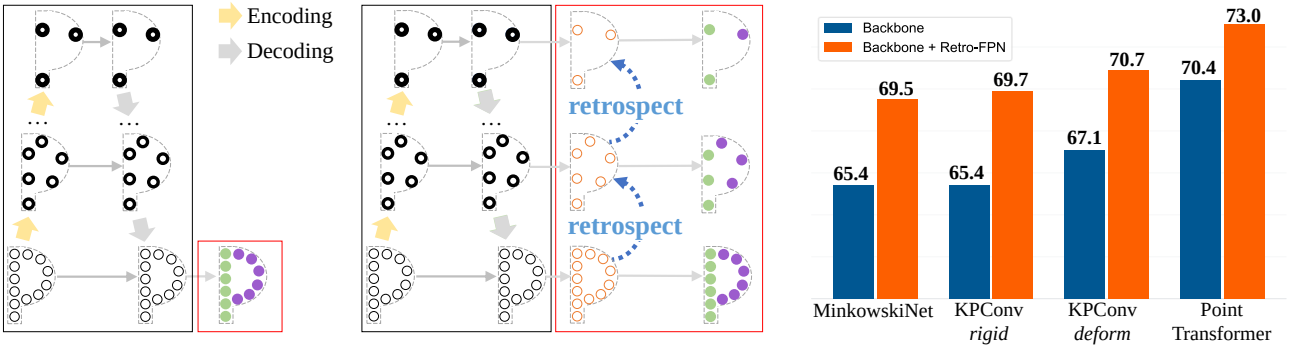
Retro-FPN: Retrospective Feature Pyramid Network for Point Cloud Semantic Segmentation

Anonymous ICCV submission

Paper ID 6304



(a) Encoder-Decoder Architecture (b) Retrospective Feature Pyramid Network (c) Quantitative comparison with backbones

Figure 1. (a) Encoder-decoder architecture with an inherent feature pyramid in the decoding stage. Black points with thicker outlines denote region features of larger local regions, green and purple points are the predicted semantic labels. (b) Retrospective Feature Pyramid Network, the points with orange outlines denote point-level semantic features. The rectangular areas highlighted in black and red denote local region feature learning and point-level semantic feature learning, respectively. In Retro-FPN, region information flows into points at all levels, and are retrospectively refined to the lowest level. (c) mIoU on S3DIS Area 5 with and without Retro-FPN.

Abstract

Learning per-point semantic features from the hierarchical feature pyramid is essential for point cloud semantic segmentation. However, most previous methods suffered from the ambiguous region features or failed to refine per-point features effectively, which leads to information loss and ambiguous semantic identification. To resolve this, we propose Retro-FPN to model the per-point feature prediction as an explicit and retrospective refining process, which goes through all the pyramid layers to extract semantic features explicitly for each point. Its key novelty is a retro-transformer for summarizing semantic contexts from the previous layer and accordingly refining the features in the current stage. In this way the categorization of each point is conditioned on its local semantic pattern. Specifically, the retro-transformer consists of a local cross-attention block and a semantic gate unit. The cross-attention serves to summarize the semantic pattern retrospectively from the previous layer. And the gate unit carefully incorporates the summarized contexts and refines the current semantic features. Retro-FPN is a pluggable neural network that applies to hierarchical decoders. By integrating Retro-FPN with three representative backbones, including both point-based and voxel-based methods, we show that Retro-FPN can significantly improve the performance over state-of-the-art backbones. Comprehensive experiments on widely used benchmarks can justify the effectiveness of our design. The source code will be publicly available.

1. Introduction

3D point cloud semantic segmentation [24, 4, 62, 29, 8, 44, 65, 51, 57], which aims to predict a unique category label for each point, is a critical task towards the 3D visual understanding of large-scale scenes. A typical solution to predict per-point semantic labels is the widely used encoder-decoder framework [19]. The encoder aims to learn contextual region features by gradually enlarging receptive fields. The decoder propagates the local region features from the larger receptive fields into the smaller ones, which inherently forms a feature pyramid [32] (see Figure 1 (a)).

Learning per-point feature prediction from the pyramidal region features is the target of point cloud semantic segmentation. However, most existing encoder-decoder-based networks merely reveal per-point features explicitly at the final

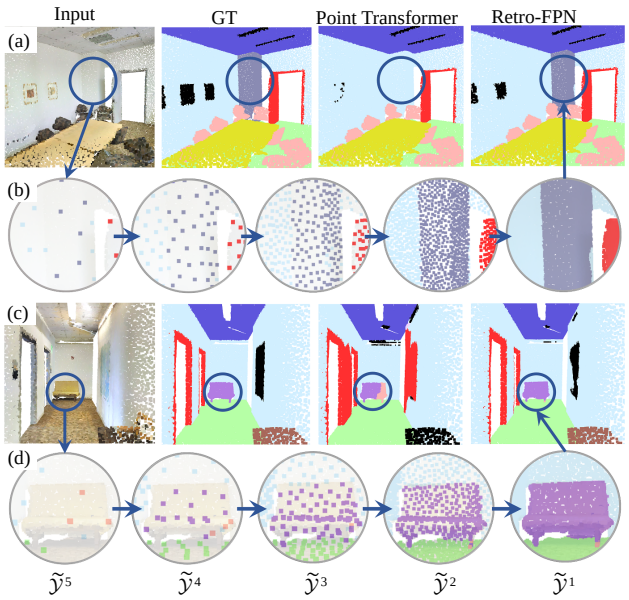


Figure 2. Visualization of segmentation process of Retro-FPN. (a) and (c) show the visual comparison with the backbone (Point Transformer [68]) network. In (a), the backbone loses the information of the column. In (c), the backbone struggles to distinguish between chair and sofa. In (b) and (d), we show the retrospective refining process by Retro-FPN over the improved areas.

layer (denoted as red box in Figure 1 (a)), leaving abundant semantic information stuck in the intermediate region features (black box in Figure 1 (a)), which cannot directly facilitate the final prediction. This may lead to the loss of semantic information and ambiguous semantic identification, as demonstrated in Figure 2 (a) and (c). Since each pyramid layer may contain useful and erroneous information simultaneously, the per-point semantic features should be carefully refined through all stages.

To resolve this, some prior works [14, 25] adopt hierarchical supervision to refine intermediate predictions explicitly. In 2D vision, PointRend [25] proposed to refine high-frequency points with hierarchical supervision, but each point is refined based on the features interpolated at a single location, which suffered to capture the local semantic pattern and may fail to obtain informative per-point features for 3D point clouds. RFCR [14] first introduced multi-scale supervision to point cloud semantic segmentation, but the supervision was on region-level and it's still difficult to obtain accurate per-point prediction from the region features.

Therefore, we propose Retro-FPN to improve per-point semantic feature prediction by fully utilizing the feature pyramid, which is achieved by an explicit and retrospective refining process (see Figure 1 (b)). Specifically, by predicting per-point labels for all the middle layers, Retro-FPN allows region information to flow into points and obtains the point-level semantic features at each stage. Then, the features are carefully refined by retrospectively summarizing the semantic pattern from the previous layer and adaptively rearranging the current semantic information.

To conduct retrospective refinement, we introduce a novel *retro-transformer* in each layer to extract per-point semantic features, which consist of two stages. The first stage aims to "retrospect" useful information from the previous layer. Since the category of each point is similar to its surrounding local region, we use a local cross-attention block to conduct retrospection, which takes the features of the current layer as queries to summarize semantic contexts from the previous layer. Different from the region-level information in the backbone features, such contextual information are built upon the per-point semantic features of the nearby points, which can fully facilitate the refinement of each point by selectively revisiting its neighbor points. The second stage serves to "refine" the current semantic features by combining them with the summarized contexts. Instead of merging the features with simple adding or concatenation, we use a lightweight semantic gate to adaptively preserve and forget the previous semantic information. The retro-transformer can establish a cross-level semantic relationship between different decoding stages, this enables the network to explicitly preserve useful information and discard erroneous information in each stage, as illustrated in Figure 2 (b) and (d).

Retro-FPN is a pluggable neural network that can extract and refine per-point semantic features for prevailing backbones, including both point-based and voxel-based methods. Specifically, we embed Retro-FPN into KPConv [53], MinkowskiNet [6], and Point Transformer [68]. Non-trivial improvements on the S3DIS [1] Area 5 benchmark (Figure 1 (c)) can verify the effectiveness of our network design. In summary, our contributions are threefold:

- We propose Retro-FPN to improve per-point semantic feature prediction for 3D point clouds. Retro-FPN models the feature propagation as an explicit and retrospective refining process on point-level semantic information, which is a plug-and-play network that can improve the performance of prevailing backbones.

- We propose a novel retro-transformer to establish a cross-level semantic relationship between different decoding stages. It utilizes a local cross-attention to retrospect the previous semantic pattern and leverages a lightweight semantic gate unit to refine the current semantic features.

- We integrate Retro-FPN with both point-based and voxel-based backbones and evaluate our method on the S3DIS [1], ScanNet [9] and SemanticKITTI [2] benchmarks. Experimental results demonstrate that our method can significantly improve performance over state-of-the-art methods.

2. Related Work

Point cloud semantic segmentation. Point cloud semantic segmentation methods [23, 28, 66] can be roughly divided into two categories. (1) The point-based [41, 55, 59, 53, 68, 67, 52, 10] methods directly handle raw point clouds. As one of the pioneering works, Point-Net++ [42] used a local sampling and grouping mechanism to extract contextual information. Followers along this line focus on effective feature aggregation technique to obtain representative features, such as convolution-like operations [53, 30] and the attention mechanism [54, 68, 27, 39, 56]. (2) The voxel-based [6, 15] methods first transform 3D point clouds into voxels, then apply sparse convolutions to learn point cloud representations. While these methods can handle large-scale scenes, they also suffer from detailed information loss due to voxelization. For both point-based and voxel-based methods, an encoder-decoder architecture is a typical solution. While previous methods [55, 59] usually highlight the importance on feature aggregation in the encoding stage, we concentrate on the explicit decoding of semantic information to unleash the performance for prevailing backbones.

Pyramidal feature representation. The feature pyramid is an important component of deep neural networks, which can perceive large-scale scenes at different scales. FPN [32] is a pioneering work that leverages the pyramid features to detect multi-scale objects. Since then, the feature pyramid has been explored in 2D dense prediction tasks, such as object detection [13, 48], instance segmentation [34, 12, 18] and panoptic segmentation [24]. Semantic segmentation requires per-point prediction at the final layer, to exploit the feature pyramid, one possible solution is to up-sample intermediate features [31, 37] or predictions to the finest resolution and fuse them like BAAF-Net [45] and PANet [34]. However, each pyramid layer may contain useful and erroneous information simultaneously, simply fusing the intermediate outputs can lead to false predictions. Another solution is to incorporate hierarchical supervision and refine the intermediate predictions by layer. In 2D vision, PointRend [25] proposed to gradually refine points in high-frequency areas, but each point is refined based on the interpolated prediction and features at a single location, which cannot provide adequate local contexts for refinement. Furthermore, the point selection procedure of PointRend is tailored for dense and regular 2D grids, which cannot directly apply to point cloud data. RFCR [14] is one of the first attempts to utilize feature pyramid with hierarchical supervision for 3D point clouds, but it focused merely on enhancing region level semantic features, which is difficult to fully preserve and refine per-point semantic information at each stage.

Compared with the previous methods, Retro-FPN takes a step further to explore a context-aware solution for refining semantic features on per-point level, which is tailored for 3D point clouds. Retro-FPN refines each point based the local semantic contexts retrospected from the previous layer and selectively preserve and forgo semantic information in consecutive layers, which enables to fully unleash the potential of prevailing backbones.

Relation to transformer. Transformer [54] was first proposed for natural language processing and soon became dominant in 2D computer vision [35]. Inspired by this success, many studies [68, 16, 38] have attempted to leverage the representation ability of transformer to process 3D point clouds. Recently, More studies further explored the attention mechanism that caters to point clouds, including the study of long range dependency [27], efficient attention mechanism [39] and powerful local attention [56]. While these methods have made substantial progress, they use self-attention for representation learning in a single stage. Differently, we propose retro-transformer to establish semantic relationships across different decoding stages.

3. Method

3.1. Overview and Motivation

We show a typical encoder-decoder architecture with L levels in Figure 3 (a), and the inherent feature pyramid hierarchy of the decoding stage is shown in Figure 3 (b). Our Retro-FPN is integrated with the backbone decoder and shown in Figure 3 (c). For clarity, we only visualize three pyramid layers (1, 2 and L).

As shown in Figure 3 (b), we denote the point set in each decoding stage as $\mathcal{P}^l \in \mathbb{R}^{N_l \times 3}$, and the local context around $\mathcal{P}^l$ is denoted as the region feature $\mathcal{F}^l \in \mathbb{R}^{N_l \times C_l}$. From $\mathcal{P}^L$ to $\mathcal{P}^1$, the decoder propagates contextual information from the larger receptive (highlighted in the red circle) fields into the smaller ones, and finally to the point-level features $\mathcal{F}^1$. However, there are two problems with this paradigm. First, the backbone decoder propagates semantic information simplicitly, where the long path from the intermediate levels (layer 2-L) to the prediction layer (layer 1) may cause information loss. Second, although the high-level features have large receptive fields, it is still difficult to precisely capture the accurate semantic contexts of the underlying local regions, especially when there are different semantic objects within the same region, e.g., at the boundary of window, wall and bookcase.

Based on the above observation, we propose Retro-FPN extract accurate per-point semantic features from the feature pyramid, which is conducted by explicitly and retrospectively refining the point-level semantic information.

3.2. Retro-FPN

As shown in Figure 3 (c), Retro-FPN is designed to explicitly extract and refine semantic information for all pyramid levels. In level l, the region feature $\mathcal{F}^l$ is first refined and converted into point-level semantic feature $\mathcal{H}^l$ by a retro-transformer. Then, we explicitly predict per-point

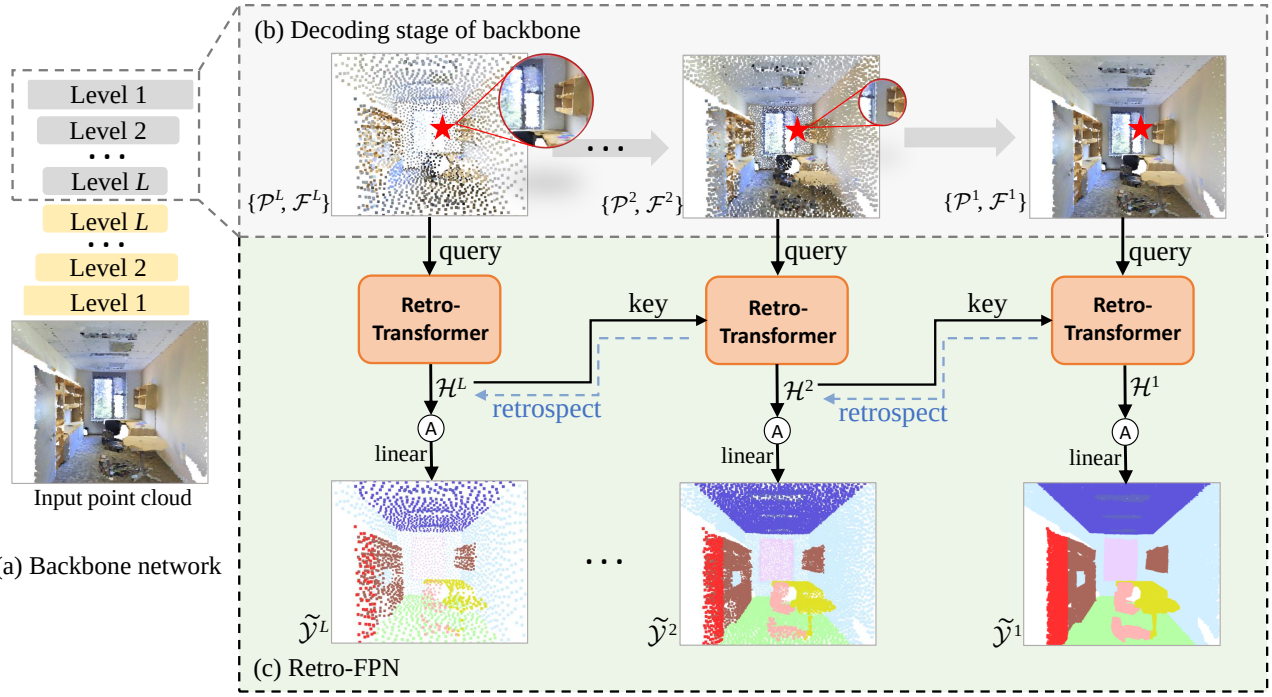


Figure 3. (a) shows an encoder-decoder architecture. (b) In the decoding stage of backbone, only three pyramid layers (1, 2 and L) are shown for clarity, $\mathcal{P}^l$ is point set in each decoding stage, and $\mathcal{F}^l$ is the region feature of $\mathcal{P}^l$. The larger circular area highlighted in red denotes larger local region around $\mathcal{P}^l$, which is characterized by $\mathcal{F}^l$. (c) For Retro-FPN, $\mathcal{H}^{l+1}$ is point-level semantic feature from previous layer, which provides key and value for retro-transformer. $\mathcal{F}^l$ provides query to retrospectively summarize semantic pattern from $\mathcal{H}^{l+1}$.

labels $\tilde{\mathcal{Y}}^l$ from $\mathcal{H}^l$ using an activation function followed by a linear transformation.

There are two advantages to the design of Retro-FPN. First, instead of struggling to perceive the complex local regions like RFCR [14], the explicit prediction of per-point labels allows Retro-FPN to focus on point level semantic information. The intuition is that for a point $\mathbf{p}_i \in \mathcal{P}^l$, it is easier to identify its single semantic category than recognize all the semantic objects within the surrounding local region. This scheme enables Retro-FPN to incorporate accurate semantic information into $\mathcal{H}^l$, which significantly facilitates the retrospective refinement. Second, although the global contexts are essential for scene understanding, the saturated contextual prior could hamper the network to perceive detailed local semantic information [36]. This problem could be even worse in higher pyramid layers. Hence, encouraging the middle layers to focus on per-point semantic information can help the network to balance global scene contexts and the detailed semantic information.

While the overall architecture of Retro-FPN can help to learn accurate per-point semantic information, now the more critical problem is to refine the information and facilitate the final prediction. Since $\mathcal{H}^l$ from intermediate layers ($l > 1$) may contain false semantic information, two goals have to be achieved: (1) preserving useful information and (2) discarding erroneous information. Previous methods like PointRend [25] refine each point based on the coarse prediction and the interpolated features at a single location, but the accurate semantic category of each point is dominated by its local neighborhood, the interpolated features cannot provide adequate semantic contexts for per-point refinement. Differently, we propose a novel *retro-transformer* and leverage attention mechanism to selectively summarize local semantic information. The detailed structure of retro-transformer is described below.

3.3. Retro-Transformer

The structure of Retro-Transformer is shown in Figure 4, which consists of a local cross-attention block (Figure 4 (a)) and a semantic gate unit (Figure 4 (b)). The cross-attention aims to conduct "retrospection". The per-point semantic features from the previous layer can provide rich semantic contexts and guide the current layer. Hence, for each point, we leverage the attention mechanism to attentively summarize semantic contexts by revisiting its neighbor points from the previous layer. Further, The semantic gate serves to achieve "refinement". Because intermediate semantic features will inevitably contain erroneous information, the gate mechanism allows the retro-transformer to selectively retain and forgo information from both the previous and the current layer.

Local cross-attention block. As shown in Figure 4 (a), the cross-attention takes the previous semantic feature $\mathbf{h}^{l+1} \in \mathbb{R}^C$ and the current region feature $\mathbf{f}^l \in \mathbb{R}^C_l$ as inputs to

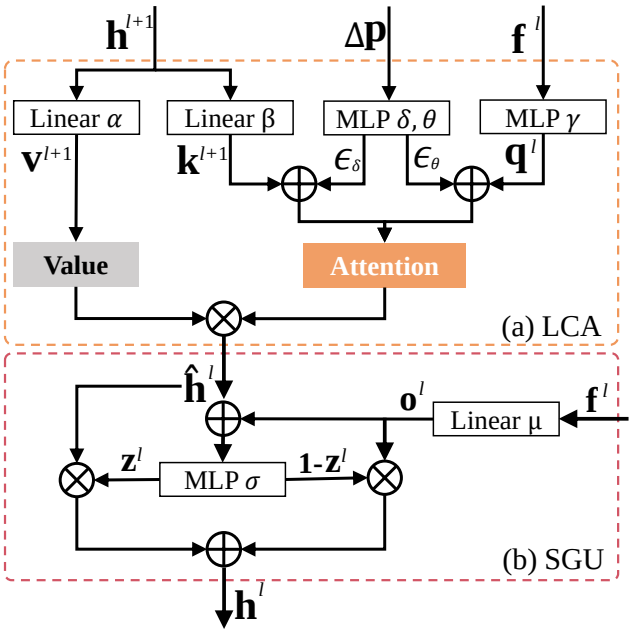


Figure 4. The structure of Retro-Transformer. (a) The local cross-attention (LCA) block. (b) The semantic gate unit (SGU).

summarize semantic contexts. Since $\mathbf{f}^l$ and $\mathbf{h}^{l+1}$ are from different branches and may have large discrepancy, unlike previous transformers [54, 68] that produce the query vector with a linear layer, we use the non-linear transformation of multi-layer perceptron (MLP) to obtain $\mathbf{q}^l \in \mathbb{R}^C$, which can bridge the gap between the two branches with more learnable capacities. Then, the value and key vectors are produced from $\mathbf{h}^{l+1}$ using linear layer as follows:

$$\begin{aligned} \mathbf{q}^l &= \text{MLP}_\gamma(\mathbf{f}^l), \\ \mathbf{v}^{l+1} = Linear_\alpha(\mathbf{h}^{l+1}), &\quad \mathbf{k}^{l+1} = Linear_\beta(\mathbf{h}^{l+1}). \end{aligned} \tag{1}$$

Furthermore, since the semantic information of each point $\mathbf{p}_i^l$ is dominated by the surrounding local region, we adopt local attention to aggregate semantic contexts from its nearby points $\mathbf{p}_j^{l+1}$ (subscript i and j denote the point index) in the previous layer. The neighborhood of $\mathbf{p}_i^l$ is defined as the K-nearest neighbor (K-NN) points. The K-NN strategy lets retro-transformer focus on local semantic contexts, which also reduces computation cost significantly. It is worth noting that the point clouds of the previous layer are usually much sparser than the current ones, so that even a small K-NN search can effectively enlarge receptive field. Moreover, since the complex local region may increase the difficulty for learning robust contexts, we enhance the query and key vectors with learnable position embedding to incorporate positional relationship. Specifically, for each q_i^l, we denote the key vectors of the K-nearest neighbors as $\{\mathbf{k}_{i,k}^{l+1} | k = 1, 2, \ldots, \text{K}\}$, where subscript k denotes the k-th neighbor and calculate attention as:

$$w_{ik} = \langle \mathbf{q}_i^l + \epsilon_\delta,\, \mathbf{k}_{i,k}^{l+1} + \epsilon_\theta \rangle / \sqrt{C}, \tag{2}$$

where the position embedding ϵ_δ and ϵ_θ are obtained by passing the relative position $\Delta\mathbf{p}$ ($\Delta\mathbf{p} = \mathbf{p}_i^l - \mathbf{p}_{i,k}^{l+1}$) through two MLPs. Then, the aggregated semantic contexts $\hat{\mathbf{h}}_i^l$ are given as follows:

$$\hat{\mathbf{h}}_i^l = \sum_{k=1}^{\text{K}} Softmax(\mathbf{w}_i)_k \mathbf{v}_{i,k}. \tag{3}$$

Note that there is no $\mathbf{h}^{L+1}$ for the highest pyramid layer (the L-th layer), where the cross-attention degrades to self-attention and takes $\mathbf{f}^L$ as query, key and value.

Semantic gate unit. As shown in Figure 4 (b), we refine the region feature $\mathbf{f}^l$ with the summarized semantic contextual feature $\hat{\mathbf{h}}^l$ using the gate mechanism. To reduce computation cost, we take inspiration from gated recurrent unit (GRU) [7] and adopts a single update gate to control information flow. Specifically, given region feature $\mathbf{f}^l \in \mathbb{R}_l^C$, we first compacts its information into vector $\mathbf{o}^l \in \mathbb{R}^l$ by $\mathbf{o}^l = Linear_\mu(\mathbf{f}^l)$. Then, the update gate $\mathbf{z}^l$ is given as:

$$\mathbf{z}^l = \text{MLP}_\sigma(\hat{\mathbf{h}}^l + \mathbf{o}^l). \tag{4}$$

Finally, we obtain the point-level semantic feature $\mathbf{h}^l$ by the following equation:

$$\mathbf{h}^l = \mathbf{z}^l \odot \hat{\mathbf{h}}^l + (1 - \mathbf{z}^l) \odot \mathbf{o}^l. \tag{5}$$

3.4. Integration with backbones

Retro-FPN can be integrated with prevailing backbones that adopt an encoder-decoder architecture, including both point-based and voxel-based methods. To employ Retro-FPN, we only need the point set $\mathcal{P}^l$ of each decoding stage, the corresponding region feature $\mathcal{F}^l$ and the ground-truth label $\mathcal{Y}^l$. For point-based methods, we record the ground-truth labels $\mathcal{Y}^l$ along the downsampling process of the encoding stage, and directly use $\mathcal{F}^l$ from the decoder. For voxel-based methods, we take the voxels in each layer as intermediate point clouds and also focus on learning per-point semantic information from the voxel features. Since each voxel may correspond to multiple category labels, we use the most common one as its ground-truth label. Moreover, for both point-based and voxel-based backbones, the intermediate layer may contain too many points (voxels) due to small downsampling rates, which severely increases computation cost. Meanwhile, the K-NN search in a dense point cloud also leads to limited receptive fields. To avoid the above problems, we further use random sampling to downsample the intermediate point clouds.

3.5. Training loss

We use cross entropy loss to guide the predictions from all decoding stages, the training loss is formulated as $\mathbf{L} = \sum \lambda_l \mathbf{L}_l$, where $\mathbf{L}_l$ is the loss of the l-th layer. λ_l is the weight to balance losses in each layer, which is empirically set to 1.0 in our experiments.

Table 1. Quantitative results on S3DIS [1] dataset, evaluated on Area 5. Red number means better results than baseline. **Bold** numbers denote the best results among all methods.

Method	Input	mIoU
RFCR [14]	point	68.7
DeepViewAgg [47]	point + 2D	67.2
RepSurf [46]	point	68.9
CBL [50]	point	71.0
Fast Transformer [39]	point	70.3
EQ-Net [63]	voxel	71.3
Stratified Transformer [27]	point	72.0
Point Mixer [5]	point	71.4
Point Transformer V2 [56]	point	71.6
MinkowskiNet (5cm) [6]	voxel	65.4
MinkowskiNet + Retro-FPN	voxel	69.5
KPConv *rigid* [53]	point	65.4
KPConv *rigid* + Retro-FPN	point	69.7
KPConv *deform* [53]	point	67.1
KPConv *deform* + Retro-FPN	point	70.7
PointTransformer [68]	point	70.4
PointTransformer + Retro-FPN	point	**73.0**

4. Experiments

4.1. Datasets and metric

S3DIS. The S3DIS [1] dataset comprises point clouds of 271 rooms in six areas. There are 273 million points in total, and each point is assigned a semantic label of 13 categories. Following previous methods [42, 52, 68], we evaluate our method on the Area 5 and 6-fold benchmarks.

ScanNet v2. The ScanNet v2 [9] provides 1,613 indoor scans, where the train/val/test split is 1,2101/312/100, respectively. The training and validation sets contain point-level annotations, and the test set is provided without ground-truth annotations.

SemanticKITTI. The SemanticKITTI [2] dataset provides 43,553 LIDAR scans that belong to 21 sequences. The training set contains 19,130 scans from sequences 00-07 and 09-10, and the validation set has 4,071 scans from sequence 08. The testing set contains 20,351 scans from sequences 11-21, which is set for online testing and only the 3D coordinates are provided.

Evaluate metric. For the above benchmarks, we adopt the mean Intersection-over-Union (mIoU) as evaluation metric.

4.2. Backbones and experimental settings

Backbones. On the S3DIS [1] Area 5 benchmark, we embed Retro-FPN into both point-based (Point Transformer [68] and KPConv [53]) and voxel-based [6] methods to prove the generalization ability of Retro-FPN. Since the six areas of S3DIS have large discrepancies, we further choose the high-performing Point Transformer to evaluate the robustness of Retro-FPN on the S3DIS 6-fold benchmark. As for the ScanNet [9] and SemanticKITTI [2] datasets, we use MinkowskiNet as backbone, because it is a more popular choice that has been widely adopted as backbone by previous methods like BPNet [20] and SPVNAS [49].

Table 2. Quantitative results on S3DIS [1] dataset, evaluated on 6-fold cross validation.

Method	mIoU
KPConv [53]	70.6
FPConv [33]	68.7
PAConv [59]	69.3
SCF-Net [11]	71.6
CBL [50]	73.1
DeepViewAgg [47]	74.7
RepSurf [46]	74.3
EQ-Net [63]	**77.5**
PointNeXt [43]	74.9
PointTransformer [68]	73.5
PointTransformer + Retro-PFN	77.3

Experimental settings. We implement Retro-FPN using PyTorch [40]. To have fair and solid experiments, we integrate Retro-FPN based on the official implementation of the baseline methods and keep the experimental settings the same as the backbones. We provide more experimental details in the supplementary materials.

4.3. Quantitative results

S3DIS Area 5. Table 1 shows the results of point cloud semantic segmentation on the S3DIS [1] Area 5 benchmark, from which we can find that Retro-FPN can significantly improve the segmentation performance of the backbone networks. Particularly, we achieve the best performance by integrating Retro-FPN with Point Transformer [68] and yield a state-of-the-art record of 73.0 in terms of mIoU. Additionally, by integrating with KPConv *deform*, Retro-FPN is able to improve the overall performance by 3.6 in terms of mIoU. It is worth noting that RFCR [14] also adopts KPConv *deform* as backbone, which improves performance (1.6 on mIoU) by enhancing the feature pyramid on region level semantic information. Compared with RFCR, Retro-FPN can better stimulate the potential of the backbone network (3.6 versus 1.6 in terms of mIoU improvements over KPConv *deform*), this should be credited to the retrospective refinement on point-level semantic features. Furthermore, by assembling with KPConv *rigid* [53], Retro-FPN is able to significantly raise mIoU by 4.3. In addition, Retro-FPN can also improve the voxel-based MinkowskiNet [6] by 4.1 in terms of mIoU. Note that the intermediate layers of voxel-based methods lack precise per-point information due to the convolution, our Retro-FPN can complement the drawback and explicitly extracts point-level semantic infor-

Table 3. Quantitative results on ScanNet v2 [9] in terms of mIoU.

Method	Val	Test
KPConv [53]	69.2	68.6
JSENet [22]	-	69.9
FusionNet [64]	-	68.8
SparseConvNet [15]	69.3	72.5
BPNet [20]	73.9	74.9
VMNet [21]	73.3	74.6
StratifiedFormer [27]	74.3	74.7
EQ-Net [63]	75.3	74.3
Point Transformer V2 [56]	**75.4**	**75.2**
MinkowskiNet (5cm) [6]	68.0	-
+ Retro-FPN	70.4	-
MinkowskiNet (2cm) [6]	72.1	73.6
+ Retro-FPN	74.0	74.4

mation from voxel features.

S3DIS 6-fold. In Table 2, we show the quantitative results of the 6-fold cross validation on S3DIS [1] dataset. From Table 2, we can find that Retro-FPN can significantly improve over Point Transformer by 3.8 absolute percentage points. The result indicates that although the Point Transformer is a strong baseline, it still suffers from the information loss of implicit region features and Retro-FPN can still improve its performance robustly.

ScanNet V2. In table 3, we evaluate the performance of Retro-FPN on ScanNet v2 [9] dataset. We follow the same practice of [20, 6, 36] and adopt MinkowskiNet as the backbone to conduct experiments under voxel size 2cm and 5cm. As shown in Table 3, Retro-FPN is able to improve the segmentation performance under various voxel sizes, where Retro-FPN raises mIoU by 2.4 and 1.9 under voxel size of 5cm and 2cm, respectively. Also, Retro-FPN improves the result on the test set to 74.4.

SemanticKITTI. Besides indoor datasets, we also integrate Retro-FPN with MinkowskiNet [6] and evaluate its performance on the SemanticKITTI benchmark. Following the same experimental settings of SPVNAS [49], we report the mIoU on both the validation and test sets. From Table 4, we can find that Retro-FPN can improve the mIoU by 3.5 and 4.9 on the validation and test sets, respectively. The results on both the indoor and outdoor benchmarks can well demonstrate the effectiveness of Retro-FPN.

4.4. Qualitative results

In Figure 5, we give the visualization results of Retro-FPN and the qualitative improvements over the backbone (MinkowskiNet [6]). Moreover, we also visualize the refining process of semantic labels in each layer, which is highlighted in black circles. The visual results show that Retro-FPN can help to improve segmentation in challenging areas, such as the bicycle in the first example and the car in the second example. The improved ability of perceiving small objects should be credited to the retrospective refinement on point-level semantic information.

Table 4. Quantitative results on the SemanticKITTI [2] benchmark. We report the mIoU on the validation and test sets.

Method	Val	Test
KPConv [53]	-	58.8
FusionNet [64]	-	61.3
KPRNet [26]	-	63.1
JS3C-Net [60]		66.0
SPVNAS [49]	64.7	66.4
Cylinder3D [69]	-	68.9
RPVNet [58]	-	70.3
$(AF)^2$-S3Net [3]	-	70.8
PVKD [17]	-	71.2
2DPASS [61]	-	72.9
MinkowskiNet [6]	61.9	63.1
MinkowskiNet + Retro-FPN	65.4	68.0

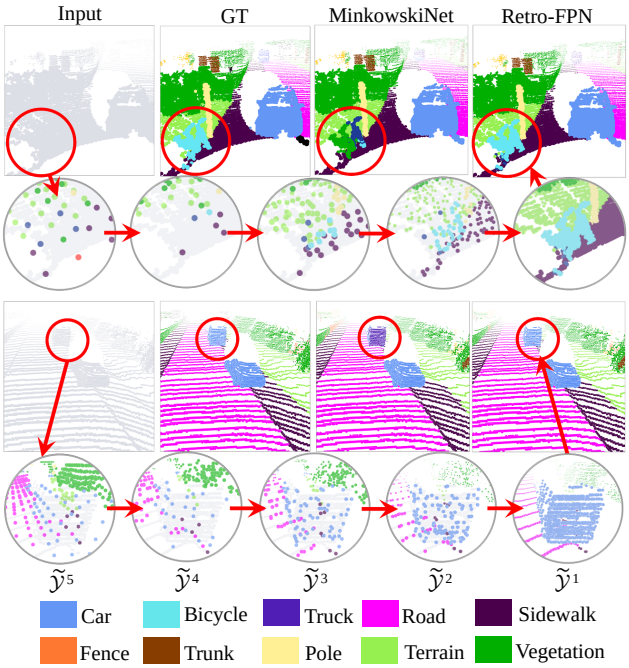


Figure 5. Visualization results of Retro-FPN and the improvements over the backbone networks. The circular areas highlighted in blue visualize the refining process of the improved areas.

5. Model Analysis

In this section, we first provide ablation study regarding each part in Retro-FPN, then we analyze the method in terms of model complexity and run-time efficiency. More model analysis is provided in the supplementary materials.

5.1. Ablation study

We analyze the effect of each part in Retro-FPN in Table 5, where we typically choose Point Transformer [68]

Table 5. Effect of each part in retro-transformer. **HS**: hierarchical supervision **Cross-att**: local cross-attention. **PointEmb**: learnable position embedding. **SemGate**: semantic gate unit.

ID	HS	Cross-att	PosEmb	SemGate	mIoU
I					70.4
II	✓				70.6
III	✓	✓			71.9
IV	✓	✓	✓		72.4
V	✓	✓		✓	72.2
VI		✓	✓	✓	70.8
VII	✓	✓	✓	✓	**73.0**

as the backbone and analyze Reto-FPN on the S3DIS [1] Area 5 benchmark. Note that retro-transformer consists of vanilla cross-attention (Cross-att), learnable position embedding (PosEmb) and semantic gate unit (SemGate).

Effect of explicit refinement. By comparing Exp. II, VI with the baseline I, we show that hierarchical supervision (HS) and retro-transformer is an inseparable integration, neither of them can't take effect alone. Without HS guiding per-point predictions, the retro-transformer still suffers from the ambiguous region features and cannot fully utilize the feature pyramid. Meanwhile, without retro-transformer to refine per-point semantic information, the explicit intermediate features produced by HS cannot facilitate the final prediction. Exp. II and VI can prove the importance of explicit refinement on point-level semantic information.

Effect of retrospective refinement. By comparing Exp. III with the baseline (Exp. I), we can find that the Retro-FPN with the vanilla cross-attention can already improve the backbone by 1.5 in terms of mIoU, which justifies the effectiveness of retrospective refinement.

Effect of retro-transformer. The results of Exp. IV, V and VII indicate that both the learnable position embedding and the semantic gate unit can further improve the refining capacity upon the vanilla cross-attention. Since the local distribution of points may change dramatically, the learnable position embedding can help the local cross-attention to better capture positional relationships. And the semantic gate unit can further screen and control semantic information refinement. Moreover, the combination of PosEmb and SemGate improves mIoU by 1.1 over the vanilla cross-attention, which further validates the design of retro-transformer.

5.2. Model Complexity

We analyze the model complexity of Retro-FPN in Table 6, which is evaluated in terms of parameter number and inference latency. To have a fair comparison, we keep the testing settings the same as backbone networks, where the latency is computed by randomly select a scene/scan and summing the inference time of 100 forward passes. The results in Table 6 show that Retro-FPN leads to negligible extra parameters, ranging from 0.08M to 0.27M. Particularly, for MinkowskiNet on the SemanticKITTI dataset, the increased parameter number (0.08M) is only 0.37% of the backbone network (21.73M). Meanwhile, Retro-FPN leads to consistent computation cost across all backbones, ranging from 0.83s to 2.02s. For lightweight backbones (MinkowskiNet and KPConv), Retro-FPN leads to 20%-30% extra computation overhead. For Point Transformer backbone, Retro-FPN introduces marginal computation cost of 1.11s, which is 2.1% of Point Transformer (54.05s in terms of inference time). In summary, the extra parameters are negligible and the extra computation cost can be effectively controlled. Since Retro-FPN can be conveniently integrated with existing backbones, it provides a valuable trade-off between time and better performance.

Table 6. Run-time model complexity against backbones.

Dataset	Method	Params (M)	Latency (s)	mIoU
S3DIS [1]	MinkowskiNet [6]	15.49	4.44	65.4
	+Retro-FPN	15.57	5.58	69.5
	KPConv *rigid* [53]	24.38	3.81	65.4
	+Retro-FPN	24.65	4.64	69.7
	KPConv *deform*[53]	25.59	4.96	67.1
	+Retro-FPN	25.86	6.32	70.7
	Point Transformer [68]	7.77	54.05	70.4
	+Retro-FPN	7.86	55.16	73.0
ScanNet[9]	MinkowskiNet [6]	15.49	5.83	68.0
	+Retro-FPN	15.57	7.59	70.8
SemanticKITTI[2]	MinkowskiNet [6]	21.73	6.82	63.1
	+Retro-FPN	21.81	8.84	68.0

6. Conclusions and Limitations

We present Retro-FPN to improve per-point semantic feature prediction for 3D point clouds, which can fully exploit the feature pyramid and models the feature propagation as an explicit and retrospective refining process on point-level semantic information. By further introducing a retro-transformer in each pyramid layer, Retro-FPN can effectively extract and refine semantic information from all pyramid levels to the final prediction layer. We integrate Retro-FPN with three prevailing backbones and conduct experiments on widely used benchmarks. Experimental results demonstrate that Retro-FPN can significantly improve segmentation performance over state-of-the-art methods.

The primary limitation of Retro-FPN is that the retro-transformer relies on the K-NN search to capture local semantic contexts. Since the point distribution of point clouds may vary dramatically in different local regions, a fixed number of nearest neighbors may fail to provide informative contextual information for refinement, especially in dense and complex areas. Meanwhile, a large number of K-NN search will also lead to more computation cost. Therefore, a promising future direction is explore flexible neighbor searching strategy, in order to capture more accurate semantic contexts and further bring down computation cost.

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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1534–1543, 2016. 2, 6, 7, 8

[2] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019. 2, 6, 7, 8

[3] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12547–12556, June 2021. 7

[4] Hung-Yueh Chiang, Yen-Liang Lin, Yueh-Cheng Liu, and Winston H Hsu. A unified point-based framework for 3D segmentation. In *2019 International Conference on 3D Vision (3DV)*, pages 155–163. IEEE, 2019. 1

[5] Jaesung Choe, Chunghyun Park, Francois Rameau, Jaesik Park, and In So Kweon. PointMixer: Mlp-mixer for point cloud understanding. In *European Conference on Computer Vision*, pages 620–640. Springer, 2022. 6

[6] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 3, 6, 7, 8

[7] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 5

[8] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds for autonomous driving, 2020. 1

[9] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 2, 6, 7, 8

[10] Angela Dai and Matthias Niessner. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 3

[11] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14504–14513, June 2021. 6

[12] Naiyu Gao, Yanhu Shan, Yupei Wang, Xin Zhao, Yinan Yu, Ming Yang, and Kaiqi Huang. SSAP: Single-shot instance segmentation with affinity pyramid. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 3

[13] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V. Le. NAS-FPN: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[14] Jingyu Gong, Jiachen Xu, Xin Tan, Haichuan Song, Yanyun Qu, Yuan Xie, and Lizhuang Ma. Omni-supervised point cloud segmentation via gradual receptive field component reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11673–11682, 2021. 2, 3, 4, 6

[15] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3, 7

[16] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. PCT: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, Apr 2021. 3

[17] Yuenan Hou, Xinge Zhu, Yuexin Ma, Chen Change Loy, and Yikang Li. Point-to-voxel knowledge distillation for lidar semantic segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8469–8478, 2022. 7

[18] Miao Hu, Yali Li, Lu Fang, and Shengjin Wang. A2-FPN: Attention aggregation based feature pyramid network for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15343–15352, June 2021. 3

[19] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11108–11117, 2020. 1

[20] Wenbo Hu, Hengshuang Zhao, Li Jiang, Jiaya Jia, and Tien-Tsin Wong. Bidirectional projection network for cross dimension scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14373–14382, 2021. 6, 7

[21] Zeyu Hu, Xuyang Bai, Jiaxiang Shang, Runze Zhang, Jiayu Dong, Xin Wang, Guangyuan Sun, Hongbo Fu, and Chiew-Lan Tai. VMNet: Voxel-mesh network for geodesic-aware 3d semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15488–15498, October 2021. 7

[22] Zeyu Hu, Mingmin Zhen, Xuyang Bai, Hongbo Fu, and Chiew-lan Tai. Jsenet: Joint semantic segmentation and edge detection network for 3D point clouds. In *European Conference on Computer Vision*, pages 222–239. Springer, 2020. 7

[23] Qiangui Huang, Weiyue Wang, and Ulrich Neumann. Recurrent slice networks for 3D segmentation of point clouds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2626–2635, 2018. 3

[24] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings*

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6399–6408, 2019. 1, 3

[25] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 3, 4

[26] Deyvid Kochanov, Fatemeh Karimi Nejadasl, and Olaf Booij. KPRNet: Improving projection-based lidar semantic segmentation. *arXiv preprint arXiv:2007.12668*, 2020. 7

[27] Xin Lai, Jianhui Liu, Li Jiang, Hengshuang Zhao Liwei Wang, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3D point cloud segmentation. In *CVPR*, 2022. 3, 6, 7

[28] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[29] Huan Lei, Naveed Akhtar, and Ajmal Mian. SegGCN: Efficient 3D point cloud segmentation with fuzzy spherical kernel. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1

[30] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 3

[31] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[32] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 1, 3

[33] Yiqun Lin, Zizheng Yan, Haibin Huang, Dong Du, Ligang Liu, Shuguang Cui, and Xiaoguang Han. FPConv: Learning local flattening for point convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 6

[34] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018. 3

[35] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[36] Alexey Nekrasov, Jonas Schult, Or Litany, Bastian Leibe, and Francis Engelmann. Mix3D: Out-of-context data augmentation for 3d scenes. In *2021 International Conference on 3D Vision (3DV)*, pages 116–125. IEEE, 2021. 4, 7

[37] Dong Nie, Rui Lan, Ling Wang, and Xiaofeng Ren. Pyramid architecture for multi-scale processing in point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17284–17294, June 2022. 3

[38] Xuran Pan, Zhuofan Xia, Shiji Song, Li Erran Li, and Gao Huang. 3D object detection with Pointformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7463–7472, June 2021. 3

[39] Chunghyun Park, Yoonwoo Jeong, Minsu Cho, and Jaesik Park. Fast Point Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3, 6

[40] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019. 6

[41] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[42] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5099–5108, 2017. 3, 6

[43] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. PointNeXt: Revisiting pointnet++ with improved training and scaling strategies. In *Advances in Neural Information Processing Systems*, 2022. 6

[44] Haibo Qiu, Baosheng Yu, and Dacheng Tao. GFNet: Geometric flow network for 3D point cloud semantic segmentation. *Transactions on Machine Learning Research*, 2022. 1

[45] Shi Qiu, Saeed Anwar, and Nick Barnes. Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1757–1767, June 2021. 3

[46] Haoxi Ran, Jun Liu, and Chengjie Wang. Surface representation for point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18942–18952, June 2022. 6

[47] Damien Robert, Bruno Vallet, and Loic Landrieu. Learning multi-view aggregation in the wild for large-scale 3D semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 6

[48] Mingxing Tan, Ruoming Pang, and Quoc V. Le. EfficientDet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[49] 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*, 2020. 6, 7

[50] Liyao Tang, Yibing Zhan, Zhe Chen, Baosheng Yu, and Dacheng Tao. Contrastive boundary learning for point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8489–8499, June 2022. 6

[51] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3D. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[52] Lyne P. Tchapmi, Christopher B. Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. SEGCloud: Semantic segmentation of 3D point clouds. In *International Conference on 3D Vision (3DV)*, 2017. 3, 6

[53] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. KPConv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420, 2019. 2, 3, 6, 7, 8

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 3, 5

[55] Wenxuan Wu, Zhongang Qi, and Li Fuxin. PointConv: Deep convolutional networks on 3D point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 3

[56] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. In *NeurIPS*, 2022. 3, 6, 7

[57] 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*, pages 1–19. Springer, 2020. 1

[58] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16024–16033, October 2021. 7

[59] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan 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 (CVPR)*, pages 3173–3182, June 2021. 3, 6

[60] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3101–3109, 2021. 7

[61] Xu Yan, Jiantao Gao, Chaoda Zheng, Chao Zheng, Ruimao Zhang, Shuguang Cui, and Zhen Li. 2DPASS: 2D priors assisted semantic segmentation on lidar point clouds. In *ECCV*, 2022. 7

[62] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang 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*, pages 5589–5598, 2020. 1

[63] Zetong Yang, Li Jiang, Yanan Sun, Bernt Schiele, and Jiaya Jia. A unified query-based paradigm for point cloud understanding. In *CVPR*, 2022. 6, 7

[64] Feihu Zhang, Jin Fang, Benjamin Wah, and Philip Torr. Deep fusionnet for point cloud semantic segmentation. In *European Conference on Computer Vision*, pages 644–663. Springer, 2020. 7

[65] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1

[66] Zhiyuan Zhang, Binh-Son Hua, and Sai-Kit Yeung. Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1607–1616, 2019. 3

[67] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. Pointweb: Enhancing local neighborhood features for point cloud processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[68] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. 2, 3, 5, 6, 7, 8

[69] 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 Conference on Computer Vision and Pattern Recognition*, pages 9939–9948, 2021. 7