

---

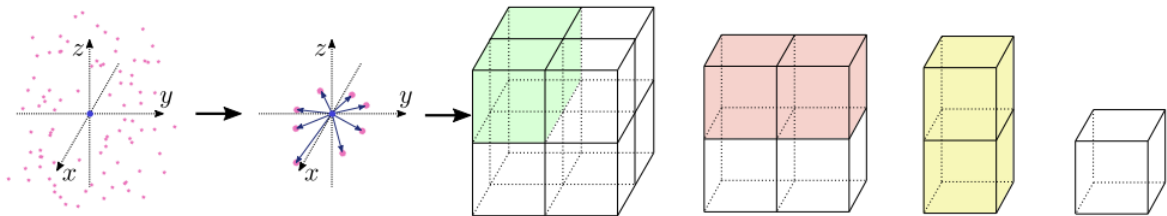
## PointSIFT: A SIFT-like Network Module for 3D Point Cloud Semantic Segmentation

Created by Mingyang Jiang, Yiran Wu, Tianqi Zhao, Zelin Zhao, Cewu Lu (corresponding author).

### Introduction

PointSIFT is a semantic segmentation framework for 3D point clouds. It is based on a simple module which extract featrues from neighbor points in eight directions. For more details, please refer to our arxiv paper.

**PointSIFT is freely available for free non-commercial use, and may be redistributed under these conditions. For commercial queries, contact Cewu Lu.**



### Installation

In our experiment, All the codes are tested in Python3.5(If you use Python 2.7, please add some system paths), CUDA 8.0 and CUDNN 5.1.

1. Install TensorFlow (We use v1.4.1).
2. Install other python libraries like *h5py*
3. Compile TF operator (Similar to PointNet++). Firstly, you should find Tensorflow include path and library paths.

```
1 import tensorflow as tf
2 # include path
3 print(tf.sysconfig.get_include())
4 # library path
5 print(tf.sysconfig.get_lib())
```

Then, change the path in all the compile file, like `tf_utils/tf_ops/sampling/tf_sampling_compile.sh`. Finally, compile the source file, we use `tf_sampling` as example.

```
1 cd tf_utils/tf_ops/sampling
2 chmod +x tf_sampling_compile.sh
3 ./tf_sampling_compile.sh
```

---

## Usage

If you want use our model in your own project. After compiling the TF operator, you can import it easily. Here shows a simple case.(we take **batch\_size \* num\_point \* input\_dim** as input and get **batch\_size \* num\_point \* output\_dim** as output)

```
1 import tensorflow as tf
2 # import our module
3 from tf_utils.pointSIFT_util import pointSIFT_module
4 # input coordinates
5 xyz = tf.placeholder(tf.float32, shape=(batch_size, num_point, 3))
6 # input features
7 point_feature = tf.placeholder(tf.float32, shape=(batch_size,
8           num_point, input_dim))
9 # setting phases
10 is_training = tf.placeholder(dtype=tf.bool, shape=())
11 # setting searching radius (0.1 as an example)
12 radius = 0.1
13 _, out_feature, _ = pointSIFT_module(xyz, point_feature, radius,
14           output_dim, is_training)
```

## Training and evaluating on ScanNet

1. All the data can be download from here. They are the same as PointNet++.
2. Train the data:

```
1 python train_and_eval_scannet.py
```

If you have multiple GPU:

```
1 CUDA_VISIBLE_DEVICES=0,1,2,3 python train_and_eval_scannet.py --gpu_num
   =4
```

## Citation

Please cite the paper in your publications if it helps your research:

```
1 @misc{1807.00652,
2   Author = {Mingyang Jiang and Yiran Wu and Tianqi Zhao and Zelin Zhao
3     and Cewu Lu},
4   Title = {PointSIFT: A SIFT-like Network Module for 3D Point Cloud
5     Semantic Segmentation},
6   Year = {2018},
7   Eprint = {arXiv:1807.00652},
8 }
```