

Installation and training guidelines for the PVKD codes

The PVKD codes are based on the OpenPCDet codebase.

For **SemanticKITTI**:

The following installation procedure works on our A100 server.

1) install torch

Install torch-1.10.0, cuda-11.3

spcov: pip install spconv_cu113

```
pip install nuscenes-devkit pyyaml Cython numba opencv-python torchpack strictyaml  
lxmlite easydict scikit-image tqdm SharedArray prettytable
```

```
conda install -c conda-forge mpi4py openmpi
```

Install torch_scatter:

```
pip install torch-scatter -f https://data.pyg.org/whl/torch-1.10.0+cu113.html
```

2) install openpcdet

Move to code directory, then:

```
python setup.py develop
```

3) install torchsparse in the cluster

1. Use gcc -v to check the version of gcc, 5.4.0

2. [GitHub - sparsehash/sparsehash: C++ associative containers](#) upload the code to the cluster and then unzip the file

3. create directory to store the compiling files: mkdir torchsparse_dir. Then, move to the directory in step 2

4. ./configure --prefix=/mnt/lustre/liuyouquan.vendor/torchsparse_dir/Sparsehash/ ;

5. make

6. make install

7. configuration path of .bashrc, shown as follows

```
export PATH=/mnt/lustre/liuyouquan.vendor/cmake-3.14.0-Linux-x86_64/bin:$PATH  
export CC=/mnt/lustre/share/gcc/gcc-5.4/bin/gcc  
export CXX=/mnt/lustre/share/gcc/gcc-5.4/bin/g++  
export CUDA_HOME=/mnt/lustre/share/cuda-10.2/  
export PATH=/mnt/lustre/share/gcc/gcc-5.4/bin:/mnt/lustre/share/cuda-10.2/bin:$PATH  
export LD_LIBRARY_PATH=/mnt/lustre/share/cuda-10.2/lib64:/mnt/lustre/share/boost/lib/:$LD_LIBRARY_PATH  
export CPLUS_INCLUDE_PATH=/mnt/lustre/share/boost/include/:$CPLUS_INCLUDE_PATH  
export PATH=$HOME/bin:$PATH  
export CPLUS_INCLUDE_PATH=/mnt/lustre/liuyouquan.vendor/torchsparse_dir/Sparsehash/include/:$CPLUS_INCLUDE_PATH  
export LD_LIBRARY_PATH=/mnt/lustre/liuyouquan.vendor/torchsparse_dir/Sparsehash/lib/:$LD_LIBRARY_PATH  
export PATH=/mnt/lustre/liuyouquan.vendor/0openmpi/bin:$PATH  
export LD_LIBRARY_PATH=/mnt/lustre/liuyouquan.vendor/0openmpi/lib/:$LD_LIBRARY_PATH  
export CPLUS_INCLUDE_PATH=/mnt/lustre/liuyouquan.vendor/torchsparse_dir/sparsehash-master/:$CPLUS_INCLUDE_PATH
```

8. [GitHub - mit-han-lab/torchsparse: A high-performance neural network library for point cloud processing.](#)

9. pip install --upgrade git+https://github.com/mit-han-lab/torchsparse.git@v1.4.0
(We have only performed experiments in TorchSparse 1.4.0. Using other versions may have different results.)

You may meet the following error:

```
error: invalid static_cast from type 'const torch::OrderedDict<std::basic_string<char>,  
std::shared_ptr<torch::nn::Module>>' to type  
'torch::OrderedDict<std::basic_string<char>, std::shared_ptr<torch::nn::Module> >&'
```

Solution: [\[BUG\]关于Pytorch中CUDA扩展的本地安装 - 知乎 \(zhihu.com\)](#)

Then:

```
pip install ./torchsparse
```

You can import torchsparse to check if torchsparse is installed successfully.

4) install point transformer

```
cd lib/pointops
python setup.py install
```

5) install Range Lib

```
cd pcdet/models/segmentors/rpvnet/models/semantic/range_lib
python setup.py install
```

6) train the model

Here, we take Cylinder3D as example. If you want to train SPVNAS (actually SPVCNN) or MinkowskiNet, you can just change the yaml file to spvcnn_mk18_cr10.yaml (teacher SPVCNN), spvcnn_mk18_cr5.yaml (student SPVCNN), minkunet_mk18_cr10.yaml (teacher MinkowskiNet) or minkunet_mk18_cr5.yaml (student MinkowskiNet). And remove the `--is_cyl` flag when training these non-Cylinder3D models. Here, `tools/cfgs_seg/semantickitti_models/` stores the yaml files of Cylinder3D, SPVCNN and MinkowskiNet. Remember to change the `DATA_PATH` in these yaml files which stores the SemanticKITTI dataset.

1. Train the student and the teacher on train set for 12 epochs: (performance on val set)

```
cd tools
sh scripts/slurm_train.sh shlab_adg_2 cyl_1.0x 2 --cfg_file
cfgs_seg/semantickitti_models/cylinder_cy480.yaml --is_cyl --epochs 12 # teacher
sh scripts/slurm_train.sh shlab_adg_2 cyl_0.5x 2 --cfg_file
cfgs_seg/semantickitti_models/cylinder_cy480_0.5X.yaml --is_cyl --epochs 12 # student
```

2. Load the trained weight and train the student with PVKD on train set for 36 epochs: (performance on val set)

```
pip install numpy_indexed
sh scripts/slurm_train_distil.sh shlab_adg_2 c-kd-train 2 --cfg_file
cfgs_seg/semantickitti_models/cylinder_cy480.yaml --pretrained_model
/path/to/student_model/checkpoint_epoch_12.pth --pretrained_model_t
/path/to/teacher_model/checkpoint_epoch_12.pth --epochs 36 --is_cyl
```

3. Load the trained weight and train the student on train+val set for 14 epoch:

Add '08' to the self.seqs (Line 31) of `pcdet/datasets_seg/SemanticKITTI/semantickitti.py` to train on train+val set

```
sh scripts/slurm_train.sh shlab_adg_2 cyl_0.5x 2 \
--cfg_file cfgs_seg/semantickitti_models/cylinder_cy480_0.5x.yaml \
--is_cyl --epochs 14 \
--pretrained_model /path/to/student_model/checkpoint_epoch_xx.pth
```

4. Load weight and then TTA, then submit to the online server: (performance on test set)

```
sh scripts/slurm_test_tta.sh shlab_adg_2 cyl_0.5x 1 --cfg_file
cfgs_seg/semantickitti_models/cylinder_cy480_0.5x.yaml --save_path out_cyl/test/
--pretrained_model path/to/checkpoint_epoch_14.pth
```

Note: If you do not want to use TTA, you can just comment Line 324-325 in `tools/test_seg_tta.py`.

5. Convert label number back to the original dataset format before submitting:

```
python remap_semantic_labels.py -p out_cyl/test -s test --inverse
cd out_cyl/test
zip -r out_cyl.zip sequences/
```

6. Finally, upload `out_cyl.zip` to the SemanticKITTI online server. (You need to create an account first before submitting the predictions. Regarding how to create an

account, you can go to the official site of SemantickITTI and follow the registration guidelines.)