Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Semantic segmentation of 3D indoor LiDAR point clouds through feature pyramid architecture search

Haojia Lin [a], Shangbin Wu [a], Yiping Chen [a,*], Wen Li [a], Zhipeng Luo [a], Yulan Guo [b], Cheng Wang [a], Jonathan Li [a,c,*]

[a] Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University, Xiamen, China
[b] College of Electronic Science and Technology, National University of Defense Technology, Changsha, China
[c] Department of Geography and Environmental Management, University of Waterloo, Waterloo, Canada

A R T I C L E I N F O

Keywords:
Semantic segmentation
Point clouds
3D vision
Indoor scenes
Neural architecture search
Feature pyramid network

A B S T R A C T

Semantic segmentation of 3D Light Detection and Ranging (LiDAR) indoor point clouds using deep learning has been an active topic in recent years. However, most deep neural networks on point clouds conduct multi-level feature fusion via a simple U-shape architecture, which lacks enough capacity on both classification and localization in the segmentation task. In this paper, we propose a Neural Architecture Search (NAS) method to search a Feature Pyramid Network (FPN) module for 3D indoor point cloud semantic segmentation. Specifically, we aim to automatically find an effective feature pyramid architecture as a feature fusion neck in a designed novel pyramidal search space covering all information communication paths for multi-level features. The searched FPN module, named SFPN, contains the most important connections among all the potential paths to fuse representations at different levels. Our proposed SFPN is generic and effective as well as capable to be added to existing segmentation networks to augment the segmentation performance. Extensive experiments on ScanNet and S3DIS show that consistent and remarkable gains of segmentation performance can be achieved by different classical networks combined with SFPN. Specially, PointNet++-SFPN achieves mIoU gains of 7.8% on ScanNet v2 and 4.7% on S3DIS, and PointConv-SFPN achieves 4.5% and 3.7% improvement respectively on the above datasets.

1. Introduction

With the rapid development of inexpensive indoor 3D LiDAR sensors and acquisition techniques, point clouds have become widely available, which sparks research interests in 3D scene understanding. As a challenge task, point cloud semantic segmentation aims to classify each point in the point set. It widely benefits a lot of real-world applications, such as autonomous navigation, virtual reality, robot object manipulation and high-quality indoor mapping.

Several deep convolution neural networks have been undertaken for point cloud segmentation, such as the pioneer work, PointNet (Qi et al., 2017a) and many other improved approaches, e.g., Qi et al. (2017b), Li et al. (2018b), Wu et al. (2019), Yan et al. (2020), Lin et al. (2020), Lin et al. (2020), Fang and Lafarge (2019). Generally, an effective semantic segmentation system consists of two main components (Li et al., 2019): a feature extractor backbone and a feature fusion neck. Most existing works develop effective feature extractor backbones (Li et al., 2018b; Wu et al., 2019; Yan et al., 2020) while utilizing a simple U-shape feature fusion neck (which is composed of skip connection and interpolation) to fuse multi-level features. Recently, a growing interest in developing more effective feature fusion neck (Fang and Lafarge, 2019; Lin et al., 2020) is emerging. We follow this trend and aim to discover an effective and general feature pyramid architecture for feature fusion neck.

Feature Pyramid Network (FPN) is widely used in complicated image analysis tasks. Lin et al. (2017) firstly propose FPN to produce pyramidal feature representation for image object detection. A FPN first takes over the features generated at the intermediate layers in the backbone network, then builds a feature pyramid by combining the features in adjacent levels. As a result, the output features of FPN are spatially accurate and semantically rich. Due to the effectiveness and simplicity of FPN, several variants have been presented by proposing various cross-scales connections and operations for multi-level feature fusion, such as Liu et al. (2018), Li et al. (2018a), Zhang et al. (2020a). APCF-Net (Lin

* Corresponding authors at: Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University, Xiamen, China (Y. Chen).
E-mail addresses: chenyiping@xmu.edu.cn (Y. Chen), junli@uwaterloo.ca (J. Li).

https://doi.org/10.1016/j.isprsjprs.2021.05.009
Received 24 February 2021; Received in revised form 12 May 2021; Accepted 15 May 2021

et al., 2020) introduces the feature pyramid architecture into point cloud segmentation and proposes a bi-direction communication mechanism to exploit multi-scale context. Although the APCF module enriches considerably the learned features, it may be far away from the optimal FPN architecture. Therefore, we introduce Neural Architecture Search (NAS) to discover better feature pyramid architectures for point cloud segmentation.

Recently, there has been a increasing popularity in automatically discovering neural network architectures instead of manual design which largely relies on human experience and massive trials. NAS have shown promising capacity of discovering architectures that outperform manually designed ones in image classification (Zoph et al., 2018; Liu et al., 2019), image segmentation (Chen et al., 2018; Zhang et al., 2019) and 2D object detection (Xu et al., 2019). Recent work SPVNAS (Tang et al., 2020) performs NAS for point cloud segmentation. However, SPVNAS focuses on searching the backbone network while the architecture of feature fusion neck is ignored. This lack makes the pipeline of SPVNAS alike with the NAS method on classification task, i.e., ignoring the demand of localization information for point cloud segmentation. Contrarily, our method performs NAS on the feature fusion neck part and aims to explore effective feature pyramid architecture. Hence, it is oriented for the segmentation task.

In this paper, we propose a NAS method to search an effective feature pyramid architecture for point clouds semantic segmentation. First, we design a novel pyramidal search space covering all possible communication path for features at different levels in the backbone. Second, inspired by the differentiable NAS methods (Liu et al., 2019; Zhang et al., 2019), we reformulate the designed search space in a continuous relaxation form. The architecture search is mostly achieved via Stochastic Gradient Descent (SGD). The cost of search is very low and takes only 3 h on one 2080Ti GPU for the ScanNet v2 dataset (Dai et al., 2017) and 4 h for the S3DIS dataset (Armeni et al., 2016), which is relatively close to a standard training for the entire segmentation model. As Fang and Lafarge (2019) proposed, instead of achieving state-of-the-art performances, our goal is to propose an architecture search method that can discover a generic feature fusion module to augment the point-wise representation. The proposed searched feature pyramid network, named SFPN, works well with different backbone networks, which shows its effectiveness and generality. The main contributions of our method are summarized as follows:

- Different from conventionally designing the neural network in a manual manner, we propose a network architecture search method for point cloud semantic segmentation. This automation on neural architecture design save massive trial and error by human.
- We design a novel pyramidal search space covering all information communication paths for multi-level features. Besides, the SFPN discovered from this space consists of the most important connections among all the potential ones, thus is able to conduct effective information communication.
- Extensive experiments on ScanNet v2 and S3DIS show the great effectiveness and generality of SFPN. The FPN discovered by our architecture search method can work as a plug-and-play module, which is able to be integrated with different backbones and achieve consistent and significant segmentation performance improvements.

The rest of this paper is organized as follows: Section 2 reviews the related work. Section 3 introduces the proposed method. Section 4 presents and discusses the experimental results. Section 5 concludes this paper as well as limitation and future work.

2. Related work

3D semantic segmentation based on deep learning. PointNet (Qi et al., 2017a) is considered as a pioneer in point set deep learning. By applying a shared MLP on each point individually, it learns pointwise representation for the input point set. Inspired by PointNet, recent works (Li et al., 2018b; Qi et al., 2017b; Wu et al., 2019; Yan et al., 2020; Zhao et al., 2019; Wang et al., 2019; Li et al., 2019) devote to directly processing point cloud. Compared with its projection-based (Su et al., 2015; Boulch et al., 2018) and voxel-based (Maturana and Scherer, 2015; Qi et al., 2016) counterparts, the point-based methods not only avoid heavy computation and expensive memory cost, but also bypass the deleterious quantization error and artifacts, thus make it the mainstream today. In PointNet++ (Qi et al., 2017b), local mini-PointNet is alternated with farthest points sampling to capture the local dependency for each point. Several works focus on improving the subsampling methods. To conduct representative subsampling, PAT (Yang et al., 2019) proposes an end-to-end learnable samping operation. PointASNL (Yan et al., 2020) uses the nonlocal (Wang et al., 2018b) operation in a local manner to alleviate the biased effect caused by the outlier points. Additionally, some works borrow the idea of message passing neural networks (Justin et al., 2017) to exploit local context effectively. LocSpec (Wang et al., 2018a) constructs the nearest neighbour graphs for each point and conducts spectral graph convolution on these graphs. PointWeb (Zhao et al., 2019) presents a local densely connected graph for input points to exploit the interaction between points. By leveraging the efficiency of grid space, Grid-GCN (Xu et al., 2020) constructs the graph efficiently with higher coverage. Moreover, some works explore convolution operation specific for point clouds. PointCNN (Li et al., 2018b) utilizes the learned X-transformation to transforms the points into a canonical space and then emploies standard convolution to exploit local dependency. PointConv (Wu et al., 2019) uses multi-layer perceptrons as the weight function to generate convolution weights and reformulate this convolution in an efficient manner. By learning a local flattening projection, FPConv (Lin et al., 2020) defines the convolution on surface geometry. All these methods focus on developing local feature extractors to construct a more effective backbone network but ignore the feature fusion for the features at different levels in the backbone. Recently, a growing interest in exploring more effective feature fusion neck (Fang and Lafarge, 2019; Lin et al., 2020) is emerging. A 3D pyramid pooling module is proposed by 3D-PSPNet (Fang and Lafarge, 2019) to enrich pointwise features with multi-scale context. APCF-Net (Lin et al., 2020) presents an adaptive communication mechanism for multi-level features to capture the scales of contextual information. We follow this trend and aim to discover an effective and general feature pyramid architecture for feature fusion.

Feature Pyramid Network. FPN is widely employed in many image analysis tasks that require multi-scale processing. Lin et al. (2017) firstly propose FPN to generate pyramidal feature representation for image object detection to recognize small pattern more effectively. Using the lateral and top-down connections across adjacent levels, FPN propagates the rich semantic information from high-level features to low-level features, which produces feature representations that are both semantically strong and spatially accurate. Many FPN variants have been proposed to further strengthen multi-level feature fusion. To promote the propagation of localization signals in low levels, PANet (Liu et al., 2018) adds an extra bottom-up pathway on FPN. Li et al. (2018a) introduces an attention mechanism into FPN to achieve feature fusion under the guidance of rich semantic context in high-level features. In recent years, there have been many scientific works propose various cross-level pathways or modules to produce pyramidal feature representation on object detection (Kim et al., 2018; Yu et al., 2018), semantic segmentation (Zhang et al., 2020a) and pose estimation (Jiang et al., 2020). APCF-Net (Lin et al., 2020) first introduces the feature pyramid architecture into point cloud segmentation. Inspired by this work, we further introduce NAS to explore a more effective and generic FPN with better communication mechanism for multi-level context.

Neural Architecture Search. NAS devotes to automatically discovering top-performing neural network architecture for a certain task. According to (Elsken et al., 2019), a NAS method can be abstracted into three parts: search space, search strategy and performance

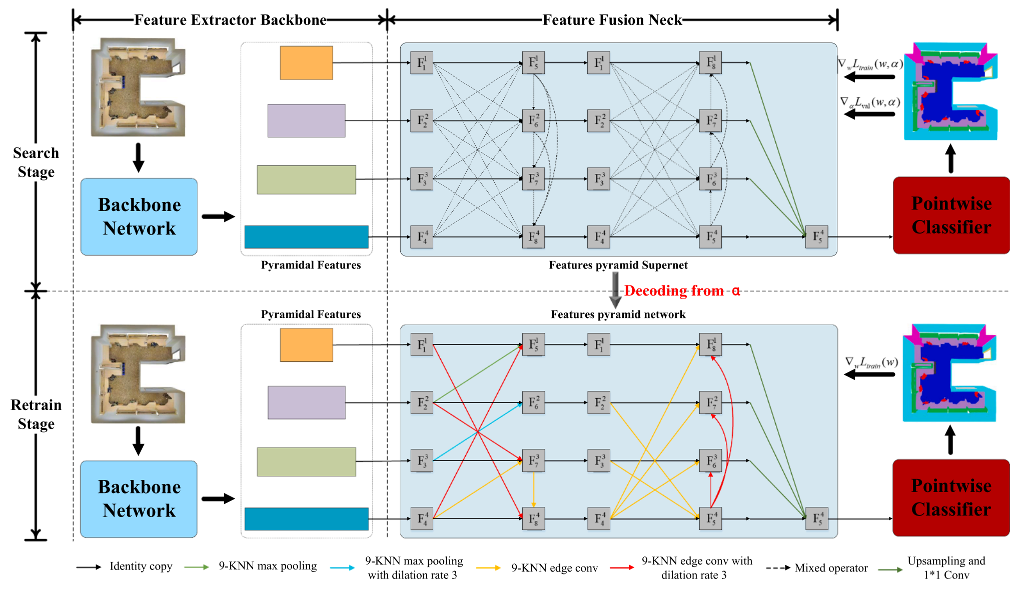


Fig. 1. Framework of the proposed method.

estimation strategy. The search space defines which architectures can be discovered in principle. Prior knowledge about adaptation to the specific task can reduce the size of search space and accelerate the search process. The search strategy details how to select a good architecture in the predefined search space according to the estimated performance returned by the performance estimation strategy. According to the search strategy adopted by a NAS method, NAS can be mainly classified into three categories: 1) Evolutionary Algorithms (EA) based methods (Real et al., 2019; Guo et al., 2020), which evolves the architecture by iterative mutation on the current best architectures. 2) Reinforcement learning based methods (Zoph et al., 2018) utilize a RNN policy controller to produce the actions that specify the architecture. 3) Gradient based methods (Liu et al., 2019; Xu et al., 2019) apply a continuous relaxation on the discrete search space, thus the SGD can be used to conduct the efficient optimization for the search. Most of these proposed methods perform to search on image classification, only a few works focus on more challenging vision task such as semantic segmentation (Chen et al., 2018; Xu et al., 2019; Zhang et al., 2019). Recently, SPVNAS (Tang et al., 2020) propose an efficient 3D perception operator and use NAS to search the optimal architecture for the proposed operator. Different from SPVNAS which focuses on searching the backbone architecture, our method performs NAS on the feature fusion part and aims to explore effective feature pyramid architecture thus is more task-oriented for the segmentation task.

3. Method

Fig. 1 shows the framework of the proposed method. This framework can be divided into two stages: search stage and retrain stage. In each stage, the pipeline of the segmentation system consists of two main components: the feature extractor backbone and the feature fusion neck. The backbone network absorb input point cloud and produce multi-level features, then the feature fusion neck takes over these features and conduct feature fusion for better segmentation. In this paper, we use existing networks as the backbone and explore feature fusion neck using automatical architecture search. The objective is to find an effective FPN architecture to play the role of fusion neck. In the search stage, we conduct the feature pyramid architecture search to discover an effective FPN architecture. In the retrain stage, we integrate the searched FPN with the backbone network and retrain the entire segmentation network.

The retrain stage follows a standard training procedure which is common in existing deep learning approaches. Therefore, here we elaborate the search stage. Firstly, we design the search space in which we search the FPN architecture (Section 3.1). Then, we apply the continuous relaxation on the proposed discrete search space to generate a feature pyramid supernet (Section 3.2). Finally, the supernet is optimized during which the better architectures are emerging over the time (Section 3.3).

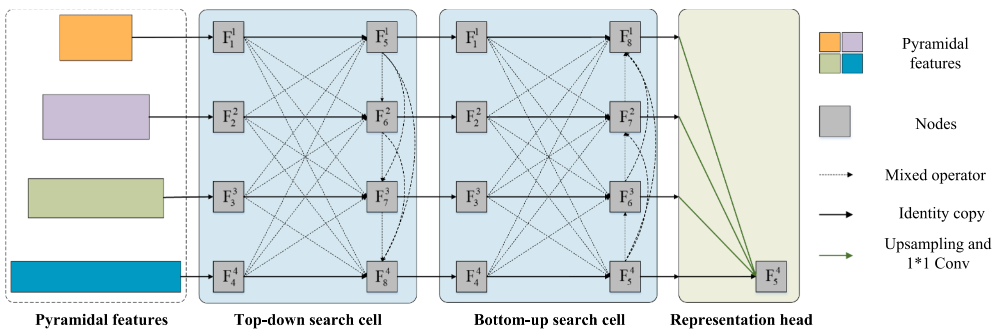


Fig. 2. The designed FPN search space.

3.1. Search space

The search space defines which architectures can be discovered in principle. Previous studies (Radosavovic et al., 2019); Yang et al. (2020) have indicated that the search space plays a significant role in the success of NAS. Therefore, we design a search space that covers all information communication paths across different levels. The communication paths in the designed search space can be formulated as a directed acyclic graph (DAG) (Section 3.1.1). Furthermore, we adopt a bidirectional search strategy (Section 3.1.2) to strengthen the effectiveness of communication. As a result, this DAG can be divided into two parts: the top-down cell and the bottom-up cell. To strengthen the supervision for features at each level, we use a representation head (Section 3.1.3) to rescale and integrate all the features in the pyramid into a final fusion. Fig. 2 shows the designed FPN search space. The stripes with four colors denote the pyramidal features. Different arrows represent different operations in the search space.

3.1.1. Information communication DAG

We take the top-down cell for example to illustrate the communication path configuration in our designed search space. This configuration has a symmetric fashion in the bottom-up cell.

Nodes. The top-down cell can be abstracted as a DAG composed of an ordered sequence of nodes $\{F_i^r | 1 \leqslant i \leqslant 8,\ 1 \leqslant r \leqslant 4\}$, where i denotes the order index and r denotes the resolution level. These nodes can be cut into two subsets: the input nodes $\{F_1^1, F_2^2, F_3^3, F_4^4\}$ and the intermediate nodes $\{F_5^1, F_6^2, F_7^3, F_8^4\}$. The input nodes are the features generated at different levels by the backbone while the intermediate nodes are the acquired features in this top-down fusion feature pyramid. From F_1^1 to F_4^4 and F_5^1 to F_8^4, the resolution is gradually subsampled.

Edges. Each directed edge (i,j) between node $F_i^{r_1}$ and node $F_j^{r_2}$ is associated with some operation $O^{(i,j)} \in \mathscr{O}_{\mathscr{N}}$ that transforms $F_i^{r_1}$. The set of possible operations includes the following operators:

- no connection (none)
- 9 K N N max pooling
- 9 K N N dilated max pooling with dilation rate 3
- 9 K N N edg e conv olution
- 9 K N N dilated edg e conv olution with dilation rate 3

Considering that the spatial-awareness and receptiveness field are relatively important for the feature fusion involving multiple levels, dilated operators are introduced to our search space.

Communication paths. Under the setting about the edges and nodes described above, the information communication paths can be formulated as:

$$F_j^{r_2} = \sum_{i<j} S\left(O^{(i,j)}\left(F_i^{r_1}\right)\right) \tag{1}$$

where $5 \leqslant j \leqslant 8$. Each intermediate node is computed based on all of its predecessors. An operator $O^{(i,j)}(\circ)$ is first applied on each predecessor node of the intermediate node $F_j^{r_2}$. Then, an upsamping or down-sampling operator $S(\circ)$ will be utilized to rescale the nodes into the corresponding target resolution at level r_2. Finally, all the transformed predecessors are summed to determine the target intermediate node $F_j^{r_2}$. Note that, once the target intermediate node share the same resolution as its predecessor node, it is not necessary to apply any nonlinear feature transformation and sampling operator. Therefore, we set $O^{(i,j)}(\circ)$ to be an simple identity copy operator and cancel $S(\circ)$ under this circumstance.

3.1.2. Bidirection search

To further strengthen the effectiveness of multi-level information communication, we follow a bidirection search strategy to design our FPN search space. The search space consists of a top-down cell and a bottom-up cell. The bottom-up cell take the output of the top-down cell as input. The only difference between them is the level configuration of the intermediate nodes. Different from the top-down cell described in Section 3.1.1, the bottom-up cell place its multi-level intermediate nodes in an reverse order, i.e., $\{F_5^4, F_6^3, F_7^2, F_8^1\}$. Fig. 2 illustrates this difference.

3.1.3. Representation head

After adequate information communication across different levels, a mixed feature pyramid which contains both semantically rich and spatially accurate representation can be obtained. Before inputting to the final point-wise classifier, this feature pyramid should be squeezed into point-wise representations. The spatial size and the number of channels of high-level features are rescaled to the lowest level. Then all these features are concatenated together. Different from the U-Net framework employed by previous works (Qi et al., 2017b; Yang et al., 2021); Wu et al. (2019) which only utilize the highest-resolution feature, we push the features at all levels to the final classifier, which strengthen the supervision for the FPN.

3.2. Continuous relaxation

Architecture parameters. Following DARTS (Liu et al., 2019), we use the continuous relaxation by introducing an architecture parameter $\alpha \in R^{|\xi| \times |\mathscr{O}_{\mathscr{N}}|}$, where $|\mathscr{O}_{\mathscr{N}}|$ refers to the number of possible operators and $|\xi|$ denotes the number of edges. The categorical choice of a particular operation is relaxed to a softmax distribution $\alpha = \left\{\alpha_O^{(i,j)} \middle| (i,j) \in \xi, O \in \mathscr{O}_{\mathscr{N}}\right\}$ over all the possible operators:

$$\overline{O}^{(i,j)}\left(x\right) = \sum_{O \in \mathscr{O}_{\mathscr{N}}} \frac{\exp\left(\alpha_O^{(i,j)}\right)}{\sum_{O' \in \mathscr{O}_{\mathscr{N}}} \exp\left(\alpha_{O'}^{(i,j)}\right)} O\left(x\right) \tag{2}$$

Algorithm 1. optimization for the FPN search

```
for edge $(i,j)$ in $\xi$ do
    Create a mixed operation $\overline{O}^{(i,j)}$ parameterized by initializing $\alpha^{(i,j)}$ for each edge $(i,j)$
while not converged do
    1. Fixing architecture $\alpha$, update weight $w$ by descending $\nabla_w \mathscr{L}_{\text{train}}(w, \alpha)$.
    2. Fixing weight $w$, update architecture $\alpha$ by descending $\nabla_\alpha \mathscr{L}_{\text{val}}(w, \alpha)$.
    Decode the discrete architecture from the learned $\alpha$.
```

Therefore, the architecture search is reformulated as the learning for α. Furthermore, together with the network parameter w, the designed search space in Section 3.1 can be instantiated as a feature pyramid supernet $SN(\alpha, w)$. Consequently, the search process in the search space can be converted into the optimization for this supernet.

Decoding architecture from α. After the search, a discrete architecture can be derived from the architecture parameter. For each edge (i, j), the mixed operation $\overline{O}^{(i,j)}(x)$ is replaced with the most likely operator, i.e., $O^{(i,j)} = argmax_{O \in \mathscr{O}_{\mathscr{N}}} \alpha_O^{(i,j)}$. For each intermediate node, K edges with the top intensity are retained in the final architecture. We use $K = 2$ due to the trade-off between effectiveness and efficiency.

3.3. Optimization

Using continuous relaxation, the architecture search can be reformulated as the joint optimization for the entire network parameter w and the FPN architecture parameter α. As mentioned at the beginning of Section 3, the objective is to find an effective FPN architecture to play the role of fusion neck. Therefore, the FPN architecture parameter α only involves the fusion neck. Note that we do not use a pre-trained backbone network in the search process to keep an end-to-end style. As a result, w involves the weight parameters of all the three components of the entire network including the backbone network, the feature fusion neck and

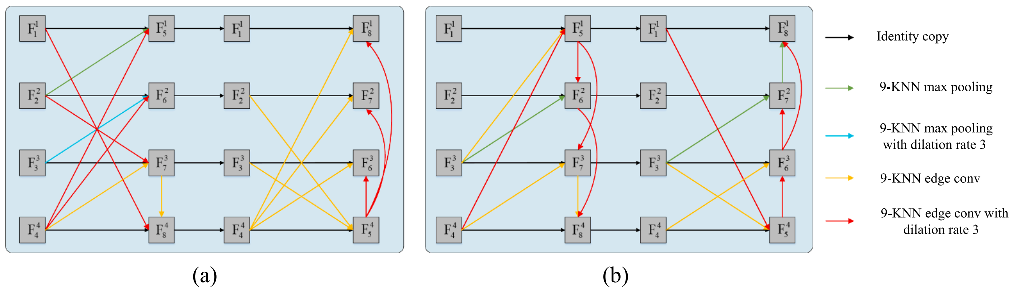

(a)
(b)
Identity copy
9-KNN max pooling
9-KNN max pooling with dilation rate 3
9-KNN edge conv
9-KNN edge conv with dilation rate 3

Fig. 3. Searched feature pyramids architecture on ScanNet v2 (a) and S3DIS (b).

the pointwise classifier. The training data is divided into two disjoint parts: a training set and a validation set. The optimization for w and α are alternated between these two data sets until convergence: to optimize w, we fix the architecture α and update network parameter w by $\nabla_w \mathscr{L}_{\text{train}}(w, \alpha)$; to optimize α, we fix the network parameter w and update architecture parameter α by $\nabla_\alpha \mathscr{L}_{\text{val}}(w, \alpha)$. The iterative optimization procedure is outlined in Alg.1.

3.4. Evaluation metrics

Overall Accuracy (OA) and mean Interaction-over-Union (mIoU) are as the major evaluation metrics. OA is the percentage of correctly predicted points in all points, while mIoU is defined as:

$$mIoU = \frac{\sum_{i=1}^{N} IoU_i}{N} \tag{3}$$

$$IoU_i = \frac{p_{ii}}{p_{ii} + \sum_{j \neq i} p_{ij} + \sum_{k \neq i} p_{ki}} \tag{4}$$

where p_{ii} is class i predicted to be the correct class i,p_{ij} denotes the number of points belong to class i but are misclassified as class j,p_{ki} denotes the number of points belong to class k but are misclassified as class i, and N is the total number of classes. IoU_i denotes Interaction-over-Union of the class i.

4. Results and discussion

In this section, we conducted extensive experiments to evaluate our proposed method. Section 4.1 introduced the experimental settings. Section 4.1 and Section 4.2 presented the segmentation results on ScanNet v2 and S3DIS, respectively. Section 4.4 analyzed the architecture search.

4.1. Experiment settings

Datasets. We conducted extensive experiments on ScanNet v2 (Dai et al., 2017) and S3DIS (Armeni et al., 2016). The ScanNet dataset includes 1513 scans for training and 100 scans for testing. The number of the category is 21. The labels of the test dataset is only available in the official evaluation server, therefore we submitted the prediction on test split to the benchmark website for comparison with other methods. The S3DIS dataset contains 13 classes. It contains 6 large-scale point clouds of 271 rooms, which are collected from 3 different buildings.

Implementation details. Note that our goal is not to achieve state-of-the-art performance, but to improve the segmentation performance of existing network. Therefore, in this paper, we use PointNet++ and PointConv as the baseline networks to evaluate our SFPN. Specifically, we fix all the hyperparameters and experiments protocol while only compare the results with and without using SFPN. In the comparison with PointConv, we replace the origin enormous decoder with SFPN. Besides a significant segmentation improvement, this replacement leads to a much higher efficiency.

Table 1
Comparison with existing methods on ScanNet v2 dataset in mIoU(%).

Method	mIoU
PointCNN (Li et al., 2018b)	45.8
MS-PCNN (Ma et al., 2021)	56.8
PointConv-CE (Liu et al., 2020)	60.9
HPEIN (Jiang et al., 2019)	61.8
FusionAwareConv (Zhang et al., 2020b)	63.0
PointASNL (Yan et al., 2020)	63.0
APCF-Net (Lin et al., 2020)	63.1
PointNet++ (Qi et al., 2017b)	55.3
PointNet++-SFPN (**Ours**)	**63.1**
	($\uparrow$ 7.8)
PointConv (Wu et al., 2019)	59.4
PointConv-SFPN (**Ours**)	**64.1**
	($\uparrow$ 4.7)

4.2. Semantic segmentation on the ScanNet v2 Dataset

The ScanNet v2 dataset (Dai et al., 2017) provides 1513 3D indoor scene scans, which can be splited into 1201 scans to form the training set and 312 scans to form the validation set. We follow the experiment protocol adopted by FPConv (Lin et al., 2020). To preprocess the training data, we randomly sample $2m \times 2m \times 3m$ cubes with 8192 points. While for validation or testing, all points are covered. We use 3D coordinates without RGB as the input of our model. As described in Section 3.3, during the architecture search, the training data and validation data are respectively used for updating the model parameters and the achitecture parameters. In the search stage, to reduce the computational cost, we use PointNet++ as the backbone for our entire segmentation model. The number of training epochs for the search is 40. Fig. 3 depicted that the searched feature pyramids architecture on two test datasets. The color of black, blue, yellow, red and green are meaning different operations of identity copy, 9-knn max pooling, 9-KNN max pooling with dilation rate 3, 9-knn edge con and 9-knn edge conv with dilation rate 3, respectively. Fig. 3 (a) shows the searched feature pyramid architecture on ScanNet v2. In the training stage, we respectively use the PointNet++ and PointConv as the backbone to demonstrate the generality and effectiveness of the FPN architecture obtained in the search stage. We train the whole model (i.e., PointNet++/PointConv plugged with the searched FPN) for 300 epochs. The evaluation are conducted on the online test dataset.

Quantitative analysis. Two networks are integrated with the searched FPN to compare with several existing methods including PointNet++, PointConv, PointCNN, MS-PCNN (Ma et al., 2021), PointConv-CE, HPEIN (Jiang et al., 2019), FusionAwareConv (Zhang et al., 2020b), PointASNL (Yan et al., 2020), and APCF-Net. To achieve a fair comparison, we train the PointNet++ and PointConv as the baseline

Table 2
Segmentation results on the ScanNet v2 dataset in per-class IoU (%).

Method	bathtub	bed	bookshelf	cabinet	chair	counter	curtain	desk	door	floor
PointNet++	63.3	64.8	65.9	43.0	80.0	39.0	59.2	45.4	37.1	93.9
PointNet++-SFPN	77.1	69.2	67.2	52.4	83.7	44.0	70.6	53.8	44.6	94.4
(Ours)	(↑ 13.8)	(↑ 4.4)	(↑ 1.3)	(↑ 8.6)	(↑ 3.7)	(↑ 5.0)	(↑ 11.4)	(↑ 8.4)	(↑ 7.5)	(↑ 0.5)
PointConv	74.4	65.1	65.1	51.7	78.0	41.9	58.7	45.0	39.3	94.9
PointConv-SFPN	77.6	70.3	72.1	55.7	82.6	45.1	67.2	56.3	48.3	94.3
(Ours)	(↑ 3.2)	(↑ 5.2)	(↑ 7.0)	(↑ 6.0)	(↑ 4.6)	(↑ 3.2)	(↑ 8.5)	(↑ 11.3)	(↑ 9.0)	(↓ 0.6)
Method	otherfurniture	picture	refrigerator	shower	sink	sofa	table	toilet	wall	window
PointNet++	36.8	13.6	36.8	44.8	56.0	71.5	48.6	88.2	72.0	46.2
PointNet++-SFPN	42.1	21.9	55.2	75.1	59.1	73.7	54.3	90.1	76.8	55.7
(Ours)	(↑ 5.3)	(↑ 8.3)	(↑ 18.4)	(↑ 20.3)	(↑ 3.1)	(↑ 2.2)	(↑ 5.7)	(↑ 1.9)	(↑ 4.8)	(↑ 9.5)
PointConv	37.0	17.1	54.0	69.1	65.9	67.4	49.0	87.2	76.7	50.6
PointConv-SFPN	42.5	16.2	64.4	72.6	65.9	70.9	57.2	87.5	78.6	55.9
(Ours)	(↑ 5.5)	(↓ 0.9)	(↑ 10.4)	(↑ 3.5)	(↑ 0.0)	(↑ 3.5)	(↑ 8.2)	(↑ 0.3)	(↑ 1.9)	(↑ 5.3)

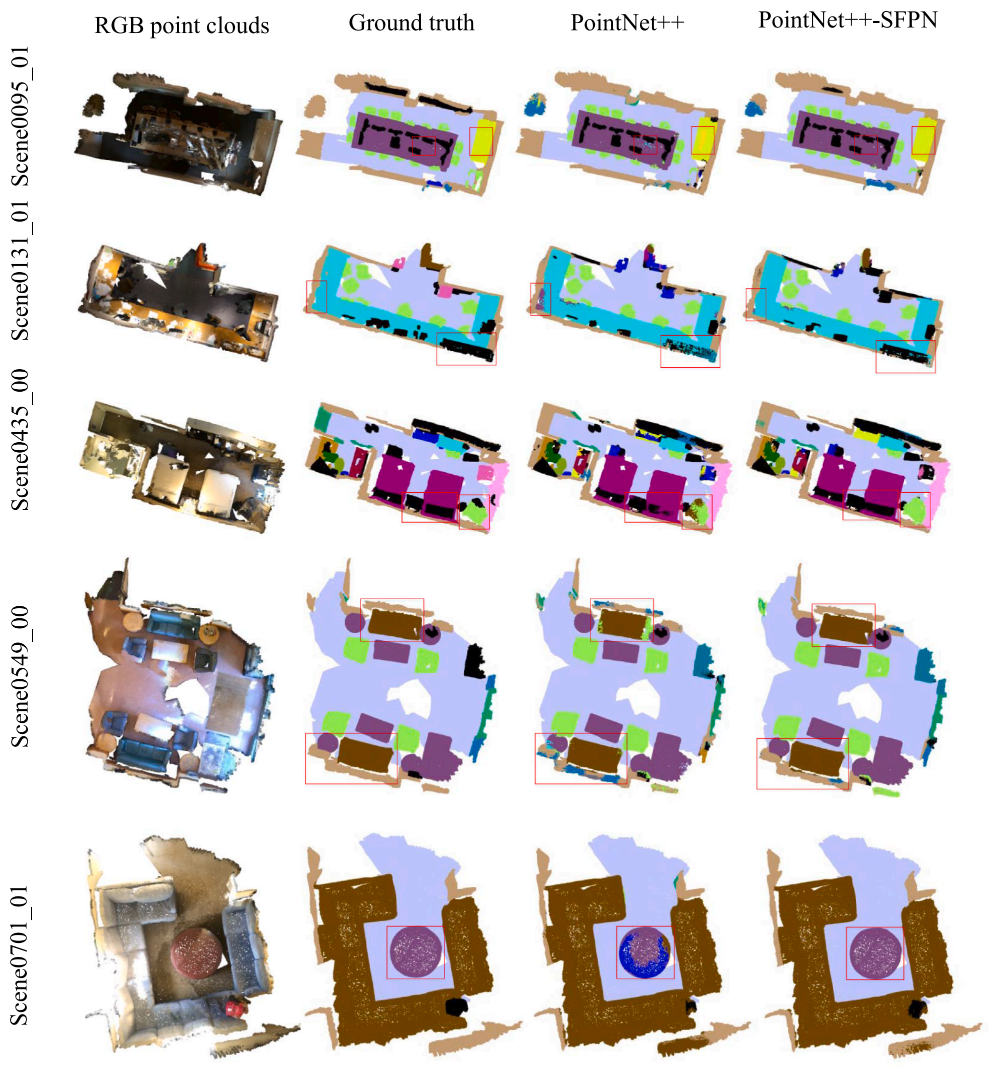


Fig. 4. Visualization of segmentation results of the baseline network PointNet++ and its SFPN variant on the ScanNet v2 dataset.

method and evaluate our searched FPN in exactly the same experiment protocol. Table 1 reports the mIoU results by comparing our proposed method with existing methods.

We can see that both PointNet++-SFPN and PointConv-SFPN achieve remarkable performance improvement. Specifically, PointNet++-SFPN and PointConv-SFPN respectively obtain a mIoU gain of 7.8% and 4.7% against their baselines (indicated by the red arrows and numbers). In addition, the IoU comparison for each category is shown in Table 2. It can be seen that PointNet++-SFPN and PointConv-SFPN obtain IoU improvement in most categories. This demonstrates both the effectiveness and generality of the searched architecture. Furthermore, our PointConv-SFPN surpasses all the latest methods in an evident margin.

Table 3
Comparison with existing methods on the S3DIS dataset in mIoU.

Method	mIoU(%)	OA(%)
PointNet (Qi et al., 2017a)	41.1	-
TangentConv (Tatarchenko et al., 2018)	52.6	85.5
DeepGCNs (Li et al., 2019)	53.8	-
PointCNN (Li et al., 2018b)	57.3	85.9
GridGCN (Xu et al., 2020)	57.8	86.9
SPG (Landrieu and Simonovsky, 2018)	58.0	86.4
HPEIN (Jiang et al., 2019)	61.9	87.2
PointASNL (Yan et al., 2020)	62.6	87.7
FPConv (Lin et al., 2020)	62.8	88.3
PointNet++ (Qi et al., 2017b)	59.3	86.0
PointNet++-SFPN (**Ours**)	**63.8**	**88.3**
	(↑ 4.5)	(↑ 2.3)
PointConv (Wu et al., 2019)	57.2	86.9
PointConv-SFPN (**Ours**)	**60.9**	**87.5**
	(↑ 3.7)	(↑ 0.6)

Note that, we do not focus on achieving state-of-the-art performance, but to improve the segmentation performance of existing networks. We notice that both PointConv-CE and APCF-Net develop a plug-and-play module on PointConv, which follow a similar strategy with ours. Although both of these two methods achieve significant improvement against the baseline PointConv, our method is able to earn a larger performance gains than them.

Qualitative visualization. To exhibit the comparison intuitively, we visualize the semantic segmentation results of PointNet++ and PointNet++-SFPN on several challenging scenes in Fig. 4. We use red boxes to highlight the main difference between these visualization. As shown in Fig. 4, we selected five scenes from the ScanNet v2 dataset listed from up to bottom. The first column is examples of RGB point clouds, the second column is segmentation ground truth, the third and fourth column are baseline of PointNet++ and combination of PointNet++ and our proposed SFPN module. Our PointNet++-SFPN generates much more accurate segmentation that its baseline network PointNet++. With the searched feature pyramid network, our method is able to capture much larger context information in a large receptive field and avoid some misclassification on large object. For example, on Scene00549_00, PointNet++ predicts the arm of the sofa as chair and misclassify some points on the wall. On in Scene0701_01, PointNet++ classify a part of the table as *other furniture*. This kind of mistake is mainly due to the lack of adequate context for PointNet++ to exploit. In addition, our network can manage more fine-grained details than its baseline. In Scene0095_01, PointNet++ predicts a few points on the table as desk. In Scene0131_01, PointNet++ commit some mistakes in the adjacent areas of objects. However, with SFPN, our proposed network can provide with more semantically rich high-resolution representation and avoid these errors.

4.3. Semantic segmentation on the S3DIS Dataset

The S3DIS dataset (Armeni et al., 2016) contains point clouds scanned from 6 large-scale indoor areas. These areas vary in appearance and structure, including office, copyroom, hallways, conference rooms, auditoriums and lounges areas. Similar to ScanNet, we adopt the same way to prepare the data. In the searching stage, we use the data of 1∼ 4 area to update the model parameters and use the *Area-6* data to update the architecture parameters. To reduce the search cost, we use PointNet++ as the backbone and only conduct the search for 30 epochs. Fig. 3 (b) shows the searched feature pyramid architecture on S3DIS. In the training stage, the epochs for training is 100. *Area-5* is used to test the generalization power and the rest areas are used for training. On purpose, the test data *Area-5* is absent during searching for a fair evaluation.

Quantitative analysis. We integrated the SFPN into two baseline networks and then compare them to several existing methods including PointNet (Qi et al., 2017a), TangentConv (Tatarchenko et al., 2018), DeepGCNs (Li et al., 2019), PointCNN (Li et al., 2018b), GridGCN (Xu et al., 2020), SPG (Landrieu and Simonovsky, 2018), HPEIN (Jiang et al., 2019), PointASNL (Yan et al., 2020) and FPConv (Lin et al., 2020). Table 3 shows the quantitative comparison results of the existing methods and network with SFPN added on S3DIS. It it obviously showed that our PointNet++-SFPN achieves excellent segmentation performance, with an mIoU of 63.8%. It surpasses its baseline network PointNet++ by 4.5%. Compared to PointConv, the proposed PointConv-SFPN makes a significant improvement in mIoU (i.e., 3.7%). Our goal is to augment the segmentation capacity of existing networks, but not to achieve state-of-the-art performance. In addition, the IoU achieved by our methods for each category is listed in Table 4. We can see that PointNet++ and PointConv integrated with our SFPN obtain IoU improvement in most categories, especially in category *sofa* (16.12%) and category *board* (12.79%). This further shows that the segmentation performance can be significantly improved with our searched FPN.

Qualitative visualization. As shown in Fig. 5, we visualize the semantic segmentation results of PointConv and our PointConv-SFPN on several challenging scenes. As shown in Fig. 5, there are four volumes from left to right. From left to right, the first column is RGB point clouds data. It is ground truth, the results of PointConv baseline network, and the results of combination of our proposed SFPN module and Poinv-Conv, respectively. With the searched feature pyramid network, our method is able to capture much larger context information in a large receptive field. This capacity enables our segmentation network a much more accurate prediction for the large object and the fine-grained details than its vanilla baseline. For example, the wall segmentation results of the second line with our module in the red box is more complete than only PointConv used. Three hangings on the wall segmentation results used our proposed module in the bottom red boxes of the last line are more accuracy and complete than only baseline network used.

Table 4
Segmentation results on the S3DIS dataset (Area-5) in per-class IoU (%).

Method	celling	floor	wall	beam	column	window	door
PointNet++	91.35	97.29	79.59	0.00	26.04	57.51	40.64
PointNet++-SFPN	93.36	97.68	81.02	0.00	25.81	64.15	37.94
(Ours)	(↑ 2.01)	(↑ 0.39)	(↑ 1.43)	(↑ 0.00)	(↓ 0.23)	(↑ 6.64)	(↓ 2.70)
PointConv	93.49	97.99	79.61	0.12	19.42	55.86	25.22
PointConv-SFPN	93.76	97.99	80.24	0.00	30.98	58.23	22.85
(Ours)	(↑ 0.27)	(↑ 0.00)	(↑ 0.63)	(↓ 0.12)	(↑ 11.56)	(↑ 2.37)	(↓ 2.37)
Method	table	chair	sofa	bookcase	board	clutter	
PointNet++	73.79	84.70	48.82	66.93	57.54	46.50	
PointNet++-SFPN	78.54	86.80	64.94	73.05	70.33	55.49	
(Ours)	(↑ 4.75)	(↑ 2.10)	(↑ 16.12)	(↑ 6.12)	(↑ 12.79)	(↑ 8.99)	
PointConv	78.73	87.06	35.11	67.63	51.07	52.48	
PointConv-SFPN	78.72	87.90	53.48	70.12	63.14	54.61	
(Ours)	(↓ 0.01)	(↑ 0.84)	(↑ 18.37)	(↑ 2.49)	(↑ 12.07)	(↑ 2.13)	

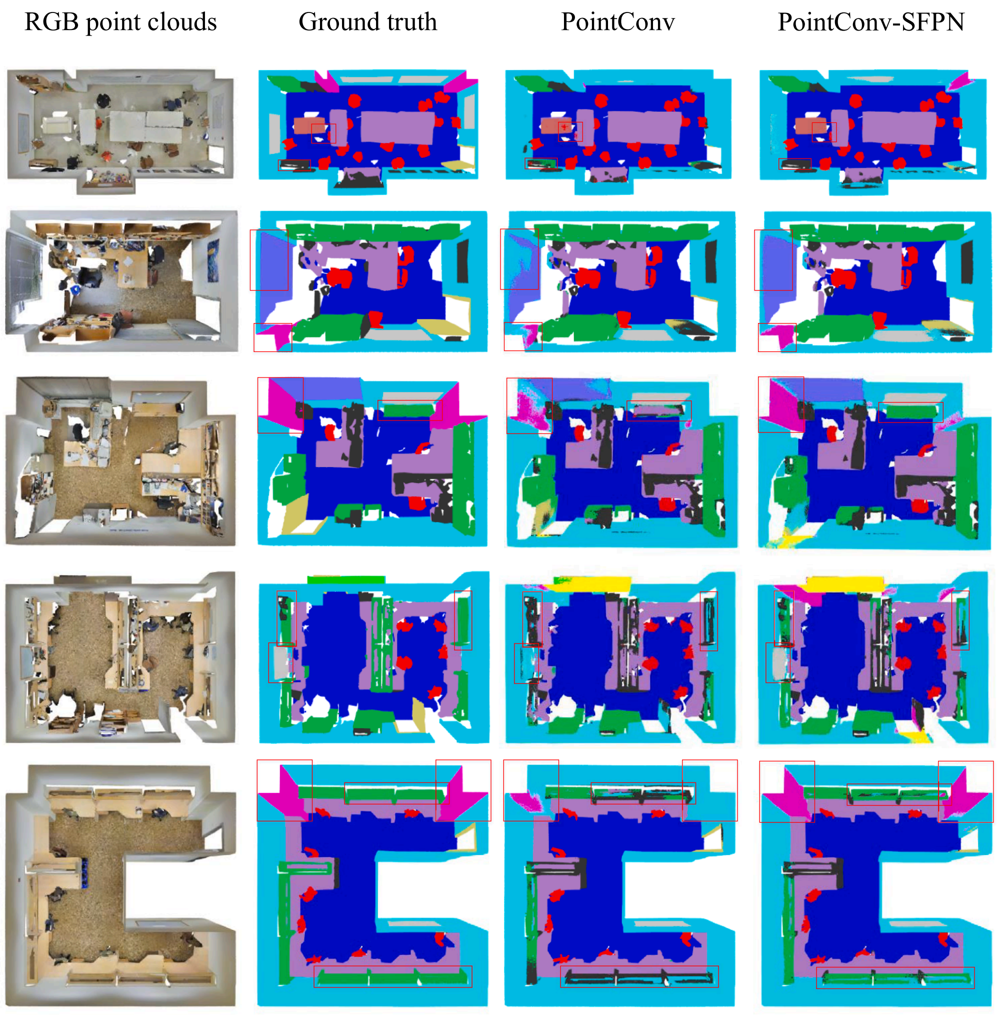


Fig. 5. Examples of semantic segmentation results of the baseline network PointConv and its SFPN variant on S3DIS dataset.

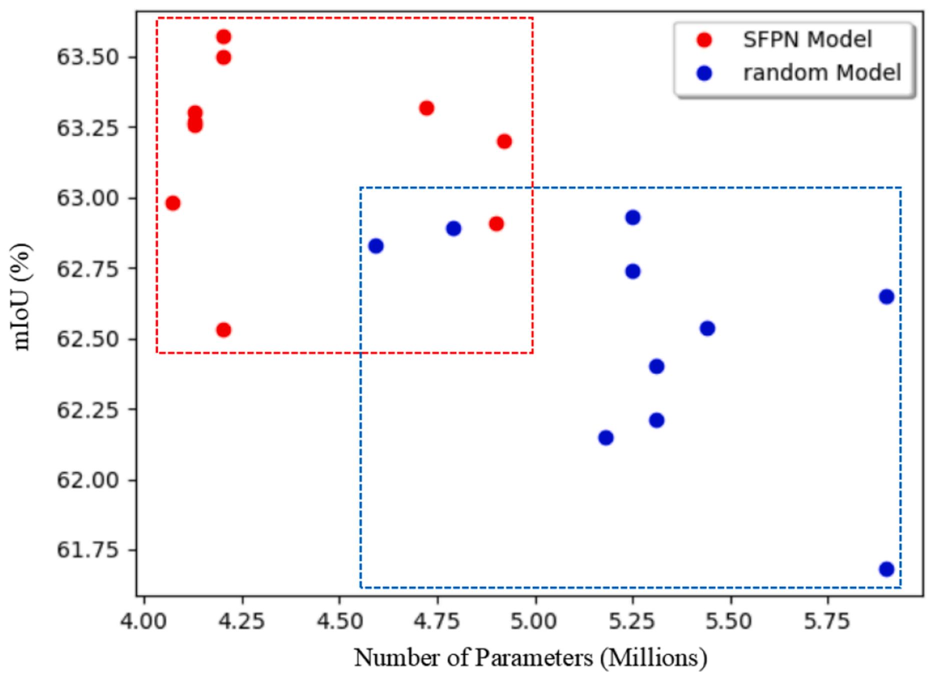


Fig. 6. Comparison between the automatically search and random sampling in the proposed search space.

Table 5
Comparison between our searched FPN and the manually designed FPN.

Method	Test mIoU(%)	Parameters(millions)
APCF-Net (Lin et al., 2020)	63.1	13.6
PointConv-SFPN(**ours**)	**64.1**	**9.6**

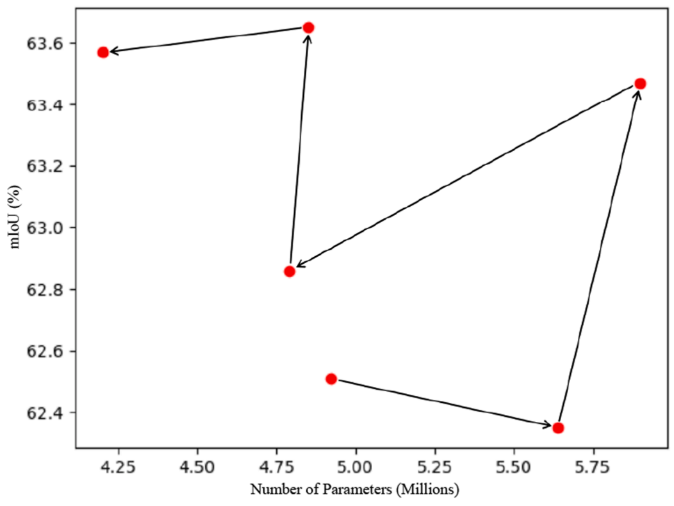
4.4. *Architecture search analysis*

Automatically search vs. random sampling. Previous studies (Radosavovic et al., 2019); Yang et al. (2020) have indicated that the search space plays a important role in the success of NAS. In this paper, we designed a search space covering all information communication paths for multi-level features. In addition, as described in Section 3.2, we applied a continuous relaxation on the designed search space and then conduct automatically search via SGD. To show that the success of our architecture search method does not entirely come from the search space, we compared the FPN architectures searched by SGD with the randomly sampled ones in the proposed search space. Specifically, we first randomly initialized the architecture parameters α and then decoded the architecture from α. This sampling will be conducted for 10 times and we get 10 random sampled FPN architectures. Meanwhile, we used our search methods to discover 10 FPN architectures. The backbone network in this experiment is PointNet++. We conducted this comparison on ScanNet v2 and these architecture were evaluated on the validation split (frequent submission for testing split result online is forbidden). As shown in Fig. 6, there are two boxes by red and blue that are denoting the mIoU distribution of proposed SFPN model and random model. The horizontal axis represents the amount of parameters, ranging from 4 million to 5.75 million, and the vertical axis represents mIoU. The red box contains the results using SFPN model and the blue box is including the results by random model. It is obviously illustrated that: 1) In general, the searched architectures by SGD outperforms the randomly sampled ones both on the segmentation performance and parameter efficiency, which is suggesting that the search algorithm benefits the proposed architecture search method more. 2) Though much more difficult than the automatically search by SGD, random sampling can still produce a relative good architecture, which can be ascribed to the reasonable design of the search space.

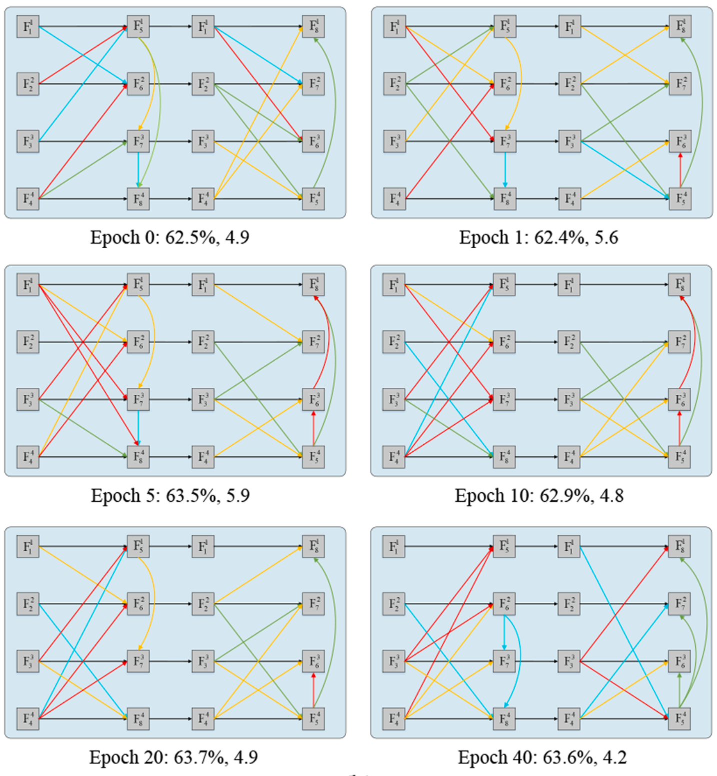
Automatical search vs. manual design. Table 1 and Table 3 have shown that PointNet++ (Qi et al., 2017b) or PointConv (Wu et al., 2019) integrated with our searched FPN can surpass many manually designed networks. Among all existing manually designed networks, APCF-Net is the only one that utilizes FPN for feature fusion. Moreover, APCF-Net adopted PointConv as the backbone network, which is same as our proposed PointConv-SFPN. To further demonstrate the effectiveness of our searched architecture against manually designed one, we conducted a more comprehensive comparison on APCF-Net and our PointConv-SFPN. Table 5 reports the comparison results in mIoU and the parameters. Compared with APCF-Net, our PointConv-SFPN can achieve higher segmentation performance with considerable less parameters, which is shown that the feature pyramid architecture discovered by our method enjoys stronger generalization power and higher parameter efficiency. It is more suitable for a variety of networks and modules with real-time requirements. Mainly duo to the operator of APCF-Net for

Table 6
Architecture transferability in mIoU(%).

Train / Search	PointNet++		PointConv	
	ScanNet	S3DIS	ScanNet	S3DIS
ScanNet	63.5(↑5.1)	62.6(↑3.3)	64.1(↑5.5)	61.9(↑4.7)
S3DIS	63.4(↑5.0)	63.8(↑4.5)	63.9(↑5.3)	60.9(↑3.7)



(a)



(b)

Fig. 7. Architecture evolution. Snapshots of the architecture search are in epoch 0, 1, 5, 10, 20 and 40, respectively.

constructing feature pyramid network is the PointConv operator, whose calculation is relatively large. Whereas the PointConv-SFPN used the lighter edge conv operator, the efficiency can be greatly improved.

Architecture transferability. To verify the performance of transferability, i.e. whether a FPN architecture searched in one dataset can perform well in another dataset,we tested the architecture transferability of our SFPN between ScanNet v2 and S3DIS. As shown in Table 6, SFPN perform a preferable architecture transferability between different datasets in general. The red upward arrows indicate the growth rates. Noted that, the reported mIoU on ScanNet v2 is the result on the validation split because frequent submission for test split result online is forbidden by the benchmark website. We can see that, with the architecture searched in PointNet++, our method boosts 5.1% mIoU for ScanNet and 5.5% in PointConv, respectively. The different datasets in these two networks, the transferability is less than the same datasets. It boosts by 3.3% of mIou for ScanNet to S3DIS in PointNet++, which is less than the performance from the same datasets.

Architecture evolution. To provide an intuition on the architecture evolution during the search, we presented the snapshots of the architecture search in epoch 0, 1, 5, 10, 20 and 40, respectively. As shown in Fig. 7(a), the architecture roughly evolves from the bottom right corner to the top left corner of the scatter plot. The red points means that the mIoU values varies with times of epoch and the number of parameters. This evolutionary trajectory from the direction of the arrow further demonstrates the effectiveness of our architecture search method. Fig. 7 (b) visualizes the architecture in each snapshot epoch including the various connection paths and states among features of each level. The physical meaning of different color of black, blue, yellow, red and green are the same with Fig. 3. It is illustrated that mIoU will be comparable higher and the parameter scale could be the smallest to reach 4.2 million as the epoch set to 40.

5. Conclusion

We proposed a feature pyramid architecture search method to automatically discover effective feature fusion necks for 3D indoor LiDAR point clouds semantic segmentation, which is the first attempt to extend NAS to feature fusion for point cloud segmentation task. The search space is specially designed for segmentation, which covers all possible information communication for features at different levels. Compared to the networks of PointNet++ and PointConv as the baseline, the searched feature pyramid network can work as a plug-and-play module to be integrated with any point-wise representation learning network. It strengthens point-wise features with more information both on semantic and localization by conducting communication across multi-level features. Extensive experiments demonstrate the generality and effectiveness of our method. Our method increases the performance by 7.8% and 4.7% in mIoU against the baseline networks. Compared with the manually designed FPN, our proposed SFPN reduces the parameter number by 4 millions (almost 30%) and improves the segmentation mIoU by 1%. Moreover, the experimental results demonstrate the excellent transferability of our method. Although the time cost of the search is relatively cheap (i.e., 3 h on ScanNet and 4 h on S3DIS), the amount of GPU memory occupation during the search process is too large. The huge memory occupation prevents us from considering more potential operators for the search space, which limits the size of the search space.

In the future, we plan to develop more memory-efficient architecture search method by introducing the architecture sampling trick into the searching process. Moreover, we will apply our method on more challenge outdoor data from various acquisition, such as TLS, Mobile or Airborne. Furthermore, we also want to extend our approach to other tasks, such as point cloud reconstruction and instance segmentation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors are grateful to Dr.Awase Syed from the Geospatial Sensing and Data Intelligence Lab of the University of Waterloo for insightful suggestions and amendments. This work was supported by the National Natural Science Foundation of China under grants of 41871380 and U1605254, and the Natural Sciences and Engineering Research Council of Canada under a grant of 50503-10284.

References

Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S., 2016. 3d semantic parsing of large-scale indoor spaces. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 1534–1543.

Boulch, A., Guerry, Y., Le Saux, B., Audebert, N., 2018. Snapnet: 3d point cloud semantic labeling with 2d deep segmentation networks. Comput. Graph. 71, 189–198.

Chen, L., Collins, M.D., Zhu, Y., Papandreou, G., Zoph, B., Schroff, F., Adam, H., Shlens, J., 2018. Searching for efficient multi-scale architectures for dense image prediction. In: Proc. of Advances in Neural Information Processing Systems (NeurIPS), pp. 8713–8724.

Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Niener, M., 2017. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 2432–2443.

Elsken, T., Metzen, J.H., Hutter, F., 2019. Neural architecture search: A survey. J. Mach. Learn. Res. 20, 55:1–55:21.

Fang, H., Lafarge, F., 2019. Pyramid scene parsing network in 3d: Improving semantic segmentation of point clouds with multi-scale contextual information. ISPRS J. Photogram. Remote Sens. 154, 246–258.

Guo, Z., Zhang, X., Mu, H., Heng, W., Liu, Z., Wei, Y., Sun, J., 2020. Single path one-shot neural architecture search with uniform sampling. In: Proc. of the European Conference on Computer Vision (ECCV). vol. 12361. pp. 544–560.

Jiang, C., Huang, K., Zhang, S., Wang, X., Xiao, J., 2020. Pay attention selectively and comprehensively: Pyramid gating network for human pose estimation without pre-training. In: ACM MM. pp. 2364–2371.

Jiang, L., Zhao, H., Liu, S., Shen, X., Fu, C., Jia, J., 2019. Hierarchical point-edge interaction network for point cloud semantic segmentation. In: Proc. of International Conference on Computer Vision (ICCV), pp. 10432–10440.

Justin, G., Samuel, S.S., Patrick, F.R., Oriol, V., George, E.D., 2017. Neural message passing for quantum chemistry. In: Proc. of International Conference on Machine Learning (ICML), pp. 1263–1272.

Kim, S., Kook, H., Sun, J., Kang, M., Ko, S., 2018. Parallel feature pyramid network for object detection. In: Proc. of Computer Vision and Pattern Recognition (CVPR). vol. 11209. pp. 239–256.

Landrieu, L., Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 4558–4567.

Li, G., Mller, M., Thabet, A., Ghanem, B., 2019. DeepGCNs: Can GCNs go as deep as CNNs? In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 9266–9275.

Li, H., Xiong, P., An, J., Wang, L., 2018a. Pyramid attention network for semantic segmentation. In: Proc. of the British Machine Vision Conference (BMVC), pp. 285–297.

Li, H., Xiong, P., Fan, H., Sun, J., 2019. Dfanet: Deep feature aggregation for real-time semantic segmentation. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 9522–9531.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B., 2018b. PointCNN:convolution on X-transformed points. In: Proc. of Advances in Neural Information Processing Systems (NeurIPS), pp. 820–830.

Lin, H., Luo, Z., Li, W., Chen, Y., Wang, C., Li, J., 2020. Adaptive pyramid context fusion for point cloud perception. IEEE Geosci. Remote Sens. Lett. doi: 10.1109/LGRS.2020.3037509.

Lin, T., Dollr, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 936–944.

Lin, Y., Yan, Z., Huang, H., Du, D., Liu, L., Cui, S., Han, X., 2020. FPConv: Learning local flattening for point convolution. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 4292–4301.

Liu, H., Guo, Y., Ma, Y., Lei, Y., Wen, G., 2020. Semantic context encoding for accurate 3d point cloud segmentation. IEEE Trans. Multimedia, doi: 10.1109/TMM.2020.3007331.

Liu, H., Simonyan, K., Yang, Y., 2019. Darts: Differentiable architecture search. In: Proc. of the International Conference on Learning Representations (ICLR), pp. 1–13.

Liu, S., Qi, L., Qin, H., Shi, J., Jia, J., 2018. Path aggregation network for instance segmentation. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 8759–8768.

Ma, L., Li, Y., Li, J., Tan, W., Yu, Y., Chapman, M.A., 2021. Multi-scale point-wise convolutional neural networks for 3d object segmentation from lidar point clouds in large-scale environments. IEEE Trans. Intell. Transp. Syst. 22 (2), 821–836.

Maturana, D., Scherer, S., 2015. Voxnet: A 3d convolutional neural network for real-time object recognition. In: IROS. pp. 922–928.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017a. PointNet: Deep learning on point sets for 3d classification and segmentation. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 984–993.

Qi, C.R., Su, H., Niener, M., Dai, A., Yan, M., Guibas, L.J., 2016. Volumetric and multi-view cnns for object classification on 3d data. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 5648–5656.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017b. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: Proc. of Advances in Neural Information Processing Systems (NeurIPS), pp. 5099–5108.

Radosavovic, I., Johnson, J., Xie, S., Lo, W., Dollar, P., 2019. On network design spaces for visual recognition. In: Proc. of International Conference on Computer Vision (ICCV), pp. 1882–1890.

Real, E., Aggarwal, A., Huang, Y., Le, Q.V., 2019. Regularized evolution for image classifier architecture search. In: Proc. of the AAAI Conference on Artificial Intelligence (AAAI), pp. 4780–4789.

Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E., 2015. Multi-view convolutional neural networks for 3d shape recognition. In: Proc. of International Conference on Computer Vision (ICCV), pp. 945–953.

Tang, H., Liu, Z., Zhao, S., Lin, Y., Lin, J., Wang, H., Han, S., 2020. Searching efficient 3d architectures with sparse point-voxel convolution. In: Proc. of the European Conference on Computer Vision (ECCV). Vol. 12373. pp. 685–702.

Tatarchenko, M., Park, J., Koltun, V., Zhou, Q., 2018. Tangent convolutions for dense prediction in 3d. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 3887–3896.

Wang, C., Samari, B., Siddiqi, K., 2018a. Local spectral graph convolution for point set feature learning. In: Proc. of the European Conference on Computer Vision (ECCV). vol. 11208. pp. 56–71.

Wang, X., Girshick, R.B., Gupta, A., He, K., 2018b. Non-local neural networks. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 7794–7803.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Solomon, J.M., 2019. Dynamic graph cnn for learning on point clouds. ACM Trans. Graph. 38 (5).

Wu, W., Qi, Z., Fuxin, L., 2019. PointConv: Deep convolutional networks on 3d point clouds. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 9613–9622.

Xu, H., Yao, L., Li, Z., Liang, X., Zhang, W., 2019. Auto-fpn: Automatic network architecture adaptation for object detection beyond classification. In: Proc. of International Conference on Computer Vision (ICCV), pp. 6648–6657.

Xu, Q., Sun, X., Wu, C., Wang, P., Neumann, U., 2020. Grid-GCN for fast and scalable point cloud learning. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 5660–5669.

Yan, X., Zheng, C., Li, Z., Wang, S., Cui, S., 2020. PointASNL: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 5588–5597.

Yang, A., Esperana, P.M., Carlucci, F.M., 2020. Nas evaluation is frustratingly hard. In: Proc. of the International Conference on Learning Representations (ICLR). pp. 1–13.

Yang, J., Kang, Z., Zeng, L., Hope Akwensi, P., Sester, M., 2021. Semantics-guided reconstruction of indoor navigation elements from 3d colorized points. ISPRS J. Photogram. Remote Sens. 173, 238–261.

Yang, J., Zhang, Q., Ni, B., Li, L., Liu, J., Zhou, M., Tian, Q., 2019. Modeling point clouds with self-attention and gumbel subset sampling. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 3318–3327.

Yu, F., Wang, D., Shelhamer, E., Darrell, T., 2018. Deep layer aggregation. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 2403–2412.

Zhang, D., Zhang, H., Tang, J., Wang, M., Hua, X., Sun, Q., 2020a. Feature pyramid transformer. In: Proc. of the European Conference on Computer Vision (ECCV). vol. 12373. pp. 323–339.

Zhang, J., Zhu, C., Zheng, L., Xu, K., 2020b. Fusion-aware point convolution for online semantic 3d scene segmentation. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 4533–4542.

Zhang, Y., Qiu, Z., Liu, J., Yao, T., Liu, D., Mei, T., 2019. Customizable architecture search for semantic segmentation. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 11633–11642.

Zhao, H., Jiang, L., Fu, C., Jia, J., 2019. PointWeb: Enhancing local neighborhood features for point cloud processing. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 5560–5568.

Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V., 2018. Learning transferable architectures for scalable image recognition. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 8697–8710.