Pyramid Point: A Multilevel Focusing Network for Revisiting Feature Layers

Nina Varney, Member, IEEE, and Vijayan K. Asari, Senior Member, IEEE

Abstract—We present a method to learn a diverse group of object categories from an unordered point set. We propose our Pyramid Point network, which uses a dense pyramid structure instead of the traditional “U” shape, typically seen in semantic segmentation networks. This pyramid structure gives a second look, allowing the network to revisit different layers created from various levels on the network, allowing for feature propagation. We introduce a focused kernel point convolution (FKP Conv), which expands on the traditional point convolutions by adding an attention mechanism to the kernel outputs. This FKP Conv increases our feature quality and allows us to weigh the kernel outputs dynamically. These FKP Convs are the central part of our Recurrent FKP Bottleneck block, which makes up the backbone of our encoder. With this distinct network, we demonstrate competitive performance on three benchmark datasets.

Index Terms—3-D, LiDAR, point clouds, semantic segmentation.

I. INTRODUCTION

Recent breakthroughs in automatic driving and the decrease in LiDAR sensor cost have increased research into 3-D object detection. One of the most popular tasks in the semantic segmentation of unorganized 3-D point clouds. Three-dimensional point clouds present a distinct type of challenge because of their sparse and unordered nature. These point clouds exist as lists of points containing X, Y, and Z information with other possible features like reflectivity or RGB data. The resolution and occlusions in a point cloud vary greatly, not only from sensor-to-sensor but also within each scene. The points are spatially related, however, the relationships are not uniform like we would find in 2-D imagery. This property makes operations like convolution difficult.

Most 3-D scene segmentation methods use the traditional U-Net type structure with encoder layers, followed by decoder layers, usually with connections between similar-sized encoder and decoder layers. However, this U structure has some shortcomings, particularly rough results that struggle to segment fine details. This failure is due to the max-pooling and subsampling of the feature layers, which results in a reduced feature map resolution. This extreme reduction in the receptive field at each encoder layer makes it very difficult for this network structure to segment small objects with a high degree of accuracy.

We propose our Pyramid Point network, which uses feature fusion to combat reduced feature map resolution issues. Pyramid Point passes features in a dense pyramid structure. This structure results in feature fusion, from the encoding layers to the corresponding decoding layers, between encoder and decoder units of different layers and between the decoder layers themselves. By allowing the features to traverse between different units and layers, not just restricting them to the defined U path, we will enable the network to gain different receptive field views. This structure also provides the advantage of having several “shallow” feature layers, which have not gone through the entire U structure. Because the decoding unit adds noise, features that have gone through one or two decoder units, instead of the traditional four, are less susceptible to segmentation errors, which is especially crucial for small objects.

We also introduce a focused kernel point convolution (FKP Conv). This convolution operation uses a kernel of points combined with a linear distance correlation to perform a convolution on a set of unordered points. After the convolution operation, we apply an attention mechanism to the output features. This attention module uses a combination of global pooling operations and shared MLPs to highlight essential features from the kernel convolution output. This FKP Conv is the critical element in our Recurrent FKP Bottleneck block.

In our evaluation section, we show the success of our semantic segmentation network. We assess our performance on three different datasets, Dayton Annotated Laser Earth Scan (DALES) [1], Paris-Lille 3D [2], and Semantic3D [3]. We chose these three LiDAR datasets because of their distinctly different signatures viz. aerial, mobile, and terrestrial type LiDAR. This evaluation strategy allows us to assess our network’s overall performance and robustness to different object categories, resolutions, and occlusions. We show that our network outperforms other networks on the DALES and Paris-Lille 3D datasets while providing competitive performance on the Semantic3D dataset and lessening the variation in results across object categories.

Our contributions are the following.

1) We propose our Pyramid Point network, a multilevel focusing network that allows for dense feature fusion across all layers of the network.

2) We introduce an FKP Conv which adds spatial and channel attention elements to the traditional kernel point convolutions (KP Convs).
3) We demonstrate the effectiveness of this network on three benchmark datasets, representing three distinct LiDAR types.

II. RELATED WORKS

A. Semantic Segmentation of Point Clouds

In point cloud data, many original semantic segmentation methods focused on transforming the point cloud into a structured dataset using voxels. Recently, newer methods have concentrated on performing operations directly on the points themselves. PointNet [4] was the first architecture to use point sets as inputs, and the authors developed a method to extract global features from these point sets. PointNet++ [5] expanded on PointNet by adding a local hierarchy. Pointwise multilayer perceptron networks, like PointNet, were prevalent, but several networks have recently defined specific point convolutions performed directly on the inputs. The most notable point convolution operation is the KP Conv [6], which uses a linear correlation to connect the kernel points to their closest input points.

B. Pyramid Networks

Medical image segmentation tasks have a rich history of hierarchical methods. U-Net [7] is the prominent architecture, suggesting a hierarchical approach to image segmentation with cross-architecture connections. NABLA-N (N^N) Net [8] uses a pyramid structure of hierarchical layers. While many improvements to the typical encoder-decoder structure have been made in the 2-D image space, the 3-D image space employs mostly the same architectures, focusing on improvements to the convolutional units themselves rather than the architecture.

C. Attention Modules

Attention modules have significantly impacted deep learning with their ability to efficiently model dependencies. Attention mechanisms have many versions, but the most popular lately are varying channel and spatial attention combinations. RCAN [9] and CBAM [10] both use a 2-D convolution combined with channel attention. Some attention mechanisms are applied to the 3-D point cloud space with methods like [11].

III. PYRAMID POINT NETWORK

A. Feature Pyramids

In a U-NET-type model for semantic segmentation, there are two high-level building blocks, the encoder, and the decoder. The encoder consists of a series of subsampling and convolutional operations. The series of encoders terminates at a final bottleneck layer, the latent space. These features, which form the latent space, are then run through a series of decoders that combine an upsampling operation and a transpose convolution. For each layer of the individual encoders, the set of features generated represents different qualities. Lower layers represent low-level features, like edges and corners, while higher-level features describe more complex representations like texture or object shape. A typical U-Net structure for semantic segmentation decodes only the features from the final latent space. We also know that the decoder is sensitive to noise [12], which compounds when a set of features is run through multiple decoding units.

We propose an improvement that decodes from multiple feature spaces instead of just the final latent space, allowing us to reuse the already learned features, inspired by the DenseNet [13]. Reusing the encoding results from previous feature layers will enable us to reduce parameters. Additionally, by combining feature layers generated from different latent spaces, instead of simply those with common dimensionality, we can provide a wider variety of features.

Except for the first two downsampling layers, each layer is both downsampled and upsampled, resulting in several layers of similar features, each derived from a different point in the network. After some experimentation, we found the best configuration is to concatenate the similar elements at each layer of the network. The final network configuration is seen in Fig. 1. In Sections III-B–III-D, we discuss the adjustments we made to the KP Conv and the encoder and decoder units in detail.

B. Focused KP Conv

1) Kernel Attention: To address the idea of scale within different objects in a scene, we introduce the concept of kernel attention. In the original KP Conv solution [6], the final output features are produced by summing the outputs of each of the \( N \) number of kernels. We argue that adding attention to the kernel modules can increase the network’s discrimination ability. A direct linear correlation in the system may not always be the best application. By adding an attention mechanism, we can help determine which of the kernel weights has the most significant contribution.

We propose an attention module on the kernel outputs before the final summation to produce the output features. This adjustment will allow the network to learn discrimination between the different kernel points and allow for a more sophisticated kernel weighting, rather than relying strictly on geometric correlation alone. The kernel attention is as follows.

We consider the kernel outputs as \( F \in \mathbb{R}^{K \times N \times D} \), where \( K \) is the number of kernel points, \( N \) is the number of points and \( D \) is the feature dimension. We take the max and mean pooling of the feature \( F \), with respect to the kernel outputs, to get \( F_{\text{max}} \in \mathbb{R}^{1 \times N \times D} \) and \( F_{\text{mean}} \in \mathbb{R}^{1 \times N \times D} \). \( F_{\text{mean}} \) and \( F_{\text{max}} \) are concatenated together and run through two shared MLPs. The MLP layers are followed by a sigmoid function, \( \sigma \), which makes the kernel attention matrix \( M_K(F) \in \mathbb{R}^{1 \times N \times D} \). In the final step, we use element-wise multiplication to get the final feature \( F' \). The entire process is shown below

\[
M_K(F) = \sigma(\text{MLP}(F_{\text{mean}}, F_{\text{max}}))F_{\text{FKP}} = F \otimes M_K(F).
\]

A summation across all kernel features then reduces the final enhanced kernel outputs to make the final point features from our FKP Conv, \( F_{\text{FKP}} \in \mathbb{R}^{N \times D} \). The entire kernel attention architecture can be seen in Fig. 2.
Fig. 1. Example of the architecture of our Pyramid Point network is shown on the left. We constructed the architecture in an inverse dense pyramid shape instead of the typical U shape. At the third layer, features are both downsampled and upsampled, and same-sized layers are concatenated together in the final stages. The kernel attention module is on the right. This module is applied to the kernel outputs in order to enhance the features, before reducing the kernel dimension.

Fig. 2. Kernel attention module. This module is applied to the kernel outputs in order to enhance the features, before reducing the kernel dimension.

C. Encoder and Decoder

Once we have established our FKP Conv, we then define our encoder’s building blocks. We use the bottleneck structure commonly found in ResNet, with a few adjustments. We begin with a unary convolution, with a Leaky ReLu activation. This result is passed into a recurrent FKP Conv. We design the output of the recurrent layer as follows:

\[ O(t) = w_f^f \ast x^f(t) + w_r^f \ast x^r(t + 1). \] (2)

We can extend the equation to include any number of hidden layers. This recurrent FKP is followed by a Leaky ReLu activation and another unary convolution with Leaky ReLu activation. The final result is then summed with the original input features to form the final enhanced output.

Our encoder unit consists of two layers; the first is the recurrent FKP bottleneck block followed by an FKP strided bottleneck block. This FKP strided bottleneck uses a single strided FKP Conv in place of the recurrent FKP Conv, as shown in Fig. 1. Our decoder consists of a nearest upsampling layer, followed by a unary convolution.

D. Implementation

We used the Adam optimizer with the default parameters and a learning rate of 0.01% and a 1% decay over each epoch. We used a fixed number of kernel points, 15 total. All parameters were the same throughout each experiment except for the kernel radius, which was adjusted based on the resolution of the dataset. All experiments were performed on a single NVIDIA TITAN RTX GPU.

IV. Evaluations

We evaluate our network on three different types of LiDAR benchmark datasets; DALES [1], Semantic3D [3], and Paris-Lille 3D [2]. We chose these benchmarks because of their popularity and because they represent three distinct types of LiDAR, aerial laser scanners (ALSs), terrestrial laser scanners (TLSs), and mobile laser scanners (MLSs). These different sensors all provide various object classes, resolutions, and types of occlusions and will allow us to test our network’s robustness.

For each evaluation, we used five total layers in our architecture and began upsampling and downsampling simultaneously at layer three. The feature dimensions are 64/128/256/512/1024, respectively. For our recurrent FKP Conv bottleneck, we use three hidden layers. After several experiments, we concluded that the best configuration for combining similarly shaped layers was to concatenate them.
We changed the subsampling parameter and neighborhood radii based on the resolution of the point cloud. Aside from the subsampling rate and neighborhood radius, no other parameter changes were made when testing different datasets.

We examine our network’s performance using mean intersection over union (IoU) as our primary metric. For consistency with the benchmark, we also report per class IoU and overall accuracy, where appropriate. A discussion of the results from each dataset is provided below.

A. DALES

Table I shows the performance of our network on DALES. We can see that the Pyramid Point has a good performance with the highest mean IoU, of over 2.5% more than the next closest method. One notable performance is between our network and KPConv, whose KP Conv makes up the backbone of our FKP Conv. We can see a large margin of performance improvement, especially in some of the lower performing categories. This emphasizes our pyramid architecture’s effectiveness and ability to exploit the features’ interconnections and bridge the performance gap between object categories.

Visual results of a DALES scenes are shown in Fig. 3.

B. Paris-Lille 3D

Table II shows our results compared against others in the benchmark. Overall, our method comes in first in this dataset. We also claim the top performance in the ground, natural, and pedestrian categories. We can also examine the qualitative results by looking at a selection of scenes shown in Fig. 4.

C. Semantic3D

The results of our network are in Table III. Like others [6], [11]; we only compare published methods, and we obtain the scores from their benchmark. We can see that our method outperforms all other methods except for RandLA Net, which performs similarly with a 0.1% difference between the two methods. Pyramid Point also logs the highest performance in the cars category, by 3.5%.

V. Ablation Study

We perform two ablation studies to discover the effects of some of our contributions to the network. We focus on the impact of the recurrent layers and how many hidden layers contribute to positive network performances. We also explore
how the FKP Conv and variations to the pooling methods within FKP Conv can affect the final results. We use a subset of the Semantic3D training data to perform this experiment because the test set cannot be used for ablation experiments.

A. Recurrent Layers

The first set of experiments compares the full intact network and observes the effects of different numbers of hidden layers in our recurrent structure. Table IV shows the effects of different numbers of hidden layers. Adding the recurrent layers improves the performance, increasing when going from two to three hidden layers. This performance drops off when increasing the number of hidden layers from three to four.

B. FKP Conv

Next, we compare the FKP Conv and how it affects our network results. We first examine the network’s effects using a KP Conv, without the kernel attention elements. We explore using max and mean pooling in the attention mechanism, and then finally, we explore the combination of all of them to make up the final FKP Conv. For this evaluation, we use a subset of the Paris-Lille 3D data.

Table V compares the results from all ablated networks. We note that adding the kernel’s attention to the KP Conv gives the biggest jump in performance; we then get a marginal improvement by considering the max and mean pooling together to form the final FKP Conv element.

| Variation                          | mIoU  |
|-----------------------------------|-------|
| (1) No Focused Kernel            | 85.4  |
| (2) Max Focused Kernel           | 89.1  |
| (3) Mean Focused Kernel          | 88.3  |
| (4) Max, Mean Focused Kernel     | 90.2  |

VI. Conclusion

In this letter, we have presented a novel network for semantic segmentation in 3-D point clouds. Our Pyramid Point network uses the concept of a dense pyramid structure to increase the number of receptive fields and revisit feature layers. This structure, which varies from the traditional U-style networks, provides additional details and less noise in the feature channels. We also introduced an FKP Conv, which utilizes an attention mechanism to improve outputs from the convolutional kernel. We demonstrated the network’s success by showing results on three prominent benchmark datasets, DALES, Semantic3D, and Paris-Lille 3D. The Pyramid Point was the top performer in the DALES and Paris-Lille 3D datasets and performed competitively in the Semantic3D dataset, demonstrating the network’s success in three different types of LiDAR environments.

ACKNOWLEDGMENT

The views expressed in this letter are those of the authors and do not reflect on the official policy of the Air Force, Department of Defense or the U.S. Government.

REFERENCES

[1] N. Varney, V. K. Asari, and Q. Graehling, “DALES: A large-scale aerial LiDAR data set for semantic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2020, pp. 717–726.
[2] X. Roynard, J.-E. Deschaud, and F. Guetelle, “Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification,” Int. J. Robot. Res., vol. 37, no. 6, pp. 545–557, May 2018, doi: 10.1177/0278364918767506.
[3] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys, “Semantic3D.Net: A new large-scale point cloud classification benchmark,” 2017, arXiv:1704.03847.
[4] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, “PointNet: Deep learning on point sets for 3D classification and segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 77–85.
[5] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet++: Deep hierarchical feature learning on point sets in a metric space,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 5099–5108.
[6] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Guetelle, and L. Guibas, “KPCnon: Flexible and deformable convolution for point clouds,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 6410–6419.
[7] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. Cham, Switzerland: Springer, 2015, pp. 234–241.
[8] M. Z. Alom, T. Aspiras, T. M. Taha, and V. K. Asari, “Skin cancer segmentation and classification with NLABA-N and inception recurrent residual convolutional networks,” 2019, arXiv:1904.11126.
[9] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, “Image super-resolution using very deep residual channel attention networks,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 286–301.
[10] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, “CBAM: Convolutional block attention module,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 3–19.
[11] X. Hu et al., “RandLA-Net: Efficient semantic segmentation of large-scale point clouds,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 11105–11114.
[12] T. He, X. Tan, and T. Qin, “Hard but robust, easy but sensitive: How encoder and decoder perform in neural machine translation,” 2019, arXiv:1908.06259.
[13] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2261–2269.
[14] A. Boulch, “ConvPoint: Continuous convolutions for point cloud processing,” Comput. Graph., vol. 88, pp. 24–34, May 2020.
[15] Y. Wang, T. Jiang, T. Liu, X. Li, and C. Liang, “Hierarchical instance recognition of individual roadside trees in environmentally complex urban areas from UAV laser scanning point clouds,” ISPRS Int. J. Geo-Inf., vol. 9, no. 10, p. 595, Oct. 2020.
[16] H. Luo, C. Chen, L. Fang, K. Khoshelham, and G. Shen, “MS-RRFSegNet: Multiscale regional relation feature segmentation network for semantic segmentation of urban scene point clouds,” IEEE Trans. Geosci. Remote Sens., vol. 58, no. 12, pp. 8301–8315, Dec. 2020.
[17] A. Boulch, G. Puy, and R. Marlet, “FKAConv: Feature-kernel alignment for point cloud convolution,” in Proc. Asian Conf. Comput. Vis., 2020. [Online]. Available: https://boulch.eu/publications/2020_accv_fkaconv
[18] L. Landrieu and M. Simonovsky, “Large-scale point cloud semantic segmentation with superpoint graphs,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 4558–4567.