DeepLA-Net: Very Deep Local Aggregation Networks for Point Cloud Analysis

Ziyin Zeng[1], Mingyue Dong[1], Jian Zhou[1]*, Huan Qiu[1], Zhen Dong[1], Man Luo[2] and Bijun Li[1]
[1]Wuhan University, [2] Dongfeng USharing Technology Co.
{zengziyin, dongmy, jianzhou, huanqiu, dongzhenwhu, lee}@whu.edu.cn, tc-luoman@dfmc.com.cn

Abstract

Due to the irregular and disordered data structure in 3D point clouds, prior works have focused on designing more sophisticated local representation methods to capture these complex local patterns. However, the recognition performance has saturated over the past few years, indicating that increasingly complex and redundant designs no longer make improvements to local learning. This phenomenon prompts us to diverge from the trend in 3D vision and instead pursue an alternative and successful solution: deeper neural networks. In this paper, we propose DeepLA-Net, a series of very deep networks for point cloud analysis. The key insight of our approach is to exploit a small but mighty local learning block, which uses 10× fewer FLOPs, enabling the construction of very deep networks. Furthermore, we design a training supervision strategy to ensure smooth gradient backpropagation and optimization in very deep networks. We construct the DeepLA-Net family with a depth of up to 120 blocks — at least 5× deeper than recent methods — trained on a single RTX 3090. An ensemble of the DeepLA-Net achieves state-of-the-art performance on classification and segmentation tasks of S3DIS Area5 (+2.2% mIoU), ScanNet test set (+1.6% mIoU), ScanObjectNN (+2.1% OA), and ShapeNetPart (+0.9% cls.mIoU).

1. Introduction

Thanks to the significant development of 3D sensors, 3D point cloud analysis has garnered increasing attention in recent years and is widely applied in autonomous driving, smart cities, and robotics [22, 105]. However, 3D point clouds consist of sparse, discrete, and non-uniform point sets embedded in continuous space [56]. This complex structure poses challenges to local pattern learning.

In order to solve this problem, extensive explorations have been conducted on the local pattern of point clouds to enable fine-grained analysis. Inspired by CNNs, they introduce two robust inductive biases: locality and weight sharing, which promote the development of neural networks for analyzing 3D point clouds [57]. Specifically, these studies [12, 25, 42, 59, 70, 104, 106, 107] employ MLPs with shared weights to capture local features from neighbors. We collectively refer to them as local aggregation networks (LANets) in this paper. It is noteworthy that most LANets primarily focus on developing intricate local representation methods to explicitly explore the local patterns of point clouds, achieving success due to the robust inductive biases. However, the performance of increasingly sophisticated LANets on popular benchmarks [1, 11, 81, 101] has gradually saturated, as evidenced by minimal gains compare to PointNeXt [59] over the past few years, such as ScanObjectNN [81] (OA +0.4%, mAcc +0.5%, in Tab. 3) and ShapeNetPart [101] (cls.mIoU +0.4%, Ins.mIoU +0.2%, in Tab. 4). Particularly, the performance of most existing methods on S3DIS [1] has saturated around 73% [78, 88], as shown in Figure 1. The primary reason is that they already adequately describe the local geometric properties of

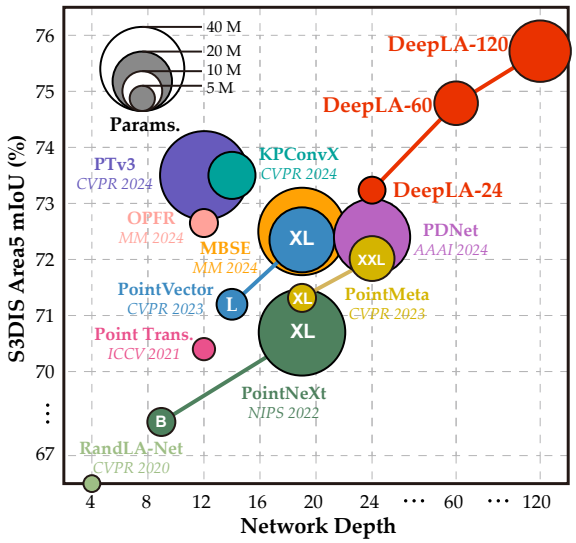


Figure 1. Illustration of segmentation performance, model efficiency and network depth on S3DIS Area5. DeepLA-24 has already achieved SOTA-equivalent performance with minimal parameters. The deeper architecture DeepLA-120 achieves a milestone (75.7%) by exceeding 75% threshold for the first time, and outperforms Point Transformer v3 [88] with fewer parameters.

*corresponding author

point clouds, and more complex designs no longer contribute to capturing additional local patterns [48, 59]. This phenomenon prompts us to rethink the design of LANets.

Interestingly, the development of 3D point cloud analysis is closely related to the evolution of 2D image processing networks [48, 57]. We observe that the design of 2D CNNs shifted towards deeper with the residual connections [24], making deep networks the mainstream backbone for feature extraction with successful performance [2, 44, 65]. Given that LANets for point clouds share similar inductive biases with CNNs, we cannot help but wonder: can we explore an alternative path by leveraging the success of deep CNNs for LANets? Unfortunately, to the best of our knowledge, there has been limited systematic exploration of deepening LANets. Therefore, in this paper, instead of following the tendency to explore sophisticated details, we are in pursuit of very deep LANets for point cloud analysis. For this purpose, we encounter two key challenges: (1) **high computational cost**, and (2) **difficult training optimization**.

Obviously, the prevailing philosophy of sophisticated and redundant local learning has significantly increased computational complexity. To circumvent the prohibitive computations, we propose a residual local feature extraction (ResLFE) block that ensures minimal computational cost. Specifically, we first employ stage-level positional embedding instead of redundant and high latency layer-level. We then introduce an efficient vector feature representation and a robust modernization structure that significantly reduces FLOPs while preserving local information interactions. As shown in Figure 2, the proposed ResLFE block shows significant computational efficiency ($10\times$ fewer FLOPs) and commendable performance improvements (+1.1% mIoU). In particular, this remarkable computational efficiency enables the construction of very deep networks.

Furthermore, very deep networks are difficult to optimize [24, 92], and the complex geometric patterns of point clouds further exacerbate the difficulty of optimization [19, 54, 73]. To address these issues, we introduce a hybrid deep supervision (HDS) strategy to facilitate learning and optimization. We employ two forms of supervision to the outputs of the encoder at each stage. Specifically, we align each output with the ground truth using cross-entropy loss, which promotes smooth gradient propagation. Meanwhile, we also align the outputs with the spatial distribution of the point cloud using mean squared error loss, which benefits in fitting the complex geometric patterns to simplify network optimization. We demonstrate in experiments that the proposed HDS strategy optimizes network training by accelerating model convergence and achieves significant performance improvements (+3.1% mIoU).

Overall, we propose very **Deep** **L**ocal **A**ggregation **Net**works (DeepLA-Net), constructed from the ResLFE block for lightweight design and the HDS strategy for optimization. In particular, we have successfully trained DeepLA-120—a representative deep network comprising up to 120 ResLFE blocks—on a single RTX 3090 with 24GB memory. The DeepLA-Net family outperforms the SOTA methods across various tasks, including S3DIS [1] and ScanNetV2 [11] for semantic segmentation, ScanObjectNN [81] for object classification, and ShapeNetPart [101] for part segmentation. Figure 1 highlights the superior performance and efficiency of our DeepLA-Net family on S3DIS Area5. Moreover, a step-by-step procedure of constructing the DeepLA-Net and the corresponding results are shown in Figure 2. Our key contributions are:

- We propose a residual local feature extraction block from the perspective of efficiency and performance, enabling the construction of deeper LANets.
- We propose a hybrid deep supervision strategy from the perspective of optimization and learning to ensure smooth gradient backpropagation in deeper LANets.
- We propose DeepLA-Net, a series of very deep local aggregation networks, surpassing SOTA in classification, part segmentation, and semantic segmentation.

2. Related Work

2.1. Deep Learning on Point Clouds

Given the disordered and unstructured nature of 3D point clouds, directly applying traditional CNNs to them remains a challenge. Therefore, some methods firstly transform point clouds into intermediate representations such as 2D images [3, 34, 51, 93, 103] or 3D voxels [7, 9, 20, 35, 50, 116] through projection or voxelization, and subsequently employ 2D/3D CNNs to process these structured data. Although promising results have been shown in certain scenarios, the intermediate transformation steps may introduce information loss and computation costs [17, 62, 107]. In contrast, point-based methods [17, 25, 40, 56, 57, 62, 87, 104, 106, 107, 113] are specifically designed to perform feature learning directly on point clouds without using intermediate representations. Moreover, some noteworthy research has leveraged techniques like Recurrent Neural Networks (RNNs) [27, 43, 100], Graph Neural Networks (GNNs) [33, 37, 83, 95, 108], point-voxel hybrid representation [45, 46, 66, 94, 110], and sparse voxel [4, 10, 69, 75, 76], achieving significant performance in various tasks.

2.2. Local Aggregation Networks for Point Clouds

The pioneering PointNet [56] uses point-wise MLPs to learn per-point features, however, it has limitations in capturing local details. To overcome this, PointNet++ [57] improves PointNet by employing shared-weight MLPs in local neighbors, aligning with the inductive biases in CNNs: localization and weight sharing. Along this direction, nu-

merous subsequent studies [39, 49, 80, 85, 87, 112] have employed the above two inductive biases by local feature aggregation operations. We summarize them as local aggregation networks (LANets). Recent LANets [12, 17, 25, 42, 53, 59, 62, 67, 104, 106, 107] have primarily focused on exploring more robust local representations. For example, Point Transformer [114] designs a local transformer layer based on vector self-attention in local neighbors. PointVector [12] improves local extractor by transforming relative features and positions of input into representative vectors. ConDaFormer [16] enhances local geometric modeling by depth-wise convolutions, capturing both long-range contextual information and local priors. However, with the development of LANets, merely development of more complex LANets is insufficient for further capturing the local patterns. In this paper, we demonstrate that without relying on intricate and complex local representations, designing very deep hierarchical architectures for LANets can also achieve significant breakthroughs in performance.

3. Preliminary

In this section, we introduce the preliminary implementation of DeepLA-Net. Most existing LANets have focused on robust local representation to effectively capture the complex local patterns of sparse and discrete point clouds. However, this focus makes them difficult to deepen due to the high computational cost. Therefore, we consider building deep networks based on the simplest PointNet++ [57].

PointNet++ [57], the pioneering LANet for point cloud analysis, can be summarized in two phases: position embedding and feature representation. In the position embedding phase, PointNet++ employs ball query to group points, and then uses relative position differences as position embedding. Subsequently, in the feature representation phase, PointNet++ concatenates position embedding and grouped features to represent local features, and then aggregates them to centroid points by using max pooling. This can be formulated as:

$$\begin{cases} p_i^k = Group(p_i) \\ PE = \mathcal{M}(p_i - p_i^k) \end{cases} \qquad \begin{cases} F_N = \mathcal{M}[f_i^k \oplus PE] \\ F_{out} = max(F_N) \end{cases} \tag{1}$$

where p_i denotes the centroid points, p_i^k denotes the grouped points, $\mathcal{M}$ denotes the multi-layer perceptron, $\oplus$ denotes the concatenation, max denotes the max-pooling, and f_i^k denotes the grouped features.

Implementation. Due to its simplicity, we employ PointNet++ [57] as the baseline and demonstrate the step-by-step "modernization" into DeepLA-Net. Firstly, we follow the PointNeXt [59] configuration, incorporating data augmentation and optimization techniques such as random height and color dropout. The baseline model achieves 63.4% mIoU and 11.3G FLOPs[1]. Subsequently, we directly extend the baseline to 24-block model, following [44] with a block ratio of [1:1:3:1]. This plain-24 model achieves 69.0% mIoU and 92.2G FLOPs. Despite achieving a respectable mIoU score, the huge FLOPs prevent us from further deepening the network. In addition, deep network poses optimization challenges. Therefore, to construct LANets as deep as possible, we propose the ResLFE block which significantly reduces the computation cost while preserving reliable accuracy. Subsequently, we propose the HDS strategy to ensure smooth gradient backpropagation in deep networks and mitigate training optimization challenges.

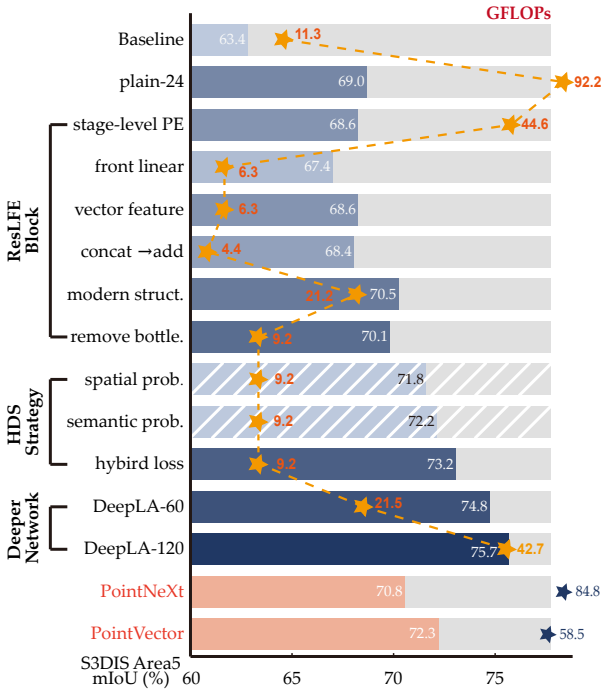


Figure 2. Illustration of the designs and results of our improvements on S3DIS [1] Area5 step-by-step. The foreground bars denote model accuracy, the stars denote model FLOPs, and hatched bars mean the modification is not adopted solely.

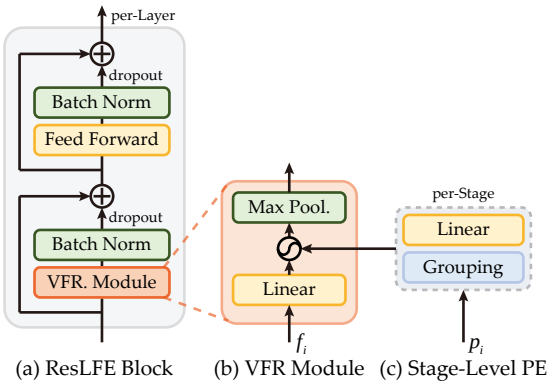
4. Methodology

In this section, we introduce the key components of our DeepLA-Net: residual local feature extraction (ResLFE) block and hybrid deep supervision (HDS) strategy. To intuitively and clearly demonstrate the role of each design, we illustrate each step of improvement in Figure 2, with all models evaluated on S3DIS Area 5 [1].

4.1. Residual Local Feature Extraction Block

To enable the construction of deeper networks, we propose the ResLFE block, as shown in Figure 3, which significantly reduces computational cost. The ResLFE block comprises

[1]FLOPs calculation follows the settings of PointNeXt [59] which evaluated on S3DIS with $16 \times 15,000$ points.



(a) ResLFE Block (b) VFR Module (c) Stage-Level PE

Figure 3. Illustration of the proposed ResLFE block. Right: stage-level position embedding. Middle: Vector Feature Representation (VFR.) module. Left: modernization structure.

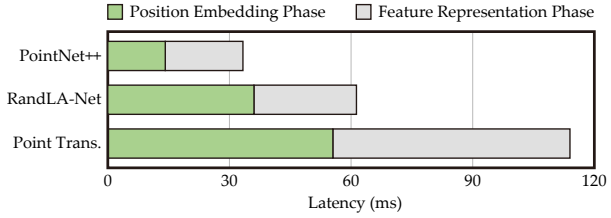


Figure 4. Illustration of the computational latency of position embedding and feature representation phases in different LANets.

three key aspects: (1) stage-level position embedding, (2) efficient vector feature representation module, and (3) powerful modernization structure.

Stage-Level Position Embedding: Existing LANets typically implement position embedding at each learning layer, resulting in high computational costs. As shown in Figure 4, the computational latency of the positional embedding phase constitutes approximately half of the total for each layer, which is unacceptable in deeper networks. As indicated by Eq. 1, we observe that the computation of position embedding relies solely on point coordinates. Considering that the resolution of points remains constant within each stage, layers in the same stage can share an identical position embedding. Therefore, we suggest pre-calculate the position embedding at the beginning of each stage, saving it to cache and reusing it per-layer in that stage to avoid subsequent real-time computations. This approach ensures that the computational costs of position embedding depend solely on the number of stages, rather than the depth of the network, thereby facilitating the efficient application of very deep networks. Our implementation of position embedding is the same as PointNet++. This operation approximately halves the FLOPs (92.2G $\rightarrow$ 44.6G), with only a minimal decrease in performance (69.0% $\rightarrow$ 68.6%).

Vector Feature Representation Module: First, we consider further improvements in computational efficiency during the feature representation phase. We observe that, in most existing LANets, feature abstraction (linear layers in Eq. 1) is typically performed after grouping the features into $f_i^k \in \mathbb{R}^{N\times K\times C}$, which incurs substantial computational costs. To address this issue, we suggest conducting feature abstraction before grouping on $f_i \in \mathbb{R}^{N\times C}$ (front-linear). Although this approach may limit local information interactions, it allows for a reduction in FLOPs by K times theoretically, leading to significant efficiency gains. Practically, this operation reduces the 7$\times$ FLOPs (44.6G $\rightarrow$ 6.3G), with a performance degradation of only 1.2% (68.6% $\rightarrow$ 67.4%).

Moreover, we suggest employing the vector feature $f_i - f_i^k$ instead of the grouped feature f_i^k. On the one hand, this approach achieves cost-effective local interaction, which can enhance relation learning. On the other hand, it aligns with the relative position embedding $p_i - p_i^k$, mitigating the potential semantic gap between position embedding and feature representation. This operation improves performance by 1.2% (67.4% $\rightarrow$ 68.6%) without incurring additional computation costs.

Furthermore, when combining the vector feature with the relative position embedding, we replace concatenation ($\oplus$) with simple addition ($+$), avoiding duplicating channels, which further reduces computational demands. This operation reduces computation (6.3G $\rightarrow$ 4.4G) while only marginally decreasing performance (68.6% $\rightarrow$ 68.4%). Finally, we employ the simple but effective max pooling for aggregation. The vector feature representation (VFR) module can be formulated as follows:

$$\begin{cases} f_i' = \mathcal{M}(f_i) \\ F_i^k = f_i' - {f'}_i^k \\ F_{out} = max(F_i^k + PE) \end{cases} \tag{2}$$

Modernization Structure: Inspired by powerful Transformers [14, 82], we adopt a modernization structure, as shown in Figure 3-(a). Similar to the Transformer block, we employ a dropout path after normalization to mitigate overfitting in deep networks. Notably, unlike the 4$\times$ Inverted Bottleneck Feed Forward Network in Transformer, which increases the channel dimension, our implementation maintains the same dimensions in the two linear layers of the FFN to conserve computational resources. We can observe that the vanilla modern structure improves performance by 2.1% (68.4% $\rightarrow$ 70.5%), but incurs significant additional computations (4.4G $\rightarrow$ 21.2G). By removing the 4$\times$ inverted bottleneck, we reduce the FLOPs by approximately half (21.2G $\rightarrow$ 9.2G), with only a 0.4% decrease in performance (70.5% $\rightarrow$ 70.1%).

At this point, we have construct the vanilla DeepLA-24. Compare to the plain-24 model, the vanilla DeepLA-24 improves performance by 1.1% (69.0% $\rightarrow$ 70.1%) while using only 10% FLOPs (92.2G $\rightarrow$ 9.2G). This small but mighty design enables the construction of deeper networks.

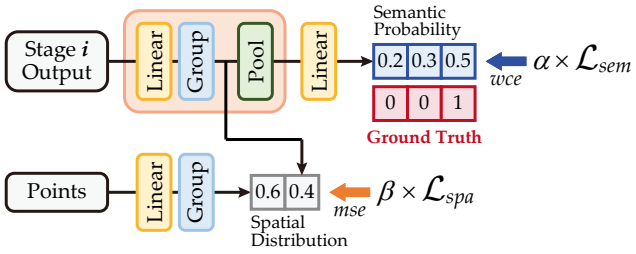


Figure 5. Illustration of the proposed HDS strategy.

4.2. Hybrid Deep Supervision Strategy

In this section, we propose the HDS strategy, as illustrated in Figure 5. By incorporating supervision signals at each encoding stage, we facilitate smoother gradient propagation and simplify network optimization. The HDS strategy comprises two key aspects: (1) semantic probability supervision, and (2) spatial distribution supervision.

Semantic Probability Supervision: Despite employing techniques such as residual connections, normalization, and dropout, optimizing very deep networks remains a challenge. The proposed semantic probability supervision encourages hidden layers to learn discriminative features, which facilitates faster convergence and regularization of the network. Specifically, the outputs of each stage are fed into a VFR module without position embedding, ensuring that the supervised signals closely reflect our training process. Then, they are processed by a plain linear layer to align with the ground truth labels. Finally, we use the widely-used weighted cross-entropy loss to calculate the loss at each stage. The loss for semantic probability supervision ($\mathcal{L}_{sem}$) is defined as the average of all stages, which can be formulated as follows:

$$\mathcal{L}_{sem}(Y,P) = -\frac{1}{N}\sum_{i=0}^{N}\sum_{c=0}^{C} y_i^c \log p_i^c \tag{3}$$

where y_i^c denotes the label vector of the i-stage, p_i^c denotes the predicted vector of the i-stage, C is the number of label categories, and N denotes the number of stages. This operation improves performance by 2.1% (70.1% $\rightarrow$ 72.2%) without additional inference computational costs.

Spatial Distribution Supervision: The sparsity and discreteness of point clouds lead to intricate complex local geometric patterns, which further exacerbate the challenges of optimizing very deep networks. To address this issue, we assume that the feature distribution is potentially consistent with the spatial distribution of the point clouds. In the early epochs of training, feature learning should trend toward fitting these geometric structures. Therefore, we propose the spatial distribution supervision that serves as a manual prior for convergence, explicitly guiding feature learning towards geometric structures during the early epochs of training. Specifically, we represent the spatial distribution by calculating the grouped points with a learnable linear layer. The weights are initialized as an identity matrix and without bias, while the input and output channels are set to 3. Subsequently, we use this spatial distribution to constrain the grouped features obtained during semantic probability supervision. Finally, we adopt the mean squared error (MSE) loss to minimize the discrepancy between the spatial supervision signals and the grouped features. The loss of spatial distribution supervision ($\mathcal{L}_{spa}$) is defined as the average of all stages, which can be formulated as follows:

$$\mathcal{L}_{spa}(Y,P) = -\frac{1}{N}\sum_{i=0}^{N}(p_i - y_i)^2 \tag{4}$$

where y_i denotes the spatial supervision signals of the i-stage, p_i denotes the predicted results from feature relative differences of the i-stage, and N denotes the number of encoder stages. This operation improves performance by 1.7% (70.1% $\rightarrow$ 71.8%).

Hybrid Loss: The loss function of the network integrates L_{sem}, L_{spa}, and the segmentation loss of final prediction (L_{pred}), which is calculated via cross-entropy loss. Furthermore, we implement different training weights with exponential decay for L_{sem} and L_{spa}, where the decay factor is defined as the reciprocal of the current epoch number. This strategy ensures that deep supervision provides significant guidance in the early training phases, while gradually diminishes its influence to avoid hindering the primary learning path as training progresses. The hybrid loss can be formulated as:

$$\mathcal{L}_{\mathcal{H}} = \alpha^n \mathcal{L}_{sem} + \beta^n \mathcal{L}_{spa} + (1 - \alpha^n - \beta^n)\mathcal{L}_{pred} \tag{5}$$

where α and β are the training weights for supervision, n denotes reciprocal of the current epoch number. This hybrid loss improves performance by 3.1% (70.1% $\rightarrow$ 73.2%).

4.3. DeepLA-Net Implementation

We construct a series of DeepLA-Net with different depths: DeepLA-24/60/120. Following successful previous works [14, 44, 114], the block ratio within each stage is set to [1:1:3:1], and the number of channels doubles in the subsequent stage. The configuration of DeepLA-Net is summarized as follows:

$C = [64, 128, 256, 512]$

- DeepLA-24: $B = [4, 4, 12, 4]$
- DeepLA-60: $B = [10, 10, 30, 10]$
- DeepLA-120: $B = [20, 20, 60, 20]$

where C refers to the channels of the output and B denotes the number of ResLFE blocks in each stage. Note that, in this paper, the term "depth" specifically refers to the number of blocks rather than linear layers. The network architecture details can be found in **Appendix**.

Table 1. Quantitative comparisons of semantic segmentation with the SOTA methods on S3DIS in terms of mIoU.

Year	Method	6-flod (%)	Area5 (%)
NIPS 2017	PointNet++ [57]	83.0	53.5
NIPS 2018	PointCNN [39]	65.4	57.3
CVPR 2019	KPConv [77]	70.6	67.1
CVPR 2020	RandLA-Net [25]	70.0	62.5
CVPR 2021	Point Trans. [114]	73.5	70.4
NIPS 2022	PointNeXt [59]	74.9	70.8
CVPR 2023	AF-GCN [111]	77.7	72.3
CVPR 2023	PointVector [12]	<u>78.4</u>	72.3
CVPR 2023	PointMeta [42]	77.0	72.0
AAAI 2024	PDNet [102]	78.3	72.3
CVPR 2024	KPConvX [78]	-	<u>73.5</u>
CVPR 2024	OneFormer3D [30]	75.0	72.4
CVPR 2024	Point Trans. v3 [88]	77.7	73.4
ACM MM 2024	OPFR [28]	76.9	72.6
ACM MM 2024	LDCNet [47]	75.4	71.8
ACM MM 2024	MBSE [86]	77.8	72.4
Ours 2024	**DeepLA-24**	77.9	73.2
Ours 2024	**DeepLA-60**	79.0	74.8
Ours 2024	**DeepLA-120**	**79.8**	**75.7**

Table 2. Quantitative comparisons of semantic segmentation with the SOTA methods on ScanNet v2 in terms of mIoU.

Year	Method	val (%)	test (%)
NIPS 2017	PointNet++ [57]	53.5	55.7
NIPS 2018	PointCNN [39]	-	45.8
CVPR 2019	KPConv [77]	69.2	68.6
CVPR 2022	Stra. Trans. [32]	74.3	73.7
NIPS 2022	PointNeXt [59]	71.5	71.2
CVPR 2023	LRPNet [38]	75.0	74.2
ICCV 2023	Retro-FPN [90]	74.0	74.4
NIPS 2023	ConDaFormer [16]	75.1	75.5
AAAI 2024	HPENet [117]	74.0	-
ICLR 2024	MVNet [96]	75.2	-
CVPR 2024	KPConvX [78]	76.3	-
CVPR 2024	OneFormer3D [30]	<u>76.6</u>	-
CVPR 2024	OA-CNN [55]	76.1	<u>75.6</u>
CVPR 2024	MirageRoom [71]	74.9	-
ACM MM 2024	LDCNet [47]	73.3	-
Ours 2024	**DeepLA-24**	74.1	-
Ours 2024	**DeepLA-60**	75.9	-
Ours 2024	**DeepLA-120**	**77.6**	**77.2**

5. Experiments

5.1. Experiment Setup

To comprehensively evaluate the effectiveness of the proposed DeepLA-Net, we conduct experiments on S3DIS [1] and ScanNet v2 [11] for semantic segmentation, ScanObjectNN [81] for object classification, and ShapeNetPart [101] for object part segmentation. We use the following quantitative metrics for evaluation: mean Intersection over Union (mIoU), Overall Accuracy (OA), and mean Accuracy (mAcc). In result tables, **Bold** indicates the best result, <u>underline</u> indicates the best result excluding ours.

In the position embedding phase, we employ the KNN for grouping, with the K specified as 24. For hybrid deep supervision, α and β are set to 0.3 and 0.005, respectively. All experiments are performed on a Nvidia RTX 3090 GPU with 24GB memory. To ensure a fair comparison, the reported performance for both ours and the compared methods excludes the use of voting, pre-trained model, and test-time augmentation strategies. The implementation details and dataset description can be found in **Appendix**.

5.2. Experiment Results

Semantic Segmentation. We show the results of our DeepLA-Net, including DeepLA-24/60/120, compared with previous state-of-the-art methods on S3DIS and ScanNet v2. In Table 1, for S3DIS, our DeepLA-60 has already surpassed previous methods. Particularly, our DeepLA-120 achieves a more significant breakthrough with an mIoU of 79.8% in 6-fold (+1.4%) and 75.7% (+2.2%) in Area5, surpassing the 75% threshold for the first time. In Table 2, for ScanNet v2, our DeepLA-24/60 achieve competitive performance, while DeepLA-120 significantly outperforms previous methods, achieving 77.6% (+1.0%) on the validation set and 77.2% (+1.6%) on the test sets. Note that, due to the submission policy of the ScanNet v2 online test set, we only report the best model on the validation set, i.e., DeepLA-120. Additionally, results on the ScanNet test set have been reported less in recent years because fully supervised methods have not achieved substantial breakthroughs in this dataset. The result of the efficient DeepLA-24 demonstrates the rationality and effectiveness of our design. Furthermore, reasonably deepening the network can significantly improve performance. More detailed per-class results and visualizations can be found in **Appendix**.

Table 3. Quantitative comparisons of classification with the SOTA methods on ScanObjectNN (PB T50 RS).

Year	Method	OA (%)	mAcc (%)
CVPR 2017	PointNet [56]	68.2	63.4
NIPS 2018	PointCNN [39]	78.5	75.1
TOG 2019	DGCNN [85]	78.1	73.6
TMM 2021	GBNet [61]	80.5	77.8
TIP 2021	PRANet [8]	82.1	79.1
ICLR 2022	PointMLP [48]	85.7	84.4
NIPS 2022	PointNeXt [59]	88.1	86.4
PAMI 2023	GSLCN [41]	85.8	84.1
ICLR 2023	ACT [13]	88.2	-
CVPR 2023	NLGAT [60]	88.4	-
CVPR 2023	PointVector [12]	88.2	86.7
CVPR 2023	PointMeta [42]	88.1	<u>86.9</u>
AAAI 2024	PDNet [102]	<u>88.5</u>	86.8
AAAI 2024	Interpretable3D [18]	88.0	86.5
ICLR 2024	MaskFeat3D [97]	88.4	-
ACM MM 2024	OPFR [28]	88.1	86.3
Ours 2024	**DeepLA-24**	**90.6**	**89.5**

Table 4. Quantitative comparisons of part segmentation with the SOTA methods on ShapeNetPart.

Year	Method	Cls. mIoU (%)	Ins. mIoU (%)
CVPR 2017	PointNet [56]	80.4	83.7
NIPS 2018	PointCNN [39]	84.6	86.1
TOG 2019	DGCNN [85]	82.3	85.1
CVPR 2020	PointASNL [98]	-	86.1
CVPR 2021	Point Trans. [114]	83.7	86.6
ICLR 2022	PointMLP [48]	84.6	86.1
NIPS 2022	PointNeXt [59]	85.2	87.0
ICLR 2023	ACT [13]	84.7	86.1
CVPR 2023	PointVector [12]	85.1	86.9
CVPR 2023	PointMeta [42]	85.1	87.1
CVPR 2023	AF-GCN [111]	85.3	87.0
NIPS 2023	ConDaFormer [16]	84.6	86.8
AAAI 2024	HPENet [117]	85.5	87.1
AAAI 2024	PDNet [102]	85.4	87.2
AAAI 2024	Interpretable3D [18]	85.6	87.2
ICLR 2024	MVNet [96]	85.2	86.8
ICLR 2024	MaskFeat3D [97]	85.5	87.0
Ours 2024	**DeepLA-24**	**86.5**	**88.0**

Object Classification. As shown in Table 3, DeepLA-24 outperforms all baselines with OA of 90.6% and mAcc of 89.5% (+2.1% OA, +2.6% mAcc). DeepLA-24 is the first method to surpass 90% on OA, with the performance of others stagnating around 88% for several years. This result indicates that merely developing more complex local representation methods is not sufficient for further advancement in shape recognition. In contrast, deeper networks can make significant contributions to capturing local patterns.

Part Segmentation. The results are reported in Table 4, where we evaluate the performance with the mean class IoU (Cls. mIoU) and mean instance IoU (Ins. mIoU). DeepLA-24 outperforms all baselines with the Cls. mIoU of 86.5% (+1.0%) and the Ins. mIoU of 88.0% (+0.8%) *without voting*. It is noteworthy that we do not experiment with the deeper variants of DeepLA-Net on object classification and part segmentation datasets due to the limited dataset scale.

5.3. Ablation Studies

In this section, we conduct ablation studies on the proposed DeepLA-Net. All the ablation models are evaluated on S3DIS Area5 [1]. If not specified, the DeepLA-24 is employed as the default model.

Hyperparameter settings. We analyze the impact of hyperparamter settings including the bottleneck in ResLFE block, the training weights in HDS strategy, the initial feature channel, and the grouping range K. Table 5 explores different bottleneck settings in ResLFE block. The results illustrate that inverted bottlenecks larger than $1\times$ marginally improve performance but significantly increase computation costs. Meanwhile, bottlenecks less than $1\times$ slightly reduce computation while causing a more pronounced degradation in performance. Therefore, to balance efficiency and performance, we employ the $1\times$ bottleneck. Table 6 shows that the setting of training weights in HDS strategy is generally insensitive, as long as they are neither too large to hinder fitting nor too small to be ineffective. Notably, the training weight for spatial distribution is much smaller than that for semantic probability. This is mainly because spatial distribution supervision encourages the network to focus more on local details rather than abstract semantics, which are more crucial for feature classification. Figure 6 illustrates the impact of different initial feature channel and grouping range K. The results demonstrate that small channel and grouping range lead to limited performance, due to inadequacy in capturing local patterns. Meanwhile, excessively large channel and grouping range tend to overfit the training examples, resulting in poor generalization and a decline in performance.

Table 5. Ablation results of the bottleneck in ResLFE block in DeepLA-24 on S3DIS Area5.

Model	mIoU (%)	△ (%)	FLOPs (G)	△ (G)
$1\times$ **bottleneck**	**73.2**	-	9.3	-
$0.5\times$ bottleneck	72.5	-0.7	7.4	-1.9
$2\times$ bottleneck	73.4	$+0.2$	13.4	$+4.1$
$4\times$ bottleneck	73.5	$+0.3$	21.3	$+13.0$

Table 6. Ablation results of training weights in HDS strategy in DeepLA-24 on S3DIS Area5.

α \ β	0	3e-3	5e-3	7e-3	1e-2
0	70.1	71.0	71.8	71.3	70.6
0.2	71.8	72.3	72.5	72.9	72.0
0.3	72.2	72.5	**73.2**	72.8	72.4
0.4	72.0	72.6	72.4	72.4	72.1
0.5	71.5	72.0	72.1	72.2	71.7

high

low

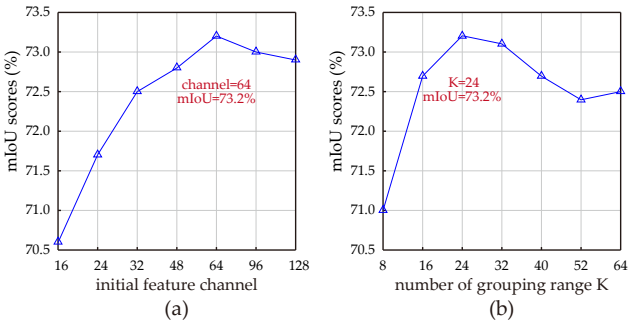


Figure 6. Ablation results of hyperparameters in DeepLA-24 on S3DIS Area5. (a) initial feature dimension. (b) grouping range K.

Ablation of Deep Supervision for Training Process. We demonstrate the performance and segmentation loss values (L_{pred} in Eq. 5) of the network with different deep supervision strategies across training epochs, as shown in Fig-

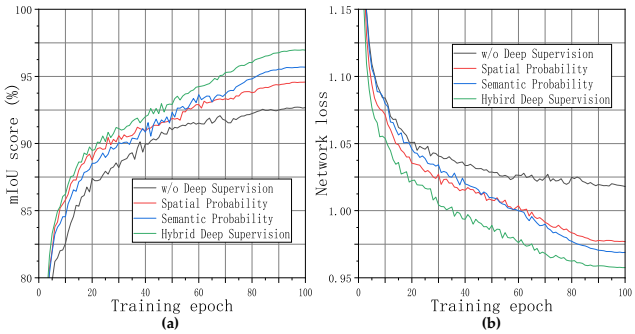


Figure 7. Training performance and loss value across training epochs in DeepLA-24 with different deep supervision strategies. (a) mIoU score, and (b) loss value.

ure 7. It is evident that the network without deep supervision exhibits a slow decline in loss, with almost no further decrease after 60 epochs. In contrast, networks using deep supervision consistently achieve higher performance and a rapid convergence in loss. Notably, the network employing the HDS strategy achieves the most reasonable and effective convergence. This result demonstrates that our HDS strategy significantly optimizes network training, facilitating a robust and substantial decline in loss.

Ablation of Network Depth for Feature Learning. We visualize the feature similarity matrix of DeepLA-Net with different depths in Figure 8. The matrix is obtained by calculating the cosine similarities of the point features in the final layer, which are then sorted according to the predicted categories. The results indicate that the feature learning of DeepLA-6 is not robust, resulting in numerous recognition errors during segmentation. In contrast, DeepLA-24 demonstrates stronger feature learning capabilities, though the feature differences between classes are not sufficiently distinct, potentially causing blurred boundaries. DeepLA-120 demonstrates the highest confidence in class distinction, showcasing the most powerful feature learning abilities among the three networks. This result demonstrates that deeper networks possess more powerful learning capabilities, enhancing segmentation performance.

Analysis on Complexity and Latency. We report the performance, model complexity, and latency using the same settings of DeepLA-Net family and previous SOTA methods [12, 59, 78, 102] in Table 7. From the results, the baseline exhibits the fastest inference speed but the lowest performance. Simply deepening the baseline (Plain-24) enhances performance but significantly increases FLOPs (9× higher) and decreases inference speed (4.7× slower). In contrast, DeepLA-24, equipped with the ResLFE block achieves performance improvements while substantially reducing FLOPs (10× fewer) and increasing inference speed (3.75× faster). The reduction in FLOPs enables the training of deeper networks. In particular, our DeepLA-120 achieves state-of-the-art performance with an inference

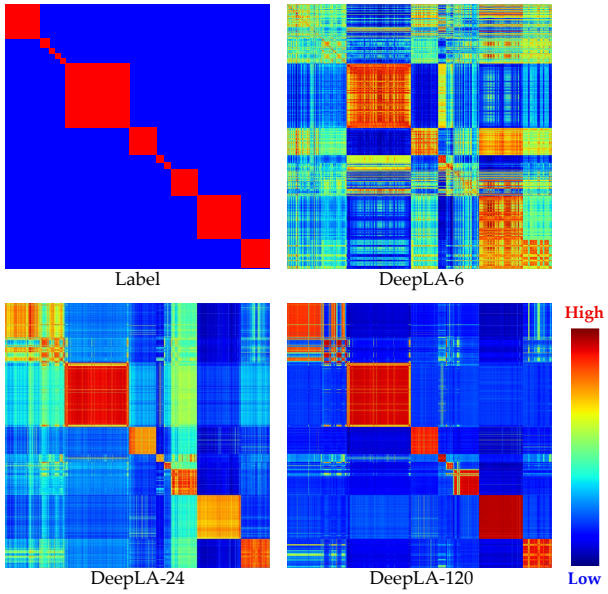


Figure 8. Visual comparison of interpreting feature similarity matrix with different network depth of the DeepLA-6 (6-block network follows [1:1:3:1] ratio), DeepLA-24, and DeepLA-120.

Table 7. Quantitative comparisons of performance, model complexity, and latency on S3DIS Area5.

Method	mIoU (%)	Params. (M)	FLOPs (G)	Thr. Put (ins./sec.)
PointNeXt-XL [59]	70.8	46.1	84.8	43
PointVector-XL [12]	72.3	24.1	58.5	40
KPConvX-L [78]	73.5	13.5	-	47
PDNet-XXL [102]	72.3	35.6	12.0	-
Baseline	63.4	3.1	11.3	207
Plain-24	69.0	9.7	92.2	44
DeepLA-24	73.2	**6.4**	**9.2**	**165**
DeepLA-60	74.8	15.8	21.5	78
DeepLA-120	**75.7**	30.3	42.7	42

speed comparable to Plain-24 and the previous SOTA works [12, 59, 78]. More ablation analysis and discussion details can be found in **Appendix**.

6. Conclusions

In this study, we demonstrate the efficiency and effectiveness of very deep local aggregation networks for point cloud analysis, addressing two primary challenges: computational cost and training optimization. In contrast to most current approaches that rely on expensive and redundant local representation, we propose a lightweight ResLFE block to significantly reduce the computational cost (10× fewer FLOPs), and make it possible to construct very deep LANets. Furthermore, the HDS strategy is also introduced to ensure smooth gradient backpropagation and mitigate optimization challenges in deep networks. The DeepLA-Net achieves SOTA performance with high efficiency across segmentation and classification tasks on four benchmarks: S3DIS, ScanNet v2, ScanObjectNN, and ShapeNetPart.

References

[1] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3D semantic parsing of large-scale indoor spaces. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2016. 1, 2, 3, 6, 7, 14

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In European conference on computer vision (ECCV), 2018. 2

[3] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3D object detection network for autonomous driving. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017. 2

[4] Yukang Chen, Yanwei Li, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Focal sparse convolutional networks for 3d object detection. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 2

[5] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Largekernel3d: Scaling up kernels in 3d sparse cnns. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023. 19

[6] Yu Chen, Chunhua Shen, Xiu-Shen Wei, Lingqiao Liu, and Jian Yang. Adversarial posenet: A structure-aware convolutional network for human pose estimation. In IEEE/CVF International Conference on Computer Vision (ICCV), 2017. 18

[7] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu. (af)2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 2

[8] Silin Cheng, Xiwu Chen, Xinwei He, Zhe Liu, and Xiang Bai. Pra-net: Point relation-aware network for 3d point cloud analysis. IEEE Transactions on Image Processing, 30:4436–4448, 2021. 6

[9] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 2

[10] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 2

[11] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017. 1, 2, 6, 14, 17

[12] Xin Deng, WenYu Zhang, Qing Ding, and XinMing Zhang. Pointvector: A vector representation in point cloud analysis. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023. 1, 3, 6, 7, 8, 17, 18, 19

[13] Runpei Dong, Zekun Qi, Linfeng Zhang, Junbo Zhang, Jianjian Sun, Zheng Ge, Li Yi, and Kaisheng Ma. Autoencoders as cross-modal teachers: Can pretrained 2d image transformers help 3d representation learning? In International Conference on Learning Representations (ICLR), 2023. 6, 7

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, and Sylvaing Gelly. An image is worth 16x16 words: Transformers for image recognition at scale. In International Conference on Learning Representations (ICLR), 2020. 4, 5

[15] Qi Dou, Hao Chen, Yueming Jin, Lequan Yu, Jing Qin, and Pheng-Ann Heng. 3D deeply supervised network for automatic liver segmentation from ct volumes. In International Conference on Medical Image Computing and Computer-Assisted Intervention, 2016. 18

[16] Lunhao Duan, Shanshan Zhao, Nan Xue, Mingming Gong, Xia Gui-Song, and Dacheng Tao. Condaformer: Disassembled transformer with local structure enhancement for 3d point cloud understanding. In Neural Information Processing Systems (NeurIPS), 2023. 3, 6, 7, 19

[17] Siqi Fan, Qiulei Dong, Fenghua Zhu, Yisheng Lv, Peijun Ye, and Fei-Yue Wang. Scf-net: Learning spatial contextual features for large-scale point cloud segmentation. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 2, 3

[18] Tuo Feng, Ruijie Quan, Xiaohan Wang, Wenguan Wang, and Yi Yang. Interpretable3d: An ad-hoc interpretable classifier for 3d point clouds. In AAAI Conference on Artificial Intelligence, 2024. 6, 7

[19] Edgar Galván and Peter Mooney. Neuroevolution in deep neural networks: Current trends and future challenges. IEEE Transactions on Artificial Intelligence, 2(6):476–493, 2021. 2

[20] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3D semantic segmentation with submanifold sparse convolutional networks. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018. 2

[21] Steven Guan, Amir A Khan, Siddhartha Sikdar, and Parag V Chitnis. Fully dense unet for 2-D sparse photoacoustic tomography artifact removal. IEEE Journal of Biomedical and Health Informatics, 24(2):568–576, 2019. 17

[22] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3D point clouds: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(12):4338–4364, 2020. 1

[23] Dongyoon Han, Sangdoo Yun, Byeongho Heo, and YoungJoon Yoo. Rethinking channel dimensions for efficient model design. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 17

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2016. 2, 17

[25] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 1, 2, 3, 6,

19

[26] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017. 17

[27] Qiangui Huang, Weiyue Wang, and Ulrich Neumann. Recurrent slice networks for 3D segmentation of point clouds. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018. 2

[28] Wang Jiangyi, Zhongyao Cheng, Na Zhao, Jun Cheng, and Xulei Yang. On-the-fly point feature representation for point clouds analysis. In ACM International Conference on Multimedia, 2024. 6

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In International Conference on Learning Representations, 2015. 14

[30] Maxim Kolodiazhnyi, Anna Vorontsova, Anton Konushin, and Danila Rukhovich. Oneformer3d: One transformer for unified point cloud segmentation. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024. 6

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Neural Information Processing Systems (NeurIPS), 2012. 17

[32] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 6, 19

[33] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018. 2

[34] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 2

[35] Truc Le and Ye Duan. Pointgrid: A deep network for 3D shape understanding. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018. 2

[36] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In Artificial Intelligence and Statistics, 2015. 18

[37] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In IEEE/CVF International Conference on Computer Vision (ICCV), 2019. 2

[38] Xiang-Li Li, Meng-Hao Guo, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Long range pooling for 3d large-scale scene understanding. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023. 6, 19

[39] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. Neural Information Processing Systems (NeurIPS), 31, 2018. 3, 6, 7, 19

[40] Ying Li, Lingfei Ma, Zilong Zhong, Dongpu Cao, and Jonathan Li. TGNet: Geometric Graph CNN on 3-D Point Cloud Segmentation. In IEEE Transactions on Geoscience and Remote Sensing, 2020. 2

[41] Jiye Liang, Zijin Du, Jianqing Liang, Kaixuan Yao, and Feilong Cao. Long and short-range dependency graph structure learning framework on point cloud. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(12):14975–14989, 2023. 6, 19

[42] Haojia Lin, Xiawu Zheng, Lijiang Li, Fei Chao, Shanshan Wang, Yan Wang, Yonghong Tian, and Rongrong Ji. Meta architecture for point cloud analysis. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023. 1, 3, 6, 7, 18, 19

[43] Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Point2sequence: Learning the shape representation of 3d point clouds with an attention-based sequence to sequence network. In AAAI conference on artificial intelligence, 2019. 2

[44] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 2, 3, 5

[45] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3D deep learning. In Neural Information Processing Systems (NeurIPS), 2019. 2

[46] Zhijian Liu, Haotian Tang, Shengyu Zhao, Kevin Shao, and Song Han. Pvnas: 3d neural architecture search with point-voxel convolution. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(11):8552–8568, 2022. 2

[47] Shoutong Luo, Zhengxing Sun, Yi Wang, Yunhan Sun, and Chendi Zhu. Ldcnet: Long-distance context modeling for large-scale 3d point cloud scene semantic segmentation. In ACM International Conference on Multimedia, 2024. 6

[48] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. In International Conference on Learning Representations (ICLR), 2022. 2, 6, 7, 15, 18

[49] Yanni Ma, Yulan Guo, Hao Liu, Yinjie Lei, and Gongjian Wen. Global context reasoning for semantic segmentation of 3D point clouds. In IEEE/CVF Winter Conference on Applications of Computer Vision, 2020. 3

[50] Daniel Maturana and Sebastian Scherer. Voxnet: A 3D convolutional neural network for real-time object recognition. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015. 2

[51] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet ++: Fast and accurate lidar semantic segmentation. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019. 2

[52] Gregory Naitzat, Andrey Zhitnikov, and Lek-Heng Lim. Topology of deep neural networks. The Journal of Machine Learning Research, 21(1):7503–7542, 2020. 17

[53] Dong Nie, Rui Lan, Ling Wang, and Xiaofeng Ren. Pyramid architecture for multi-scale processing in point cloud segmentation. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 3

[54] Amanda Olmin and Fredrik Lindsten. Robustness and re-

liability when training with noisy labels. In International Conference on Artificial Intelligence and Statistics, 2022. 2

[55] Bohao Peng, Xiaoyang Wu, Li Jiang, Yukang Chen, Hengshuang Zhao, Zhuotao Tian, and Jiaya Jia. Oa-cnns: Omni-adaptive sparse cnns for 3d semantic segmentation. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024. 6

[56] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017. 1, 2, 6, 7, 19

[57] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Neural Information Processing Systems (NeurIPS), 2018. 1, 2, 3, 6, 19

[58] Guocheng Qian, Hasan Hammoud, Guohao Li, Ali Thabet, and Bernard Ghanem. Assanet: An anisotropical separable set abstraction for efficient point cloud representation learning. In Neural Information Processing Systems (NeurIPS), 2021. 18

[59] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Abed Al Kader Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. Neural Information Processing Systems (NeurIPS), 35:23192–23204, 2022. 1, 2, 3, 6, 7, 8, 14, 15, 18, 19

[60] Shengwei Qin, Zhong Li, and Ligang Liu. Robust 3d shape classification via non-local graph attention network. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023. 6

[61] Shi Qiu, Saeed Anwar, and Nick Barnes. Geometric back-projection network for point cloud classification. IEEE Transactions on Multimedia, 24:1943–1955, 2021. 6

[62] Shi Qiu, Saeed Anwar, and Nick Barnes. Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 2, 3, 19

[63] Damien Robert, Hugo Raguet, and Loic Landrieu. Efficient 3d semantic segmentation with superpoint transformer. In IEEE/CVF International Conference on Computer Vision (ICCV), 2023. 19

[64] Damien Robert, Bruno Vallet, and Loic Landrieu. Learning multi-view aggregation in the wild for large-scale 3d semantic segmentation. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 19

[65] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018. 2

[66] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 2

[67] Hui Shuai, Xiang Xu, and Qingshan Liu. Backward attentive fusing network with local aggregation classifier for 3D point cloud semantic segmentation. IEEE Transactions on Image Processing, 30:4973–4984, 2021. 3

[68] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In IAPR Asian Conference on Pattern Recognition, 2015. 17

[69] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice networks for point cloud processing. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018. 2

[70] Yanfei Su, Weiquan Liu, Zhimin Yuan, Ming Cheng, Zhihong Zhang, Xuelun Shen, and Cheng Wang. Dla-net: Learning dual local attention features for semantic segmentation of large-scale building facade point clouds. Pattern Recognition, 123:108372, 2022. 1

[71] Haowen Sun, Yueqi Duan, Juncheng Yan, Yifan Liu, and Jiwen Lu. Mirageroom: 3d scene segmentation with 2d pre-trained models by mirage projection. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024. 6

[72] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 18

[73] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In International Conference on Machine Learning (ICML), 2013. 2

[74] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2015. 18

[75] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In European conference on computer vision (ECCV), 2020. 2

[76] Haotian Tang, Shang Yang, Zhijian Liu, Ke Hong, Zhongming Yu, Xiuyu Li, Guohao Dai, Yu Wang, and Song Han. Torchsparse++: Efficient point cloud engine. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023. 2

[77] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas Guibas. Kpconv: Flexible and deformable convolution for point clouds. In IEEE/CVF International Conference on Computer Vision (ICCV), 2019. 6, 19

[78] Hugues Thomas, Yao-Hung Hubert Tsai, Timothy D Barfoot, and Jian Zhang. Kpconvx: Modernizing kernel point convolution with kernel attention. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024. 1, 6, 8

[79] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In IEEE/CVF International Conference on Computer Vision (ICCV), 2015. 18

[80] Giang Truong, Syed Zulqarnain Gilani, Syed Mo-

hammed Shamsul Islam, and David Suter. Fast point cloud registration using semantic segmentation. In Digital Image Computing: Techniques and Applications, 2019. 3

[81] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In IEEE/CVF International Conference on Computer Vision (ICCV), 2019. 1, 2, 6, 15

[82] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Neural Information Processing Systems (NeurIPS), 2017. 4

[83] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 2

[84] Liwei Wang, Chen-Yu Lee, Zhuowen Tu, and Svetlana Lazebnik. Training deeper convolutional networks with deep supervision. arXiv preprint arXiv:1505.02496, 2015. 18

[85] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. ACM Transactions On Graphics, 38(5):1–12, 2019. 3, 6, 7

[86] Ziming Wang, Boxiang Zhang, Ming Ma, Yue Wang, Taoli Du, and Wenhui Li. Multi-fineness boundaries and the shifted ensemble-aware encoding for point cloud semantic segmentation. In ACM International Conference on Multimedia, 2024. 6

[87] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 2, 3

[88] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 4840–4851, 2024. 1, 6, 17

[89] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. In Neural Information Processing Systems (NeurIPS), 2022. 19

[90] Peng Xiang, Xin Wen, Yu-Shen Liu, Hui Zhang, Yi Fang, and Zhizhong Han. Retro-fpn: Retrospective feature pyramid network for point cloud semantic segmentation. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023. 6, 19

[91] Xiao Xiao, Shen Lian, Zhiming Luo, and Shaozi Li. Weighted res-unet for high-quality retina vessel segmentation. In International Conference on Information Technology in Medicine and Education, 2018. 17

[92] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017. 2, 17

[93] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In European Conference on Computer Vision (ECCV), 2020. 2

[94] Jianyun Xu, Ruixiang Zhang, Jian Dou, Yushi Zhu, Jie Sun, and Shiliang Pu. Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In IEEE/CVF International Conference on Computer Vision (ICCV), 2021. 2

[95] Qiangeng Xu, Xudong Sun, Cho-Ying Wu, Panqu Wang, and Ulrich Neumann. Grid-gcn for fast and scalable point cloud learning. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 2

[96] Siming Yan, Chen Song, Youkang Kong, and Qixing Huang. Multi-view representation is what you need for point-cloud pre-training. In International Conference on Learning Representations (ICLR), 2024. 6, 7

[97] Siming Yan, Yuqi Yang, Yuxiao Guo, Hao Pan, Peng shuai Wang, Xin Tong, Yang Liu, and Qixing Huang. 3d feature prediction for masked-autoencoder-based point cloud pretraining. In International Conference on Learning Representations (ICLR), 2024. 6, 7

[98] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 7, 19

[99] Chaolong Yang, Yuyao Yan, Weiguang Zhao, Jianan Ye, Xi Yang, Amir Hussain, Bin Dong, and Kaizhu Huang. Towards deeper and better multi-view feature fusion for 3d semantic segmentation. In International Conference on Neural Information Processing, 2023. 19

[100] Xiaoqing Ye, Jiamao Li, Hexiao Huang, Liang Du, and Xiaolin Zhang. 3D recurrent neural networks with context fusion for point cloud semantic segmentation. In European Conference on Computer Vision (ECCV), 2018. 2

[101] Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. ACM Transactions on Graphics, 35(6):1–12, 2016. 1, 2, 6, 15

[102] Xingyilang Yin, Xi Yang, Liangchen Liu, Nannan Wang, and Xinbo Gao. Point deformable network with enhanced normal embedding for point cloud analysis. In AAAI Conference on Artificial Intelligence, 2024. 6, 7, 8, 18

[103] Tan Yu, Jingjing Meng, and Junsong Yuan. Multi-view harmonized bilinear network for 3D object recognition. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018. 2

[104] Ziyin Zeng, Qingyong Hu, Zhong Xie, Bijun Li, Jian Zhou, and Yongyang Xu. Small but mighty: Enhancing 3d point clouds semantic segmentation with u-next framework. International Journal of Applied Earth Observation and Geoinformation, 2025. 1, 2, 3, 18, 19

[105] Ziyin Zeng, Huan Qiu, Jian Zhou, Zhen Dong, Jinsheng Xiao, and Bijun Li. Pointnat: Large scale point cloud semantic segmentation via neighbor aggregation with transformer. IEEE Transactions on Geoscience and Remote Sensing, 62:1–18, 2024. 1

[106] Ziyin Zeng, Yongyang Xu, Zhong Xie, Wei Tang, Jie

Wan, and Weichao Wu. Leard-net: Semantic segmentation for large-scale point cloud scene. International Journal of Applied Earth Observation and Geoinformation, 112:102953, 2022. 1, 2, 3, 19

[107] Ziyin Zeng, Yongyang Xu, Zhong Xie, Wei Tang, Jie Wan, and Weichao Wu. Large-scale point cloud semantic segmentation via local perception and global descriptor vector. Expert Systems with Applications, 2024. 1, 2, 3, 19

[108] Ziyin Zeng, Yongyang Xu, Zhong Xie, Jie Wan, Weichao Wu, and Wenxia Dai. Rg-gcn: A random graph based on graph convolution network for point cloud semantic segmentation. Remote Sensing, 14(16):4055, 2022. 2

[109] Chao Zhang, Zhiguo Cao, Xin Xiong, Ke Xian, and Xinyuan Qi. Salient object detection via deep hierarchical context aggregation and multi-layer supervision. In IEEE International Conference on Image Processing, 2019. 18

[110] Cheng Zhang, Haocheng Wan, Xinyi Shen, and Zizhao Wu. Pvt: Point-voxel transformer for point cloud learning. International Journal of Intelligent Systems, 37(12):11985–12008, 2022. 2

[111] Nan Zhang, Zhiyi Pan, Thomas H Li, Wei Gao, and Ge Li. Improving graph representation for point cloud segmentation via attentive filtering. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023. 6, 7

[112] Zhiyuan Zhang, Binh-Son Hua, and Sai-Kit Yeung. Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. In IEEE/CVF International Conference on Computer Vision (ICCV), 2019. 3

[113] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. Pointweb: Enhancing local neighborhood features for point cloud processing. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 2

[114] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H.S. Torr, and Vladlen Koltun. Point transformer. In IEEE/CVF International Conference on Computer Vision (ICCV), 2021. 3, 5, 6, 7, 19

[115] Qikui Zhu, Bo Du, Baris Turkbey, Peter L Choyke, and Pingkun Yan. Deeply-supervised cnn for prostate segmentation. In International Joint Conference on Neural Networks, 2017. 18

[116] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 2

[117] Yanmei Zou, Hongshan Yu, Zhengeng Yang, Zechuan Li, and Naveed Akhtar. Improved mlp point cloud processing with high-dimensional positional encoding. In AAAI Conference on Artificial Intelligence, 2024. 6, 7

Supplementary Materials for
DeepLA-Net: Very Deep Local Aggregation Networks for Point Cloud Analysis

Overview

This supplementary material is organized as follows:

- Section A provides the details of the network architecture.
- Section B presents the experimental settings including: evaluation metrics, implementation details, and dataset description.
- Section C provides additional discussion including: ResLFE block ratio, visual comparison of feature learning, and exploring to deeper networks.
- Section D shows detailed and additional experimentation results for semantic segmentation.
- Section E introduces more related works including: deep neural network architecture and deep super vision.
- Section F shows outlook for the future work.

A. Details of the Network Architecture

We provide detailed network architectures for segmentation and classification. As illustrated in Figure 9, the encoder comprises four encoding stages, and each encoding stage consists of a down-sampling operation and multiple ResLFE blocks, with the output being supervised by HDS strategy. In the segmentation branch, DeepLA-Net follows an encoder-decoder architecture. Each decoding stage comprises an up-sampling operation and a multi-layer perceptron. In the classification branch, global average pooling is applied to the output of the encoder to obtain the global representation. Finally, fully-connected layers with a softmax are used to predict the classification scores, where the segmentation/classification results are dictated by the label with the highest score.

B. Experimental Settings

B.1. Evaluation Metrics

To quantitatively analyze the performance of the proposed architecture, overall accuracy (OA), mean Accuracy (mAcc), per-class intersection over union (IoUs), mean IoU (mIoU), are used as evaluation metrics as follows:

$$\mathrm{OA} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{TP_i}}{\mathrm{N}} \tag{6}$$

$$\mathrm{mAcc} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{Acc_i}}{\mathrm{n}} \tag{7}$$

$$\mathrm{IoU_i} = \frac{\mathrm{TP_i}}{\mathrm{TP_i} + \mathrm{FP_i} + \mathrm{FN_i}} \tag{8}$$

$$\mathrm{mIoU} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{IoU_i}}{\mathrm{n}} \tag{9}$$

where TP denotes the number of true positive samples, FP denotes the number of false positive samples, FN denotes the number of false negative samples, i denotes the i_{th} semantic class, n denotes the number of total semantic classes and N denotes the number of total points.

B.2. Implementation Details

During the training process, we use the hybrid deep supervision strategy with label smoothing to optimize our models . We adopt the AdamW optimizer [29] with an initial learning rate of 0.004, and a scheduler with weight decay of 10^{-4} using cosine learning rate decay. For data augmentation, we use random scaling, feature dropping, and color auto contrasting whenever applicable, following [59]. For semantic segmentation, we input a fixed number of 30,000 points per batch, with a batch size of 8, and train for 100 epochs. For object classification, we input a fixed number of 1,024 points per batch, with a batch size of 32, and train for 250 epochs. For part segmentation, we input a fixed number of 2,048 points per batch, with a batch size of 32, and train for 250 epochs. In the down-sampling process, for object classification and part segmentation, we employ farthest point sampling, retaining only half of the remaining points at each stage. For semantic segmentation, we employ grid sampling with linear time complexity. The initial grid size is set to 0.04m for S3DIS and 0.02m for ScanNet v2, and doubled at each stage.

B.3. Dataset Description

For semantic segmentation, we conduct experiments on S3DIS [1] and ScanNet v2 [11]. The S3DIS comprises 272 rooms from six large-scale indoor areas. Each point is annotated with a specific semantic label from 13 classes. The ScanNet v2 comprises 1,513 room scans reconstructed from RGB-D frames. The dataset is divided into 1,201 scenes for training, 312 for validation and 100 for online testing. Each point is annotated with a specific semantic label from 20 classes. We use mIoU to assess performance on both datasets: 6-fold cross-validation and Area 5 for S3DIS, and validation and online test sets for ScanNet v2.

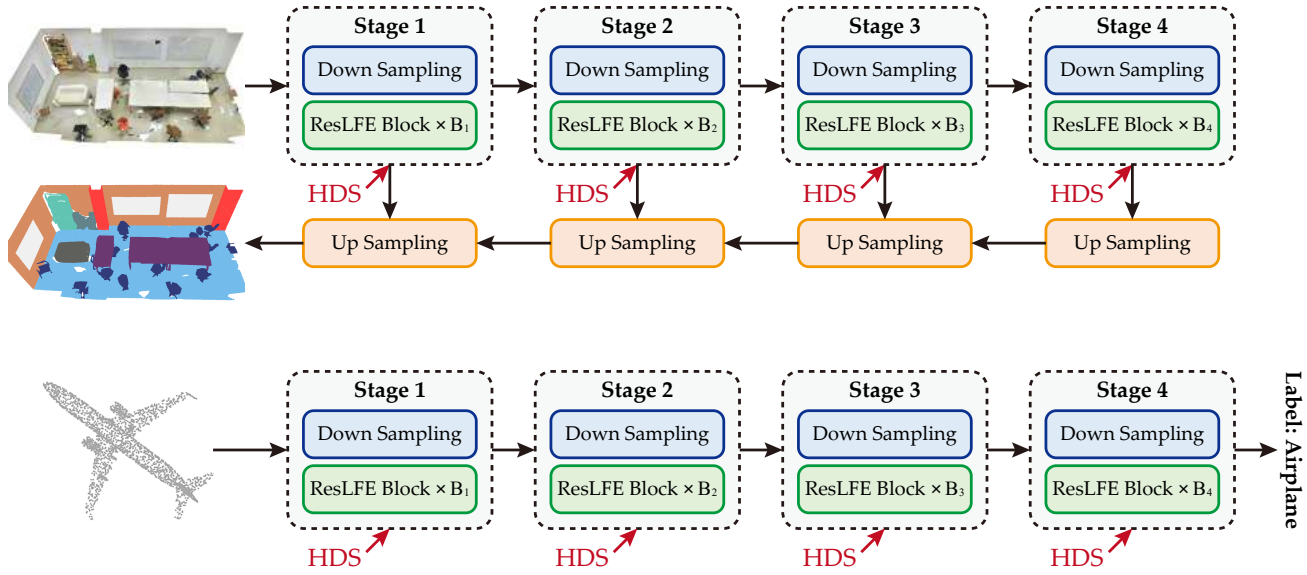


Figure 9. Architecture of the proposed DeepLA-Net for segmentation (top) and classification (bottom). **HDS** denotes hybrid deep supervision strategy. B_n denotes the number of blocks in the n-th stage.

Table 8. Ablation results of the bottleneck in ResLFE block in DeepLA-24 on S3DIS Area5.

Block Ratio	mIoU (%)	△ (%)
[1:1:3:1]	**73.2**	-
[1:1:9:1]	72.7	−0.5
[1:1:1:1]	72.5	−0.7
[5:9:5:5]	72.2	−1.0

For object classification, we conduct experiments on ScanObjectNN [81]. The ScanObjectNN contains about 15,000 real scanned objects, each annotated with a semantic label from 15 classes. Due to the existence of background elements, noise, and occlusions, ScanObjectNN poses significant challenges to the existing point cloud analysis methods. Following PointMLP [48] and PointNeXt [59], we conduct experiments on PB_T50_RS, the hardest and most commonly used variant of ScanObjectNN.

For part segmentation, we conduct experiments on ShapeNetPart [101]. The ShapeNetPart provides part-level annotation for 3D models, comprising 16,880 models across 16 distinct shape classes. Each class has 2-6 parts, amounting to a total of 50 part labels.

C. Additional discussion

C.1. Analysis on ResLFE Block Ratio

In the DeepLA-Net implementation, we set the ResLFE block ratio in encoder stages of [1:1:3:1]. As shown in Table 8, we implement different ResLFE block ratio on DeepLA-24. It is evident that the [1:1:3:1] block ratio we used achieves the best performance.

C.2. More Visual Comparison of Feature Learning with Different Network Depths

We present the visualization of the feature similarity matrix for a specific object class in 3D scenes in Figure 10. From the visualization results, it is evident that DeepLA-120 demonstrates clear segmentation boundaries. While DeepLA-24 is effective, it displays some blurred edges and occasional recognition errors. The simplest DeepLA-6 exhibits a significant number of recognition errors. These findings highlight that the feature similarity matrix of deeper LANets is more accurate and reliable, revealing more pronounced in feature differences compared to surrounding objects. This further demonstrates the enhanced capability of deep networks in feature learning, indicating that a reasonable deepening of LANets can significantly improve its ability to capture local patterns, including edge segmentation and object recognition.

C.3. Exploring to Deeper Networks

We explore aggressively deeper networks of 240 and 360 blocks. These networks are trained and tested on a single Nvidia A6000 GPU with 48GB memory, while keeping other settings consistent with DeepLA-120. As shown in Figure 11, we observe that DeepLA-240/360 exhibit better training accuracy, indicating the potential benefits of further deepening networks. However, the test results of DeepLA-240/360 are inferior to our DeepLA-120, as detailed in Ta-

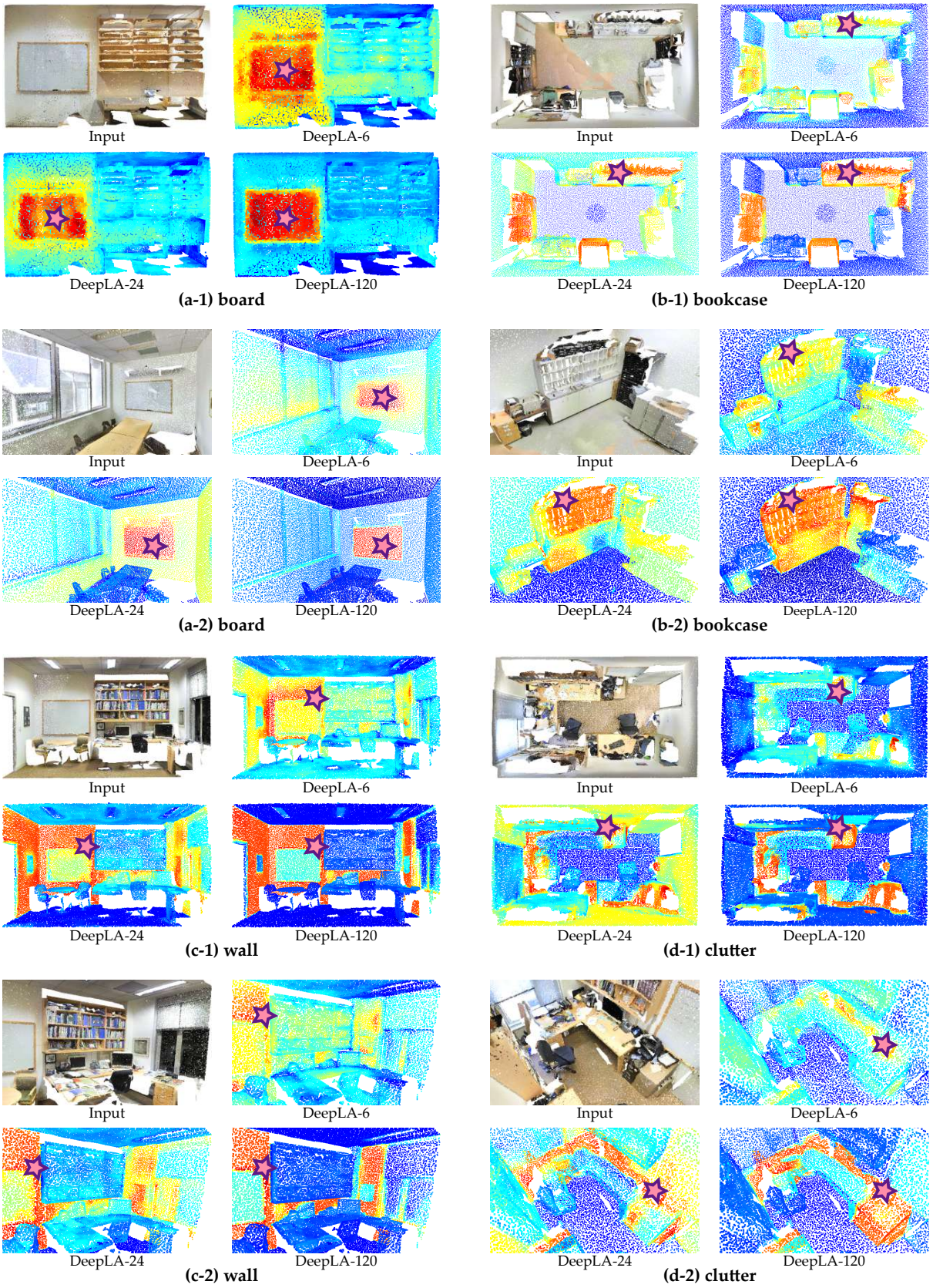


Figure 10. Visual comparison of feature similarity matrix for a specific object class predicted by DeepLA-Net of different depths. The pink stars illustrate the selected center points

ble 9. We attribute this discrepancy to overfitting. Given that point cloud data can be challenging to acquire and annotate, DeepLA-240/360 may be excessively large, potentially necessitating additional strong regularization and data augmentation methods for improved outcomes. We plan to further investigate this in future work.

Table 9. Quantitative comparisons of performance, model complexity, and latency on S3DIS Area5.

Method	mIoU (%)	Params. (M)	FLOPs (G)	Thr. Put (ins./sec.)
DeepLA-120	**75.7**	30.3	42.7	42
DeepLA-240	74.5	61.2	83.4	23
DeepLA-360	74.2	90.7	134.6	15

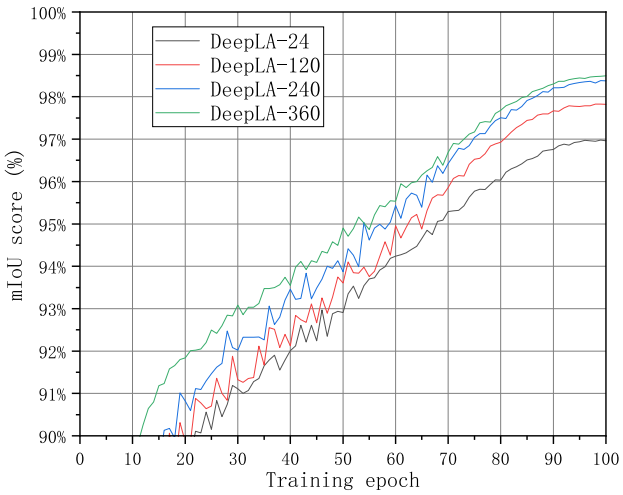


Figure 11. Training performance (mIoU score) across training epochs for deeper DeepLA-Net family on S3DIS (Area 5).

D. Additional Semantic Segmentation Results

D.1. Quantitative Comparisons

In this section, we demonstrate the per-class IoU for S3DIS Area5 (Table 11), 6-fold (Table 12), and ScanNet v2 test set (Table 13). Note that, since many methods do not show the detailed per-class IoU in the semantic segmentation task, here we only compare the methods that present per-class IoU in their papers or have released their code and model weights. For per-class IoU on the S3DIS Area5 and 6-fold, we observe that DeepLA-120 achieves the best or sub-best performance in almost all classes. This demonstrates the potential of DeepLA-Net in local pattern acquisition. Meanwhile, the DeepLA-Net family performs competitively on large-scale objects such as walls, columns, and windows. We conjecture that DeepLA-Net can obtain long-range information with further deepening of the network. Similarly, the proposed DeepLA-120 achieves best or sub-best performance in most classes in per-class IoU on the ScanNet v2 dataset, underscoring the generalizability of DeepLA-Net family.

D.2. Qualitative Comparisons

In this section, for a more perceptible comparison between various methods, we qualitatively assessed the semantic segmentation outcomes produced by PointVector-XL [12] (the best model of PointVector family) and our DeepLA-120 on S3DIS and ScanNet v2 (validation set), as illustrated in Figure 12 and Figure 13. The red boxes highlight regions where the segmentation is inaccurate or the boundary is inconspicuous in PointVector. For the S3DIS dataset, it is visually evident that our segmentation of *clutter*, *columns*, *boards*, and *bookcases* is superior to that of PointVector. These classes are challenging since they usually looks very similar to the *wall*. For example, the board, column and wall in the last row of Figure 12 have slightly different geometric shapes from one another, requiring the network to model long-range dependencies. For the ScanNet v2 dataset, the proposed DeepLA-120 can segment the boundaries more smoothly and accurately.

D.3. Discussion with PTv3 in ScanNet

Unlike our method on ScanNet v2 [11], PTv3 [88] relied on additional data for pre-train. More importantly, PTv3's open-source code reveals the use of extensive test-time augmentation (TTA), which can significantly boost performance. As highlighted in our paper, we did not utilize TTA. To ensure a fair evaluation, we disabled TTA during the testing phase of PTv3. In this case, PTv3 achieves an mIoU of only 76.3% on the validation set **using their provided weights**, which is lower than our DeepLA-120 (77.6%).

Table 10. Quantitative comparisons with PTv3 on ScanNet v2.

Method	ScanNet val
DeepLA-120	**77.6**
PT v3 (*w/o pretrain*)	77.5
– w/o TTA	76.3

E. Additional Related Works

E.1. Deep Neural Network Architecture

In the field of 2D image processing, CNNs have been deepening since the introduction of the pioneering AlexNet [31], leading to a continuous enhancement in network fitting capability. VGG [68] builds upon AlexNet by stacking small-sized convolution filters, significantly increasing network depth and substantially improving performance. Following this, ResNet [24] introduces a simple and efficient skip connection, making it possible to further deepen the network layers. The great success of ResNet not only demonstrates the effectiveness of reasonably increasing network depth, but also inspires subsequent researches to the application and exploration of deep neural architecture [21, 23, 26, 52, 91, 92].

In the field of 3D point cloud processing, researchers have largely 'avoided' exploring network depth, primarily constrained by the historical philosophy of designing networks with more complex local representation. For example, ASSANet [58] also uses pre-linear, which is also employed in our DeepLA-Net, it has an extremely complex design for the local aggregation module with 118M parameters (ASSANet-L only with 8 blocks). In contrast, our approach avoids such complex and redundant design and thus the DeepLA-24 only has 6M parameters. Although some recent works [12, 42, 48, 59, 102] incrementally increased the depth of their networks (about 10-20 blocks), these designs essentially aim to increase the number of parameters for scale-up. In this paper, instead of deliberately following the prevailing trend in the 3D vision community of exploring sophisticated details, we pursue an empirically powerful and very deep architecture for point cloud analysis.

E.2. Deep Supervision

Deep supervision is initially proposed to address the issues of gradient vanishing and slow convergence speed during the network training [36, 74]. This effective training technique has also been applied to improve performance [6, 72, 79, 104, 115]. Lee et al. [36] demonstrate that deep supervised layers can enhance the learning capabilities of hidden layers. This encourages intermediate layers to learn discriminative features, thereby enabling faster convergence and regularization of the network. Dou et al. [15] introduce a deep supervision paradigm to address optimization challenges by supervising predictions from feature maps at varying resolutions. Deep supervision can also be employed to deepen networks. Wang et al. [84] employ a gradient-based heuristic approach to enhance gradient propagation for the training of deeper neural networks. Zhang et al. [109] employ cross-entropy loss to supervise feature maps at different scales in ResNet-50, ensuring the precise capture of context and global information in deeper neural networks. Building on these insights, we have constructed very deep LANets enhanced with deep supervision, to ensure smooth gradient backpropagation in deep networks and to mitigate training optimization challenges.

F. Future Work

Despite the encouraging results, this paper still serves as a start-up work on very deep LANets. Due to the high costs associated with acquiring and annotating point cloud data, the scale of available datasets is significantly smaller compared to 2D images. The limited scale means that deep networks might be more prone to overfitting when applied to point cloud data. In future work, we plan to delve into regularization strategies for DeepLA-Net. Additionally, recent studies are exploring pre-trained models for 3D point clouds. Integrating DeepLA-Net with 3D pre-training strategy could be a promising direction for future research. Our work has the potential to contribute significantly to the development of 3D foundation models.

G. Acknowledgments

This study is supported by the State Key Program of National Natural Science Foundation of China (52332010), the National Natural Science Foundation of China (42471480), the Major Program (JD) of Hubei Province (2023AA02604). The numerical calculations in this article have been done on the supercomputing system in the Supercomputing Center of Wuhan University.

Table 11. Quantitative comparisons with the state-of-the-art methods on S3DIS Area5. **Bold** indicates the best result, underline indicates the best result excluding ours. We only report methods which have demonstrated per-class IoU in their papers.

Method	OA	mIoU	ceil.	floor	wall	beam	col.	wind.	door	table	chair	sofa	book.	board	clut.
PointNet [56]	-	49.0	88.8	97.3	69.8	0.0	3.9	46.3	10.8	59.0	52.6	5.9	40.3	26.4	33.2
PointCNN [39]	85.9	57.3	92.3	98.2	79.4	0.0	17.6	22.8	62.1	74.4	80.6	31.7	66.7	62.1	56.7
KPConv [77]	-	67.1	92.8	97.3	82.4	0.0	23.9	58.0	69.0	81.5	91.0	75.4	75.3	66.7	58.9
PointASNL [98]	87.7	62.6	94.3	98.4	79.1	0.0	26.7	55.2	66.2	83.3	86.8	47.6	68.3	56.4	52.1
RandLA-Net [25]	87.2	62.5	92.1	97.3	80.9	0.0	21.4	61.4	37.4	78.3	87.1	65.8	70.4	67.7	52.2
Point Trans. [114]	90.8	70.4	94.0	98.5	86.3	0.0	38.0	63.4	74.3	89.1	82.4	74.3	80.2	76.0	59.3
GSLCN [41]	90.5	68.1	94.3	98.5	82.9	0.0	20.6	59.4	69.8	83.1	91.4	76.9	75.4	72.5	60.7
PointNeXt [59]	90.6	70.5	94.2	98.5	84.4	0.0	37.7	59.3	74.0	83.1	91.6	77.4	77.2	78.8	60.6
Stra. Trans. [32]	91.5	72.0	96.2	98.7	85.6	0.0	46.1	60.0	76.8	**92.6**	84.5	77.8	75.2	78.1	64.0
PointVector [12]	91.6	72.6	95.6	98.6	85.9	0.0	40.1	61.9	76.4	84.9	**92.4**	**80.9**	78.5	**84.4**	64.6
PointMeta [42]	91.3	72.2	95.4	98.6	85.0	0.0	44.1	61.2	79.0	83.7	92.0	80.8	77.8	78.4	63.2
(Ours) DeepLA-24	91.6	73.2	94.6	98.3	86.9	0.0	48.4	65.5	79.7	88.0	91.1	78.9	77.4	78.9	64.2
(Ours) DeepLA-60	92.0	74.8	95.9	98.6	87.7	0.0	50.2	67.5	**86.0**	90.5	91.8	79.1	78.4	80.3	65.2
(Ours) DeepLA-120	**92.6**	**75.7**	**96.4**	**98.9**	**88.5**	0.0	**53.3**	**71.4**	82.7	92.1	92.2	78.0	**82.0**	81.5	**66.9**

Table 12. Quantitative comparisons with the state-of-the-art methods on S3DIS (6-fold). **Bold** indicates the best result, underline indicates the best result excluding ours. We only report methods which have demonstrated per-class IoU in their papers.

Method	OA	mIoU	ceil.	floor	wall	beam	col.	wind.	door	table	chair	sofa	book.	board	clut.
PointNet [56]	78.6	47.6	88.0	88.7	69.3	42.4	23.1	47.5	51.6	54.1	42.0	9.6	38.2	29.4	35.2
PointCNN [39]	88.1	65.4	94.2	97.3	75.8	63.3	51.7	58.4	57.2	71.6	69.1	39.1	61.2	52.2	58.6
KPConv [77]	-	70.6	93.6	92.4	83.1	63.9	54.3	66.1	76.6	57.8	64.0	69.3	74.9	61.3	60.3
RandLA-Net [25]	88.0	70.0	93.1	96.1	80.6	62.4	48.0	64.4	69.4	69.4	76.4	60.0	64.2	65.9	60.1
BAAF-Net [62]	88.9	72.2	93.3	96.8	81.6	61.9	49.5	65.4	73.3	72.0	83.7	67.5	64.3	67.0	62.4
LEARD-Net [106]	89.1	72.5	94.2	96.9	81.8	65.1	50.9	69.9	72.5	70.6	78.2	68.6	67.2	66.1	60.3
LACV-Net [107]	89.7	72.7	94.5	96.7	82.1	65.2	48.6	69.3	71.2	72.7	78.1	67.3	67.2	70.9	61.6
PointTrans. [114]	90.2	73.5	94.3	97.5	84.7	55.6	58.1	66.1	78.2	77.6	74.1	67.3	71.2	65.7	64.8
U-Next [104]	89.5	73.2	93.6	96.9	84.2	66.1	54.6	67.6	75.5	73.6	74.5	62.9	66.2	74.0	61.7
DeepViewAgg. [64]	-	74.7	90.0	96.1	85.1	66.9	56.3	71.9	78.9	79.7	73.9	69.4	61.1	75.0	65.9
PointNeXt [59]	90.3	74.8	94.2	96.8	85.0	61.5	64.2	68.5	78.7	77.0	70.1	72.4	70.9	70.3	63.3
SPTrans. [63]	-	76.0	93.9	96.3	84.3	71.4	61.3	70.1	78.2	**84.6**	74.1	67.8	**77.1**	63.6	65.0
PointVector[12]	91.8	78.4	95.3	97.5	86.2	64.8	65.2	69.5	81.6	77.8	**89.3**	75.6	72.2	73.9	**70.2**
PointMeta [42]	91.4	77.0	94.9	97.6	85.6	64.4	62.8	68.2	82.1	77.1	83.8	75.4	71.1	70.1	68.5
(Ours) DeepLA-24	91.4	77.9	94.2	96.9	87.0	74.5	68.5	72.5	80.4	76.4	76.9	77.0	71.3	71.3	65.7
(Ours) DeepLA-60	91.9	79.0	94.8	97.6	88.2	**76.2**	69.9	73.6	**82.7**	78.0	77.6	**78.1**	72.1	71.8	66.8
(Ours) DeepLA-120	**92.3**	**79.8**	**95.5**	**97.8**	**89.5**	75.0	**70.3**	**74.8**	82.3	77.2	78.1	77.3	75.1	**75.7**	69.2

Table 13. Quantitative comparisons with the state-of-the-art methods on ScanNet v2 (test set). **Bold** indicates the best result, underline indicates the best result excluding ours. We only report methods which have demonstrated per-class IoU in their papers.

Method	mIoU	bathtub	bed	bookshelf	cabinet	chair	counter	curtain	desk	door	floor	other furniture	picture	refrigerator	shower curtain	sink	sofa	table	toilet	wall	window
PointNet++ [57]	55.7	73.5	66.1	68.6	49.1	74.4	39.2	53.9	45.1	37.5	94.6	37.6	20.5	40.3	35.6	55.3	64.3	49.7	82.4	75.6	51.5
KPConv [77]	68.4	84.7	75.8	78.4	64.7	81.4	47.3	77.2	60.5	59.4	93.5	45.0	18.1	58.7	80.5	69.0	78.5	61.4	88.2	81.9	63.2
PointASNL [98]	66.6	70.3	78.1	75.1	65.5	83.0	47.1	76.9	47.4	53.7	95.1	47.5	27.9	63.5	69.8	67.5	75.1	55.3	81.6	80.6	70.3
RandLA-Net [25]	64.5	77.8	73.1	69.9	57.7	82.9	44.6	73.6	47.7	52.3	94.5	45.4	26.9	48.4	74.9	61.8	73.8	59.9	82.7	79.2	62.1
Stra. Trans. [32]	74.7	90.1	80.3	84.5	75.7	84.6	51.2	82.5	69.6	64.5	95.6	57.6	26.2	74.4	86.1	74.2	77.0	70.5	89.9	86.0	73.4
Point Trans. v2 [89]	75.2	74.2	80.9	**87.2**	75.8	**86.0**	55.2	89.1	61.0	**68.7**	96.0	55.9	30.4	76.6	**92.6**	76.7	79.7	64.4	**94.2**	87.6	72.2
PointMeta [42]	71.4	83.5	78.5	82.1	68.4	84.6	53.1	86.5	61.4	59.6	95.3	50.0	24.6	67.4	88.8	69.2	76.4	62.4	84.9	84.4	67.5
LargeKernel3D [5]	73.9	90.9	82.0	80.6	74.0	85.2	54.5	82.6	59.4	64.3	95.5	54.1	26.3	72.3	85.8	77.5	76.7	67.8	93.3	84.8	69.4
LRPNet [38]	74.2	81.6	80.6	80.7	75.2	82.8	**57.5**	83.9	**69.9**	63.7	95.4	52.0	32.0	75.5	83.4	76.0	77.2	67.6	91.5	86.2	71.7
Retro-FPN [90]	74.4	84.2	80.0	76.7	74.0	83.6	54.1	**91.4**	67.2	62.6	95.8	55.2	27.2	77.7	88.6	69.6	80.1	67.4	94.1	85.8	71.7
DMF-Net[99]	75.2	90.6	79.3	80.2	68.9	82.5	55.6	86.7	68.1	60.2	96.0	55.5	**36.5**	**77.9**	85.9	74.7	79.5	**71.7**	91.7	85.6	76.4
CondaFormer [16]	75.5	92.7	82.2	83.6	**80.1**	84.9	51.6	86.4	65.1	68.0	95.8	**58.4**	28.2	75.9	85.5	72.8	80.2	67.8	88.0	87.3	75.6
(Ours) DeepLA-120	**77.2**	**93.9**	**82.4**	85.4	77.1	84.0	56.4	90.0	68.6	67.7	**96.1**	53.7	34.8	76.9	90.3	**78.5**	**81.5**	67.6	93.9	**88.0**	**77.2**

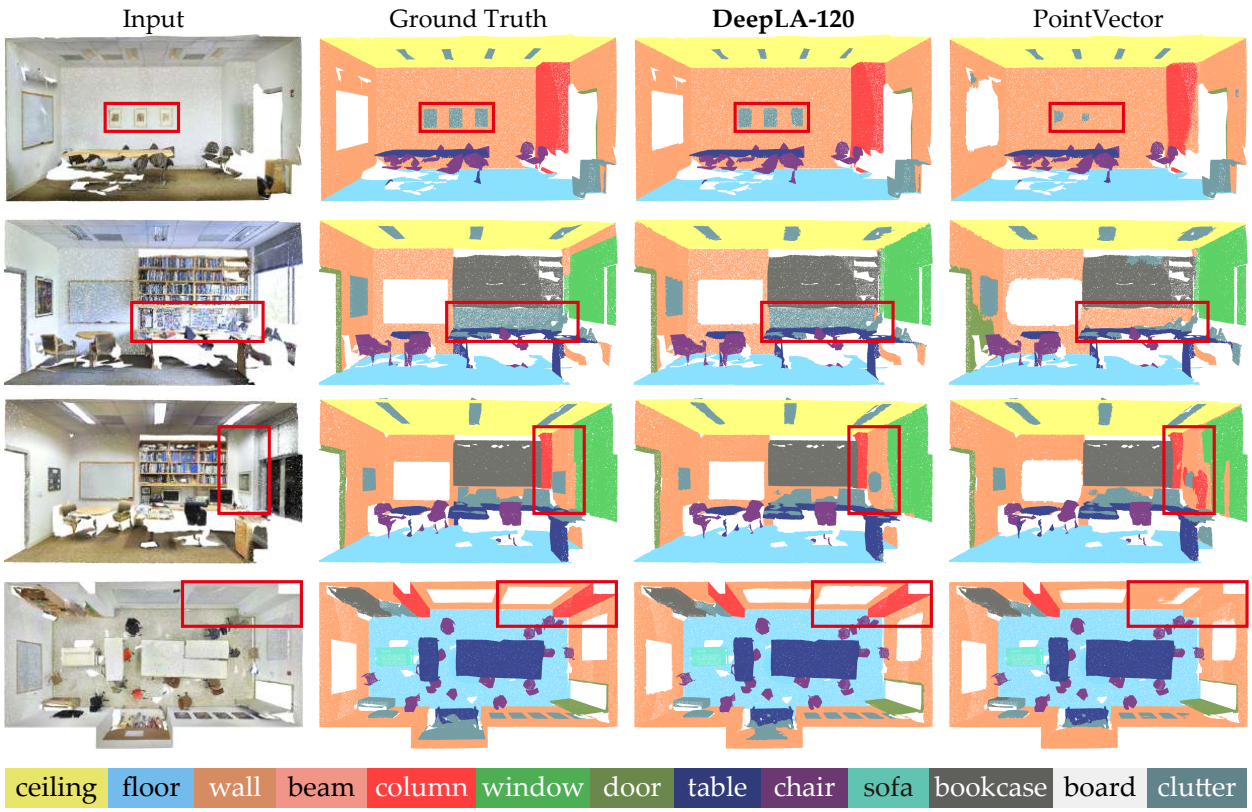


Figure 12. Visual comparison of semantic segmentation results on S3DIS dataset.

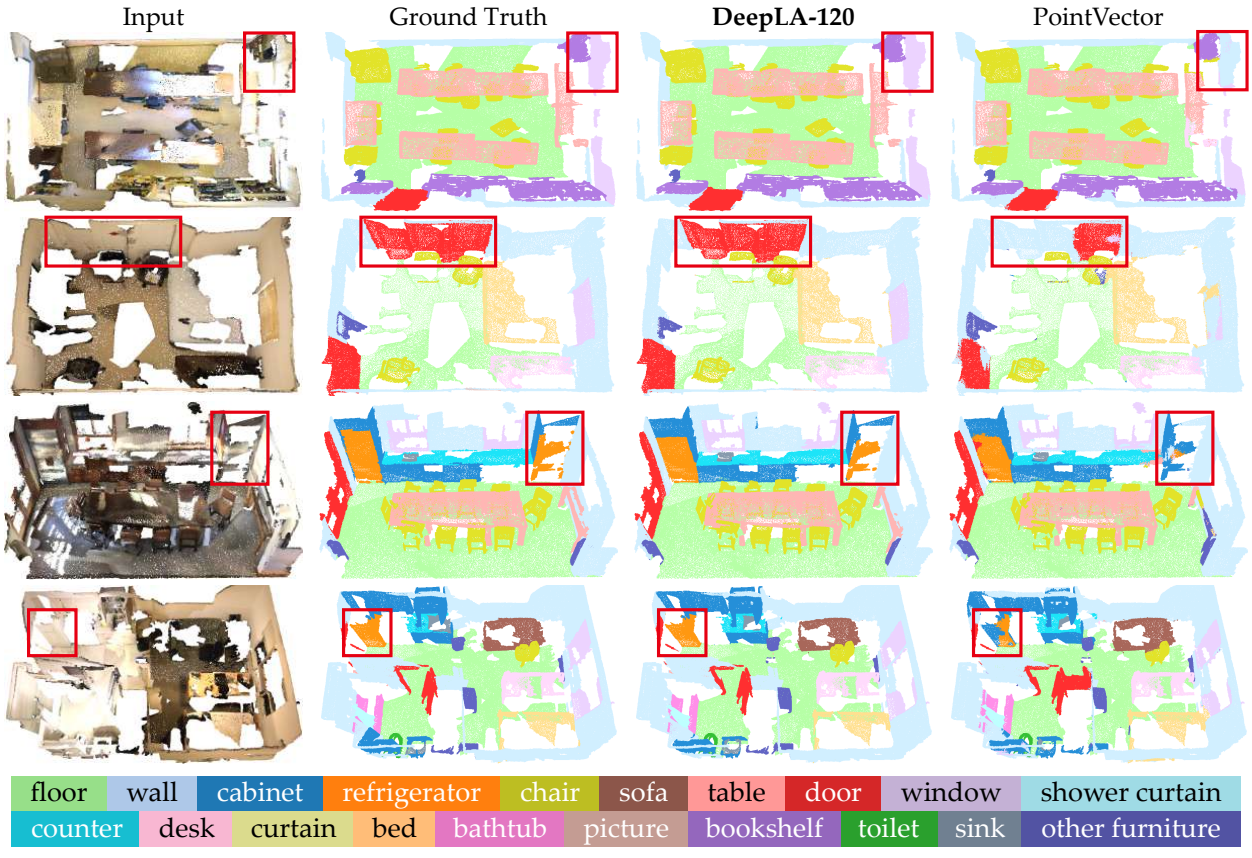


Figure 13. Visual comparison of semantic segmentation results on ScanNet v2 dataset.