimensional point cloud features, using the same loss calculation method as in section 3.4.2. However, the loss calculation is performed directly on the high-dimensional features without dimensionality reduction through a decoder. In this way, knowledge learned by the teacher model at various stages is transferred.

4 Experimental Design

This chapter will detail the experimental plan designed to verify the effectiveness and efficiency of the proposed method on the 3D semantic segmentation task. We will evaluate the model performance on standard benchmark datasets and compare it with state-of-the-art methods. Furthermore, thorough ablation studies will be conducted to analyze the contribution of each component, particularly evaluating the similarities and differences between the linear self-attention mechanism and the traditional self-attention mechanism. Of course, some details may be expanded or changed as the experiments progress.

4.1 Datasets

- **Main Evaluation Datasets**
  - ScanNet (v2): A widely used indoor scene semantic segmentation benchmark, following standard train/validation splits.
  - S3DIS: Contains 6 large indoor areas, using Area 5 for testing or 6-fold cross-validation.
- **Pre-training Datasets**: Adopting a large-scale pre-training strategy similar to Sonata [18], integrating datasets including ScanNet (v2), S3DIS, ArkitScenes, HM3D, Structured3D, etc., for pre-training to improve the model's generalization ability.

4.2 Evaluation Protocols and Metrics

To comprehensively evaluate the quality of the representations learned by the model, we will use multi-dimensional evaluation protocols:

- **Linear Probing**: Freeze the pre-trained/distilled encoder and train only a linear classification head (parameter count $< 0.2\%$) for downstream tasks (like semantic segmentation). This is the core standard for evaluating the quality of distilled learned representations, directly reflecting the semantic discrimination ability of the encoder features and reducing interference from geometric shortcuts.
- **Decoder Probing**: Freeze the pre-trained/distilled encoder, reintroduce and train only a lightweight decoder (e.g., parameter count approx. 10-15% of the total). Evaluate the performance potential of the encoder features when combined with a small number of task-related parameters, serving as an intermediate step between linear probing and full fine-tuning.
- **Full Fine-tuning**: Unfreeze the entire model (including encoder and decoder/output layer) and fine-tune end-to-end on the downstream task data to evaluate the best performance the model can achieve on that specific task.
- Use OA and mIoU as core metrics.
- Efficiency Evaluation: Parameters, FLOPs, Inference Speed.

4.3 Comparison Methods

Based on different technical routes and current progress, this research will compare with the following methods:

- Traditional methods: PointNet++ [8], KPConv [10].

- Transformer-based methods: Point Transformer v2 [14], Point Transformer v3 [15], Stratified Transformer [13], Swin3D [16].
- Distillation-related methods:
  - Sonata: As the teacher model, and also as the SOTA benchmark, especially its linear probing results.
  - MSC (Masked Scene Contrast)
  - PointContrast (PC)
- In addition to the above methods, other advanced methods such as PointVector [61] and PointNeXt [60] will be introduced later as needed for experiments.

4.4 Ablation Experiments

Design rigorous ablation experiments to verify the contribution of each module:

- **Core Component Validation:**
  - **Linear Attention vs. Standard Attention**: Under the same architecture and distillation framework, replace linear attention with standard self-attention (e.g., the attention from Point Transformer V2/V3) for comparison.
  - **Distillation vs. No Distillation**: Compare the performance of the complete model with that of a student model using only linear attention trained with standard supervision.
- **Linear Attention Design Analysis:**
  - **Polar Decomposition**: Remove or simplify the mechanism for handling positive and negative features to evaluate its impact.
  - **Kernel Function Design**: Compare the polynomial kernel with other linear kernels (like ReLU) or polynomials of fixed degrees.
  - **Point Cloud Specific Design**: Remove or simplify the positional encoding and attention modifications specific to point clouds to assess their contribution.
- **Asymmetric Distillation Strategy Analysis:**
  - **Self-Supervised Task:** Remove the self-supervised part, relying solely on teacher distillation.
  - **Teacher Distillation:** Remove the teacher distillation part, relying solely on the self-supervised task.
  - **Asymmetric Input:** Compare the effects of using symmetric inputs (same cropping/augmentation) versus asymmetric inputs.
- **Architecture Choice Validation:**
  - Compare the effects of different encoder depths.
  - Compare the effects of setting different receptive field sizes.
  - Validate the contribution of larger-scale grouping when computing attention.

5 Expected Outcomes and Contributions

This research anticipates producing the following outcomes and contributing to the field of point cloud semantic segmentation and related applications:

1. Propose a novel **point cloud-based linear self-attention mechanism**: This mechanism can effectively handle the disorderliness and sparsity of point clouds and integrate local geometry with larger-scale context. Its core contribution lies in decoupling the time complexity from the receptive field size, allowing for a significant reduction in computational burden while expanding the receptive field, thereby improving model efficiency.

2. Construct a **high-efficiency and high-accuracy point cloud semantic segmentation model**: By combining the designed linear self-attention mechanism with the asymmetric distillation strategy, it is expected to build a model that is competitive in segmentation accuracy while outperforming existing state-of-the-art Transformer or Transformer-like methods in computational efficiency (parameter count, inference speed, FLOPs).

3. Propose an effective **asymmetric online knowledge distillation framework**: This framework, through the combination of self-supervised learning and teacher-student model distillation, can effectively transfer the robust feature representations learned by a complex teacher model to a lightweight linear self-attention mechanism student model.

4. Provide a feasible solution for exploring **point cloud applications in resource-constrained scenarios**: The high-efficiency model proposed by this research is expected to lower the deployment threshold for point cloud semantic segmentation technology on resource-constrained platforms such as autonomous driving, robotics, and AR/VR, promoting the implementation and development of these cutting-edge applications.

This research aims to break the efficiency bottleneck commonly found in current high-performance point cloud segmentation models. Through innovative linear self-attention design and knowledge distillation strategies, it seeks to achieve a balance between accuracy and efficiency, providing a promising solution for large-scale, high-accuracy, real-time point cloud semantic segmentation, and promoting the practical application of related technologies.

6 Research Plan and Timeline

First Stage (Months 1-2): Literature Review and Foundational Preparation

- Gain in-depth understanding of the latest progress in related fields (linear attention, knowledge distillation, point cloud segmentation).
- Familiarize with the components of the latest version of PyTorch and reproduce project code from relevant papers.
- Complete downloading, preprocessing, and format unification for datasets (ScanNet, S3DIS, etc.).
- Preliminarily set up the experimental environment and code framework.

Second Stage (Months 3-5): Core Algorithm Design and Implementation

- Implement the point cloud-based linear self-attention mechanism (including polar decomposition, kernel function design, positional encoding integration).
- Design and implement the asymmetric online knowledge distillation framework (including Teacher-Student architecture, self-supervised tasks, knowledge transfer loss).
- First draft of the paper: Methods section.

Third Stage (Months 6-8): Model Construction and Preliminary Training

- Construct the complete semantic segmentation model backbone (including embedding layer, downsampling layer, Transformer Block, output layer).
- Implement the pre-training pipeline for large-scale datasets.
- Complete the reproduction and training of the teacher model (Sonata).
- Conduct preliminary distillation pre-training experiments.

Fourth Stage (Months 9-12): Model Tuning and Comprehensive Evaluation

- Perform comprehensive evaluations on benchmark datasets like ScanNet, S3DIS (Linear Probing, Decoder Probing, Full Fine-tuning).
- Compare performance and efficiency with SOTA methods.
- Analyze experimental results, adjust model parameters and training strategies.

Fifth Stage (Months 13-15): Ablation Experiments and Analysis

- Design and complete detailed ablation experiments to validate the effectiveness of each core component (linear attention, distillation strategy, architecture choices, etc.).
- Deeply analyze the performance differences between linear attention and standard attention.
- Write the experimental results and analysis sections.

Sixth Stage (Months 16-18): Paper Writing and Conclusion

- Complete the writing and polishing of the research paper.
- Prepare the paper for submission.
- Organize and open-source the relevant code.

References

[1] I. Armeni, O. Sener, A. R. Zamir, *et al.*, "3d semantic parsing of large-scale indoor spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1534–1543.

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

[3] C. Yeshwanth, Y.-C. Liu, M. Nießner, and A. Dai, "Scannet++: A high-fidelity dataset of 3d indoor scenes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 12–22.

[4] G. Baruch, Z. Chen, A. Dehghan, *et al.*, "Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data," *arXiv preprint arXiv:2111.08897*, 2021.

[5] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, *et al.*, "Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai," *arXiv preprint arXiv:2109.08238*, 2021.

[6] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The international journal of robotics research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[7] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, *Pointnet: Deep learning on point sets for 3d classification and segmentation*, 2017. arXiv: `1612.00593 [cs.CV]`. [Online]. Available: `https://arxiv.org/abs/1612.00593`.

[8] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, *Pointnet++: Deep hierarchical feature learning on point sets in a metric space*, 2017. arXiv: `1706.02413 [cs.CV]`. [Online]. Available: `https://arxiv.org/abs/1706.02413`.

[9] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, *Dynamic graph cnn for learning on point clouds*, 2019. arXiv: `1801.07829 [cs.CV]`. [Online]. Available: `https://arxiv.org/abs/1801.07829`.

[10] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, *Kpconv: Flexible and deformable convolution for point clouds*, 2019. arXiv: `1904.08889 [cs.CV]`. [Online]. Available: `https://arxiv.org/abs/1904.08889`.

[11] Q. Hu, B. Yang, L. Xie, *et al.*, *Randla-net: Efficient semantic segmentation of large-scale point clouds*, 2020. arXiv: `1911.11236 [cs.CV]`. [Online]. Available: `https://arxiv.org/abs/1911.11236`.

[12] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, *Point transformer*, 2021. arXiv: `2012.09164 [cs.CV]`. [Online]. Available: `https://arxiv.org/abs/2012.09164`.

[13] X. Lai, J. Liu, L. Jiang, *et al.*, "Stratified transformer for 3d point cloud segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 8500–8509.

[14] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point transformer v2: Grouped vector attention and partition-based pooling," *Advances in Neural Information Processing Systems*, vol. 35, pp. 33 330–33 342, 2022.

[15] X. Wu, L. Jiang, P.-S. Wang, *et al.*, “Point transformer v3: Simpler faster stronger,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 4840–4851.

[16] Y.-Q. Yang, Y.-X. Guo, J.-Y. Xiong, *et al.*, “Swin3d: A pretrained transformer backbone for 3d indoor scene understanding,” *Computational Visual Media*, vol. 11, no. 1, pp. 83–101, 2025.

[17] T. Cortinhal, G. Tzelepis, and E. Erdal Aksoy, “Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds,” in *Advances in Visual Computing: 15th International Symposium, ISVC 2020, San Diego, CA, USA, October 5–7, 2020, Proceedings, Part II 15*, Springer, 2020, pp. 207–222.

[18] X. Wu, D. DeTone, D. Frost, *et al.*, “Sonata: Self-supervised learning of reliable point representations,” *arXiv preprint arXiv:2503.16429*, 2025.

[19] S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding,” *arXiv preprint arXiv:1510.00149*, 2015.

[20] J. Choi, Z. Wang, S. Venkataramani, P. I.-J. Chuang, V. Srinivasan, and K. Gopalakrishnan, “Pact: Parameterized clipping activation for quantized neural networks,” *arXiv preprint arXiv:1805.06085*, 2018.

[21] Z. Liu, Y. Lin, Y. Cao, *et al.*, *Swin transformer: Hierarchical vision transformer using shifted windows*, 2021. arXiv: `2103.14030 [cs.CV]`. [Online]. Available: `https://arxiv.org/abs/2103.14030`.

[22] W. Wang, E. Xie, X. Li, *et al.*, “Pyramid vision transformer: A versatile backbone for dense prediction without convolutions,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 568–578.

[23] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” in *International conference on machine learning*, PMLR, 2021, pp. 10 347–10 357.

[24] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, “Transformers are rnns: Fast autoregressive transformers with linear attention,” in *International conference on machine learning*, PMLR, 2020, pp. 5156–5165.

[25] K. Choromanski, V. Likhosherstov, D. Dohan, *et al.*, “Rethinking attention with performers,” *arXiv preprint arXiv:2009.14794*, 2020.

[26] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, “Linformer: Self-attention with linear complexity,” *arXiv preprint arXiv:2006.04768*, 2020.

[27] Z. Qin, W. Sun, H. Deng, *et al.*, “Cosformer: Rethinking softmax in attention,” *arXiv preprint arXiv:2202.08791*, 2022.

[28] D. Han, X. Pan, Y. Han, S. Song, and G. Huang, “Flatten transformer: Vision transformer using focused linear attention,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 5961–5971.

[29] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*,

[30] J. Ba and R. Caruana, “Do deep nets really need to be deep?” *Advances in neural information processing systems*, vol. 27, 2014.

[31] L. Zhang, R. Dong, H.-S. Tai, and K. Ma, “Pointdistiller: Structured knowledge distillation towards efficient and compact 3d detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 21 791–21 801.

[32] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[33] J. Ainslie, J. Lee-Thorp, M. De Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai, “Gqa: Training generalized multi-query transformer models from multi-head checkpoints,” *arXiv preprint arXiv:2305.13245*, 2023.

[34] R. Child, S. Gray, A. Radford, and I. Sutskever, “Generating long sequences with sparse transformers,” *arXiv preprint arXiv:1904.10509*, 2019.

[35] I. Beltagy, M. E. Peters, and A. Cohan, “Longformer: The long-document transformer,” *arXiv preprint arXiv:2004.05150*, 2020.

[36] Z. Liu, Y. Lin, Y. Cao, *et al.*, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[37] J. Gou, B. Yu, S. J. Maybank, and D. Tao, “Knowledge distillation: A survey,” *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.

[38] C. Buciluă, R. Caruana, and A. Niculescu-Mizil, “Model compression,” in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 535–541.

[39] L. Frank and J. Davis, “What makes a good dataset for knowledge distillation?” *arXiv preprint arXiv:2411.12817*, 2024.

[40] Z. Shen, M. Zhang, H. Zhao, S. Yi, and H. Li, “Efficient attention: Attention with linear complexities,” in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 3531–3539.

[41] P. Kacham, V. Mirrokni, and P. Zhong, “Polysketchformer: Fast transformers via sketching polynomial kernels,” *arXiv preprint arXiv:2310.01655*, 2023.

[42] W. Meng, Y. Luo, X. Li, D. Jiang, and Z. Zhang, “Polaformer: Polarity-aware linear attention for vision transformers,” *arXiv preprint arXiv:2501.15061*, 2025.

[43] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, “Fitnets: Hints for thin deep nets,” *arXiv preprint arXiv:1412.6550*, 2014.

[44] S. Zagoruyko and N. Komodakis, “Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer,” *arXiv preprint arXiv:1612.03928*, 2016.

[45] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4133–4141.

[46] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4320–4328.

[47] Y. Zhang, Y. Qu, Y. Xie, Z. Li, S. Zheng, and C. Li, "Perturbed self-distillation: Weakly supervised large-scale point cloud semantic segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 15 520–15 528.

[48] Y. Hou, X. Zhu, Y. Ma, C. C. Loy, and Y. Li, "Point-to-voxel knowledge distillation for lidar semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 8479–8488.

[49] Z. Chen, H. Xu, W. Chen, *et al.*, "Pointdc: Unsupervised semantic segmentation of 3d point clouds via cross-modal distillation and super-voxel clustering," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 14 290–14 299.

[50] Y. Liu, L. Kong, J. Cen, *et al.*, "Segment any point cloud sequences by distilling vision foundation models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 37 193–37 229, 2023.

[51] F. Yu, R. Tian, Z. Wang, X. Wang, and X. Liang, "Cus3d: Clip-based unsupervised 3d segmentation via object-level denoise," in *2024 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2024, pp. 1–6.

[52] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," *Advances in neural information processing systems*, vol. 31, 2018.

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

[54] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2019, pp. 9621–9630.

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

[56] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia, "Pointweb: Enhancing local neighborhood features for point cloud processing," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5565–5573.

[57] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, "Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5589–5598.

[58] M. Xu, R. Ding, H. Zhao, and X. Qi, "Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3173–3182.

[59] P.-S. Wang, “Octformer: Octree-based transformers for 3d point clouds,” *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–11, 2023.

[60] G. Qian, Y. Li, H. Peng, *et al.*, “Pointnext: Revisiting pointnet++ with improved training and scaling strategies,” *Advances in neural information processing systems*, vol. 35, pp. 23 192–23 204, 2022.

[61] X. Deng, W. Zhang, Q. Ding, and X. Zhang, “Pointvector: A vector representation in point cloud analysis,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 9455–9465.

[62] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[63] F. D. Keles, P. M. Wijewardena, and C. Hegde, “On the computational complexity of self-attention,” in *International conference on algorithmic learning theory*, PMLR, 2023, pp. 597–619.

[64] T. Wang, W. Wen, J. Zhai, K. Xu, and H. Luo, “Serialized point mamba: A serialized point cloud mamba segmentation model,” *arXiv preprint arXiv:2407.12319*, 2024.

[65] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” *Advances in neural information processing systems*, vol. 33, pp. 9912–9924, 2020.