Task-driven Differentiable Grouping for Point Cloud Analysis

Shuo Cheng* Zhengjie Xu* Quan Vuong*

Abstract

Super-pixels provide an efficient low/mid-level representation of image data, which greatly reduces the number of image primitives for subsequent vision tasks. Analogy to super-pixel defined on 2D images, clique on point clouds represents a cluster of points that share the same properties (e.g., colors, semantics, or local geometries). However, existing grouping algorithms are not differentiable, making them hard to be integrated into standard deep neural networks. We propose a novel framework for end-to-end learning this useful abstraction through specified downstream tasks. Instead of computing the hard association that is adopted in most clustering algorithms, we compute the soft association between points and cliques, which makes the clique computation fully differentiable. Experiments with different point cloud analysis tasks on major benchmarks demonstrate the effectiveness of our proposed method.

1. Introduction

Super-pixels are the image regions generated by grouping image pixels. Comparing to pixels, super-pixels provides a perceptually meaningful tessellation of image content, thereby reducing the number of image primitives for subsequent image processing. Owing to their representational and computational efficiency, super-pixels have become an established low/mid-level image representation and are widely-used in computer vision algorithms such as detection [10, 9], semantic segmentation [2, 8, 1], object tracking [12] and saliency prediction [4, 11]. Super-pixels are especially widely-used in traditional energy minimization frameworks, where a low number of image primitives greatly reduce the optimization complexity.

Analogy to super-pixel defined in image space, we define clique on point clouds, which represent a cluster of points that share the same properties (e.g., colors, semantics, or local geometries). Our **motivation** is that by grouping the point cloud representation in local scales, the intermediate abstraction can simplify and ease the learning of downstream prediction tasks. In addition, the input point cloud may not evenly distributed in the spatial domain, we leverage the learnt clique to adjust the spatial density of the point representation. However, existing grouping algorithms are not differentiable, making them hard to be integrated into standard deep neural networks. Some literatures [3, 5] on clustering algorithms propose to replace the hard assignment with soft association, which makes the grouping step end-to-end trainable. Inspired by their works, we propose a novel framework for **D**ifferentiable **P**oint cloud **G**rouping (DPGNet) that allow learning of this useful abstraction through specified downstream tasks. To this end, our contributions are as follows:

- We implement a soft-clustering module for point cloud feature grouping which enables end-to-end training.
- The proposed soft-clustering module can be easily integrated into standard deep learning framework, which allows learning of task-specific mid-level abstraction.
- We evaluate our DPGNet on different point cloud analysis tasks on major benchmarks, our experiments demonstrate that the learnt representation can boost the performance of subsequent tasks.

2. Related Work

There have been many methods proposed for the task of superpixel proposal for images. They can broadly be categoried as either graph-based approach or clustering-based approach. However, the majority of these either operates on discrete variables, or requires handcrafted feature representation. These constraints limit their applicability into and end-to-end differentiable pipeline. Superpixel Sampling Network proposes to replaces the hard pixel-superpixel association in SLIC with soft association, thereby allows for gradients to back-propagated through the iterative clustering computation. Their method allows for the integration of learnable feature representation and clustering assignment into an end-to-end differentiable pipeline. However, their method has only been demonstrated for computer vision tasks in pixel domain, and not 3D domain.

* Equal contribution.

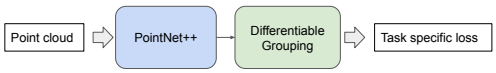

Figure 1. Illustration of the proposed DPGNet.

In the 3D domain, there are also works which propose clique, the equivalent of superpixel for point cloud representation. However, the cliques are proposed based on simple k-means clustering with hand-tuned clustering hyperparameter and is unable to adapt to the statistical patterns presented in the data.

3. Proposed Method and Implementations

This section describes our extension of differentiable grouping to 3D data.

3.1. Differentiable Grouping

In this section we introduce our implementation of the differentiable grouping module. Instead of computing hard point-clique associations, we propose to compute soft-associations between points and cliques. Specifically, for a point p and clique i at iteration t, let k be the feature dimension, n and m be the number of point and number of clique, $F_p \in \mathbb{R}^k$ be the feature vector of point p, $S_i \in \mathbb{R}^k$ be the feature of clique i. We first compute soft-associations matrix $Q \in \mathbb{R}^{n \times m}$:

$$Q_{pi}^t = e^{-||F_p - S_i^{t-1}||^2} \tag{1}$$

Correspondingly, the computation of new clique centers is modified as the weighted sum of point features,

$$S_i^t = \frac{1}{Z_i^t} \sum_{p=1}^{n} Q_{pi}^t F_p, \tag{2}$$

where $Z_i^t = \sum_p Q_{pi}^t$ is the normalization constant. The size of Q is $n \times m$ and even for a small number of cliques m, it is prohibitively expensive to compute Q_{pi} between all the points and cliques. Therefore, we constrain the distance computations from each pixel to only 27 surrounding cliques. This brings down the size of Q from $n \times m$ to $n \times 27$, making it efficient in terms of both computation and memory.

3.2. Learnable Clique for 3D Vision Tasks

We propose to integrate the differentiable grouping module with standard deep learning network (shown in Fig 1), the resulting novel framework (DPGNet) allow end-to-end learning of the point cloud feature clustering. In this section we describe the potential applications of our proposed DPGNet.

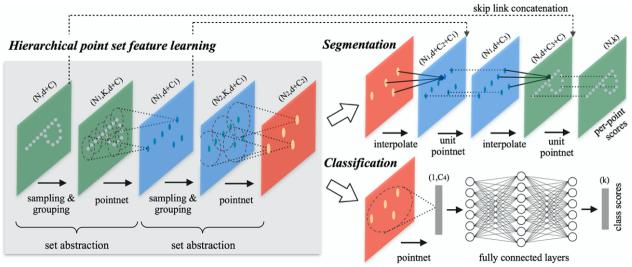

Figure 2. Architecture of PointNet++

Classification Point cloud classification aims to predict a category label of the input point cloud. Such task require a comprehensive understanding of the whole shape as well as the geometry relationship between each part. In order to improve the recognition accuracy, we embed the proposed differentiable grouping module with off-the-shelf point cloud feature extractors (i.e., PointNet [6], PointNet++ [7]). The clustered feature points are then fused to predict the final results. Our motivation is that the input point cloud is not evenly distributed in the space, the iterative grouping process acts as a role of adjusting the spatial density as well as aggregating local and global information.

Part Segmentation Part segmentation is a challenging fine-grained 3D recognition task. Given a point cloud or mesh, the task is to assign part category label (e.g. chair leg, cup handle) to each point or face. We leverage the proposed differentiable grouping module to optimize the embedding space and enforce the local feature consistency, the mid-level abstraction thus can be used to discover geometry primitives as well as simplify the downstream tasks.

Semantic Scene Parsing Our proposed DPGNet on part segmentation can be easily extended to semantic scene segmentation, where point labels become semantic object classes instead of object part labels.

3.3. Integrating differentiable grouping into PointNet++

In SSN, a convolutional network extracts feature for each pixel which are then used to perform differentiable grouping and produce the intermediate representation for downstream tasks. A straight-forward extension of this pipeline to 3D data, such as point cloud, would include introducing a multi-layer perceptron (MLP) to extract the feature for each point. It would then perform iterative grouping using the exponential of negative norm of difference to compute the similarity between cluster centroid and points and using the weighted sum function to compute new features for the cluster centroids. Additionally, SSN selects the initial centroid positions as the nodes in a 2d grid super-imposed on the image. For 3D data, we can use the nodes of a uniform 3d meshgrid as the initial positions of the centroids.

We found that a straight-forward integration of differentiable grouping, as described above, into PointNet++ was not possible due to the current design choices in PointNet++ (discussed in more details below). The architecture of PointNet++ are shown in Figure 2. The Set Abstraction Layer performs point grouping and feature aggregation through these three steps:

1. Iterative FPS to find centroids
2. Ball querying to find nearby points for each centroids
3. Compute features for each centroids with PointNet

As can be seen, PointNet++ already performs grouping of points and feature aggregation of points within one group. Instead of using the nodes of a uniform 3d meshgrid as the initial positions of the centroid, PointNet++ uses iterative furthest point sampling (FPS). PointNet++ also uses PointNet to aggregate features of the points within one cluster and compute the feature of the cluster centroid instead of the weighted sum operation in SSN.

In contrast to the grouping operation in SSN, the grouping in PointNet++ is not iterative and is not differentiable. The relevant question is then how to use the motivations of differentiable clustering in SSN to improve the grouping operation in PointNet++. We propose to modify the Set Abstraction Layer as described:

Algorithm 1 Iterative Differentiable Grouping

Initialize centroids xyz
for iter **do**
  $group\_xyz = BallQuery(xyz)$
  for all centroid **do**
    $dist = ||points\_feat - centroid\_feat||_2$
    $sim = exp(-dist)$
    $new\_xyz = \sum sim * point\_xyz$
  end for
  Use PointNet to compute new $centroid\_feat$
end for
return $centroid\_xyz\ centroid\_feat$

To compute the distance between the centroid and nearby point, we also use the weighted sum function as is in done in SSN. However, this is not possible with the default setting in PointNet++ because the features of centroids and points have different number of dimension. We change PointNet++ so that the features of centroids and points have the same number of dimension. We re-run PointNet++ on the task of classification to confirm that this does not significantly affect performance, as can be seen in Table 1.

4. Experimental Results

In this section, we first present our reproducing results for previous SSN paper, then we report our experiments on extending the differentiable grouping module to 3D learning tasks.

Method	Accuracy
PointNet++	0.905 (std 0.002)
PointNet++ same dimensionality	0.902 (std 0.003)

Table 1. Point cloud classification experiments on ModelNet40 dataset. Accuracy is computed as the mean of 3 different runs.

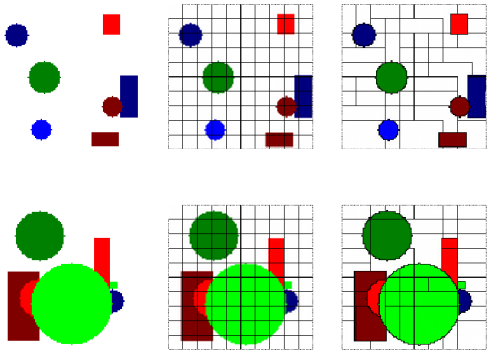

Figure 3. Toy experiments for verifying the differentiable grouping module. The first column, second column, and last column represent the input feature, initial super-pixels, and final super-pixels after 5 iterations, respectively.

4.1. Toy Examples

To verify the correctness of the differentiable SLIC module, we conduct several toy experiments. We first random generate some boxes and circles that distributed in the 2D world, where each object is represented by pure RGB colors. We assume the inputs to the differentiable SLIC module are perfect feature without any noise. We run our differentiable SLIC module directly on the input feature, and get the super-pixel ID for each input pixel. We mask the boundary between of each super-pixel (see Figure 3 for more details). The oracle experiments demonstrate that the differentiable SLIC module works very well.

4.2. Reproducing SSN Results

We re-implement the pipeline introduced in Superpixel Sampling Network (SSN) and obtained almost the same performance as SSN on the BSDS500 dataset. We use the same experimental setting as in the SSN paper, where training is performed with 100 super-pixels and testing is performed with varying number of super-pixels, usually much larger than 100. We measure the performance of our pipeline using mean Achievable Segmentation Accuracy (ASA), which represents the upper bound on the accuracy achievable by any segmentation step performed on the super-pixels. Figure 6 illustrates the performance of our re-implementation for different number of super-

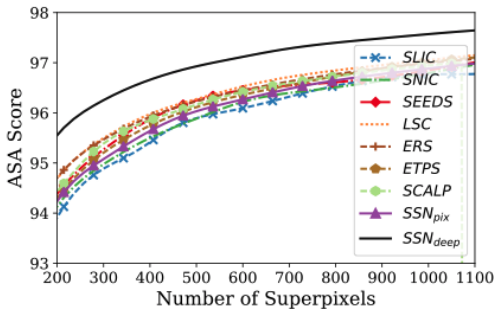

Figure 4. Figure 5 in the SSN paper. The figure demonstrates the performance of SSN as the number of superpixel increases during evaluation.

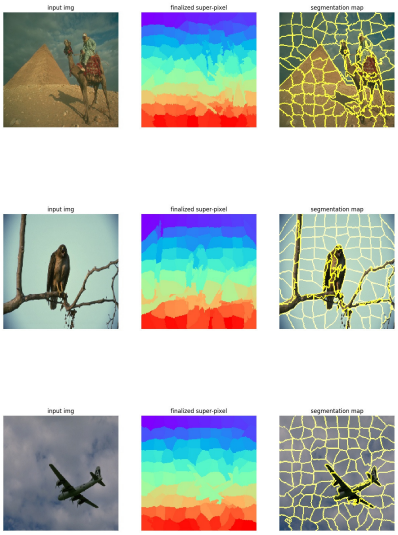

Figure 5. Super-pixel segmentation results in BSR dataset. Maximum super-pixel number is set to 100.

pixels used during evaluation. Comparing Figure 6 and Figure 4, we observe that our re-implementation obtains slightly worse performance than shown in the SSN's paper. However, we argue that this is due to small implementation details and our re-implementation preserves the main performance characteristics of SSN: as the number of superpixels used during evaluation increases, the performance also increases. Trained with the 100 superpixels setting, our re-implementation was also able to generalize to testing setting where more than 100 superpixels are used. Figure 5 shows some qualitative results of re-implemented SSN.

4.3. Point Cloud Part Segmentation

We perform the algorithm we discussed in section 3.3 into the PointNet++ [7] pipeline. We trained and tested on

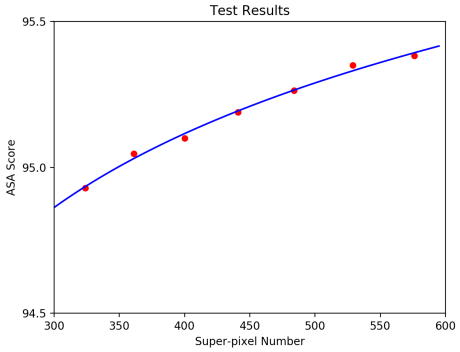

Figure 6. Test result using the SSN module trained 15,000 iterations. During the test, the slic module runs 10 times. ASA scores increases as the number of super pixels increases.

Method	Accuracy	IoU
PointNet++	93.25	83.48
Ours(iter=1)	93.30	**83.82**
Ours(iter=2)	93.19	83.43
Ours(iter=3)	**93.49**	83.71
Ours(iter=5)	93.22	83.68

Table 2. Point cloud segmentation experiments. Iter stands for the number of iteration we perform differentiable grouping in the pipeline.

the ShapeNet dataset with 16 classes and 50 different parts. The train, validation and test split follows the original split of the benchmark. We adopt Adam stochastic optimization with a batch size of 16 and decaying learning rate with initial learning rate of 0.001 and γ of 0.5. We trained for 250 epochs and preserve the best model according to the validation results. The performance of part segmentation is measured using two evaluation metrics: one is part segmentation accuracy, and the other is intersection over union (IoU). We performed the experiments on 5 models including the baseline, and tested how the iterations of differentiable grouping affects the performance. The result is as follow,

Some qualitative results are also given here. As we can see in Figure 7, our model outperforms the original PointNet++ in most joints in the shape. This is expectable, as multiple iterations of differential SLIC will group points that are similar in feature spaces, separating parts from one another much better. This can also be seen in the 2D cases where boundaries of the semantic segmentation are largely preserved in the super pixel segmentation.

5. Conclusion and Future Works

In this project we replicate previous work on differentiable super-pixel learning [5], we achieve comparable per-

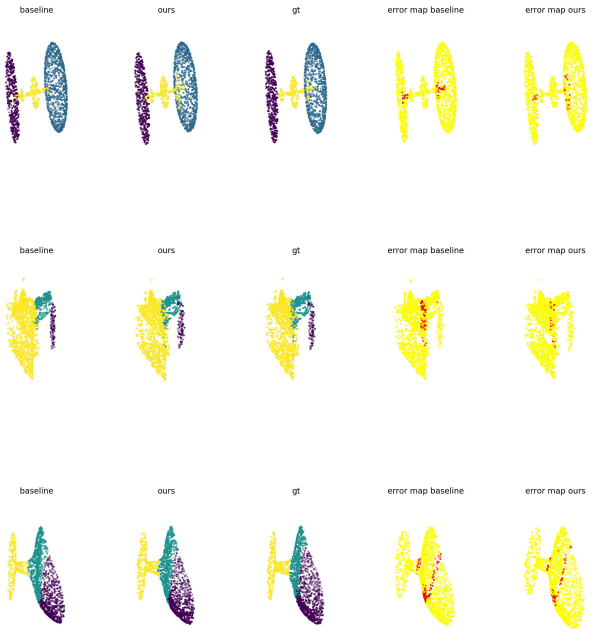

Figure 7. Test results from the ShapeNet. In the error map, red dots stand for a failure. In this test, our model performs differentiable grouping for three times in the pipeline.

formance with their official experimental setting. We further extend the idea of differentiable grouping to 3D learning tasks. Extensive experiments demonstrate the feasibility of this kind idea. For future works, we plan to generalize this framework to other 3D learning tasks, i.e., point cloud normal estimation, correspondence matching, etc.

References

[1] Raghudeep Gadde, Varun Jampani, Martin Kiefel, Daniel Kappler, and Peter V Gehler. Superpixel convolutional networks using bilateral inceptions. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016.

[2] Stephen Gould, Jim Rodgers, David Cohen, Gal Elidan, and Daphne Koller. Multi-class segmentation with relative location prior. *International Journal of Computer Vision*, 80(3):300–316, 2008.

[3] Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. In *Advances in Neural Information Processing Systems*, pages 6691–6701, 2017.

[4] Shengfeng He, Rynson WH Lau, Wenxi Liu, Zhe Huang, and Qingxiong Yang. Supercnn: A superpixelwise convolutional neural network for salient object detection. *International journal of computer vision*, 115(3):330–344, 2015.

[5] Varun Jampani, Deqing Sun, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Superpixel sampling networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 352–368, 2018.

[6] 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*, pages 652–660, 2017.

[7] Charles Ruizhongtai 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*, pages 5099–5108, 2017.

[8] Abhishek Sharma, Oncel Tuzel, and Ming-Yu Liu. Recursive context propagation network for semantic scene labeling. In *Advances in Neural Information Processing Systems*, pages 2447–2455, 2014.

[9] Guang Shu, Afshin Dehghan, and Mubarak Shah. Improving an object detector and extracting regions using superpixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3721–3727, 2013.

[10] Junjie Yan, Yinan Yu, Xiangyu Zhu, Zhen Lei, and Stan Z Li. Object detection by labeling superpixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5107–5116, 2015.

[11] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3166–3173, 2013.

[12] Fan Yang, Huchuan Lu, and Ming-Hsuan Yang. Robust superpixel tracking. *IEEE Transactions on Image Processing*, 23(4):1639–1651, 2014.