香港中文大學
The Chinese University of Hong Kong

OpenPCDet: An Open-source Toolbox for 3D Object Detection from Point Clouds

Shaoshuai Shi
Ph.D. Student of Multimedia Laboratory
The Chinese University of Hong Kong

Nov. 30, 2020

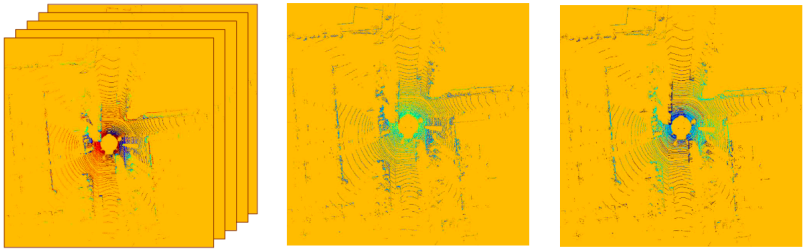
Outline

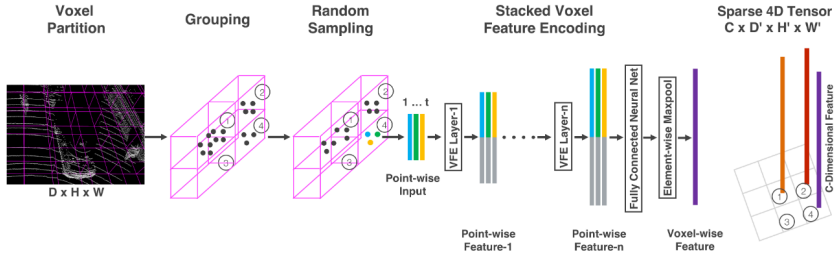
- Introduction to 3D object detection from point clouds
  - Feature learning from point clouds (voxel-based vs. point-based)
  - 3D box generation (anchor-based vs. anchor-free)
  - Two-stage detector with 3D box refinement
- OpenPCDet architecture and experiments

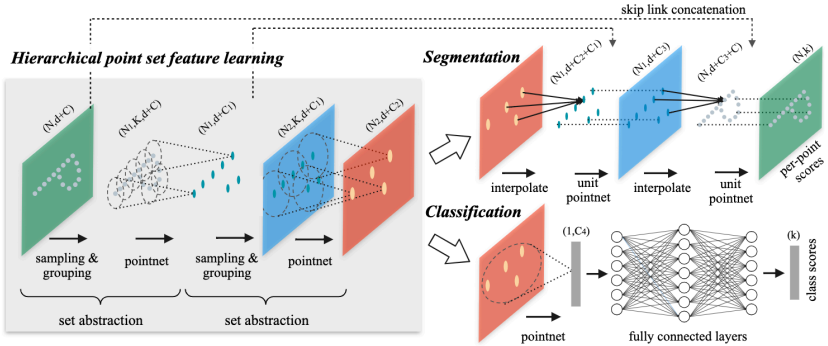
Feature learning from point clouds

香港中文大學
The Chinese University of Hong Kong

- 2D CNN with bird's view representation
- 3D Voxel CNN (or sparse conv) with Voxel Feature Encoding (VFE)
- PointNet++ series
- Combination of the above strategies





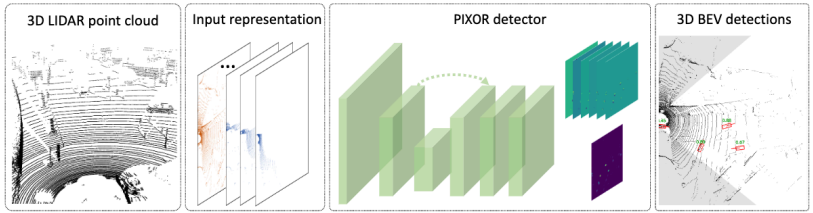


Reference
1. Xiaozhi Chen, et al., Multi-View 3D Object Detection Network for Autonomous Driving
2. Yin Zhou, et al., VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection
3. Charles Qi, et al., PointNet++: Deep Hierarchical Feature Learning onPoint Sets in a Metric Space

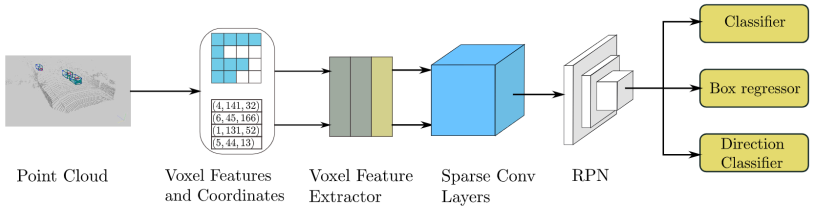
Anchor-based 3D box generation

香港中文大學
The Chinese University of Hong Kong

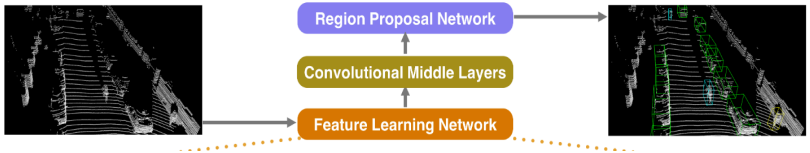
- Generate 3D boxes by selecting and tuning pre-defined 3D anchors on the bird-view 2D feature maps



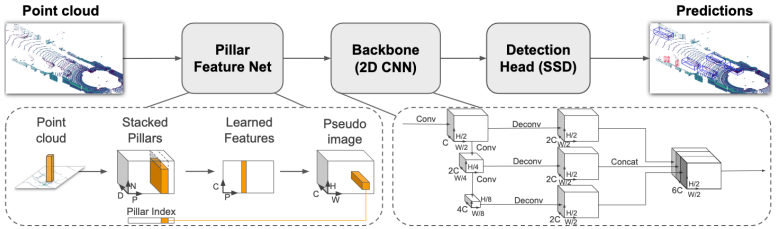
PIXOR



SECOND



VoxelNet



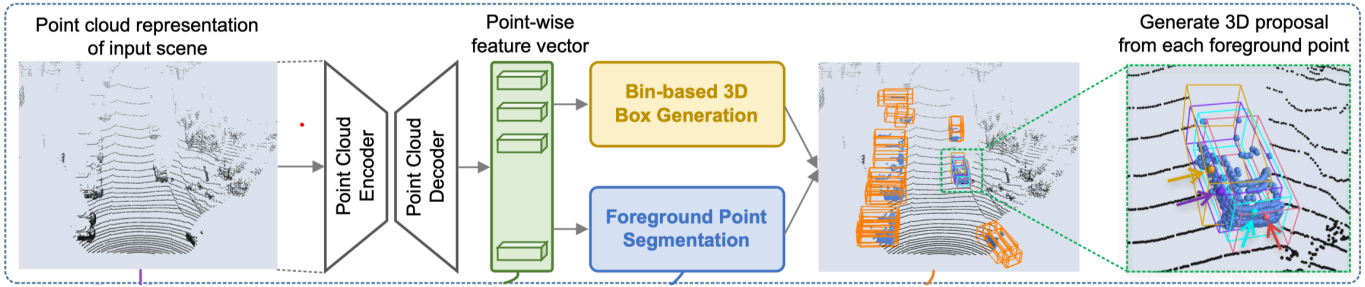
PointPillar

Reference
1. Yan Yan, et al., SECOND: Sparsely EmbeddedConvolutional Detection
2. Bin Yang, et al., PIXOR: Real-time 3D Object Detection from Point Clouds
3. Alex H. Lang, et al., PointPillars: Fast Encoders for Object Detection from Point Clouds
4. Yin Zhou, et al., VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection

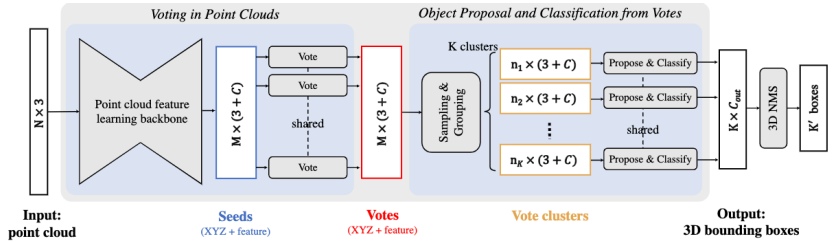
Anchor-free 3D box generation

香港中文大學
The Chinese University of Hong Kong

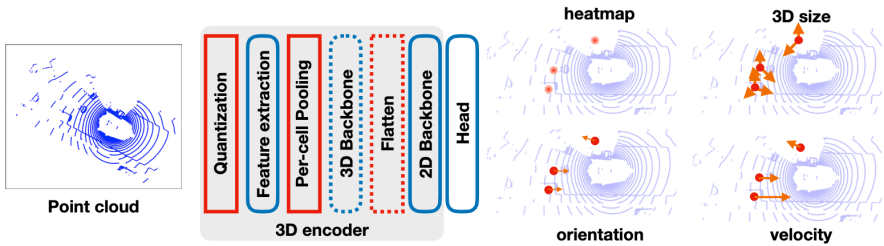
- Generating 3D box without predefined dense anchors



PointRCNN



VoteNet



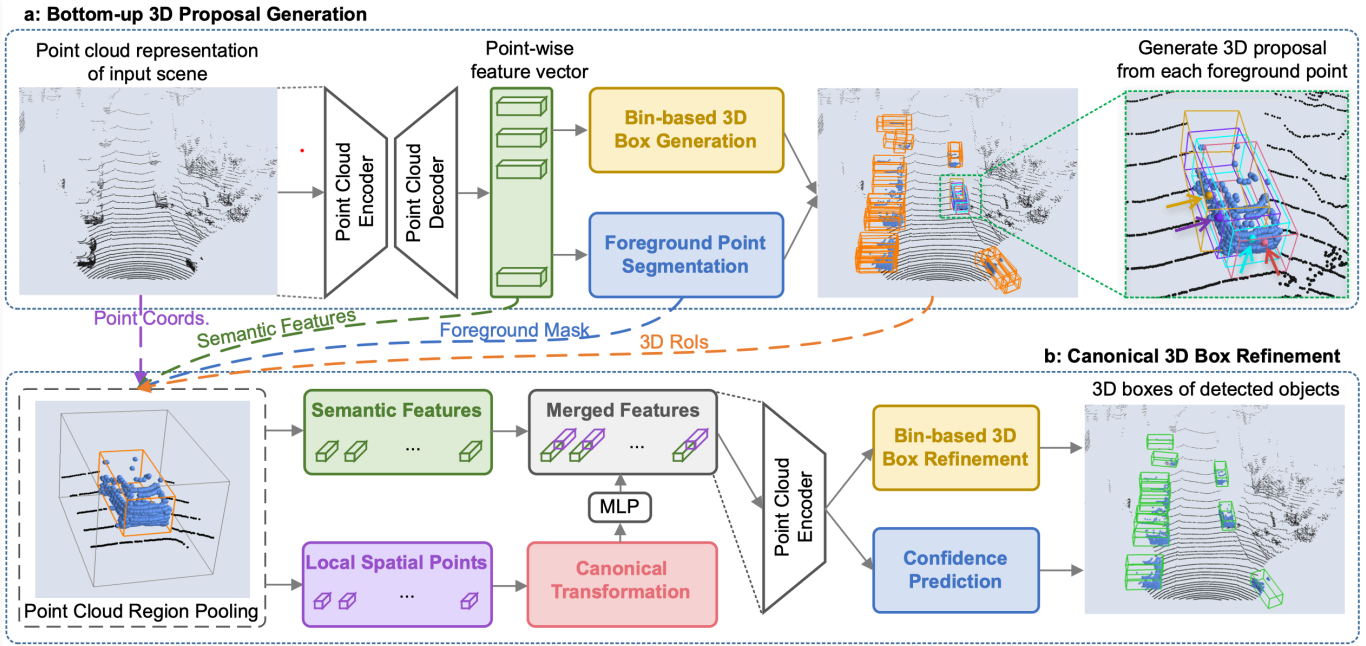
CenterPoint

Reference
1. Shaoshuai Shi, et al., PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud
2. Charles Qi, et al., Deep Hough Voting for 3D Object Detection in Point Clouds
3. Tianwei Yin, et al., Center-based 3D Object Detection and Tracking

Two-stage detector with 3D box refinement

香港中文大學
The Chinese University of Hong Kong

- PointRCNN: 3D box refinement with point cloud region pooling

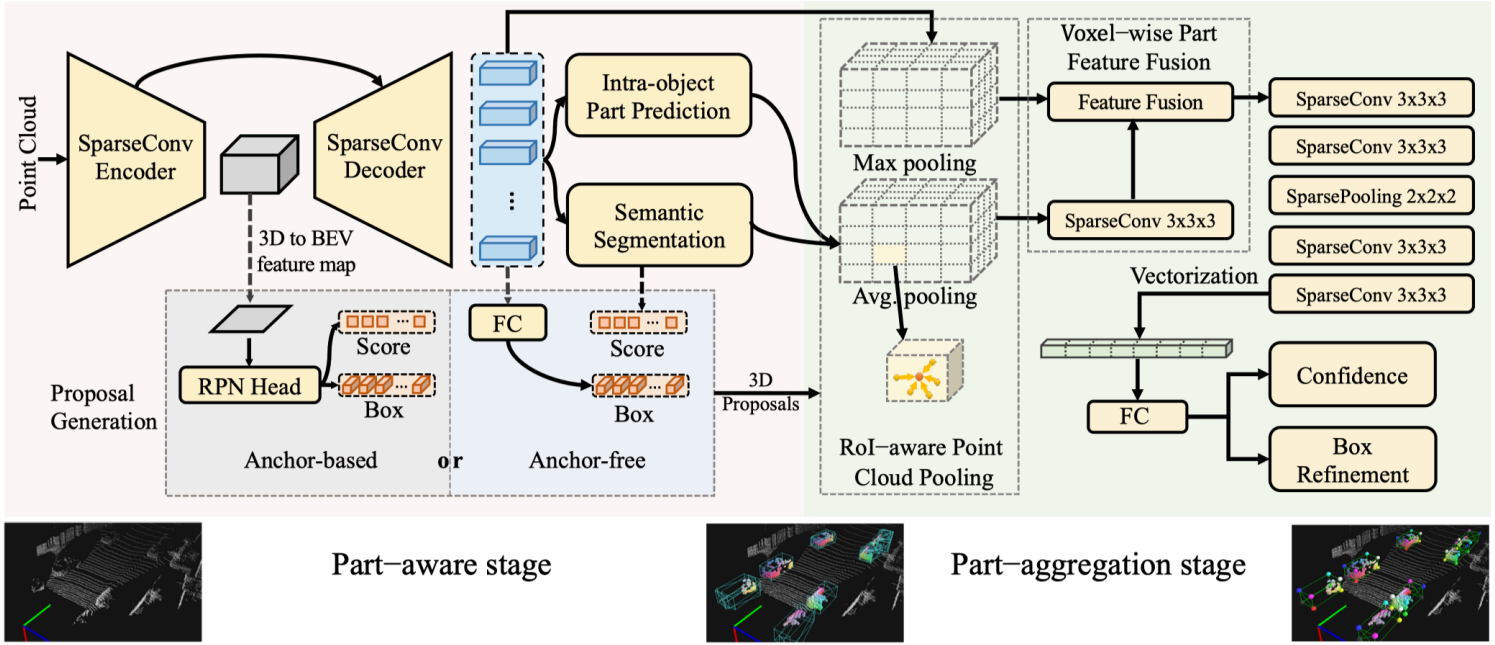


Reference
1. Shaoshuai Shi, et al., PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud

Two-stage detector with 3D box refinement

香港中文大學
The Chinese University of Hong Kong

- Part-A2-Net: 3D box refinement with RoI-aware point cloud pooling

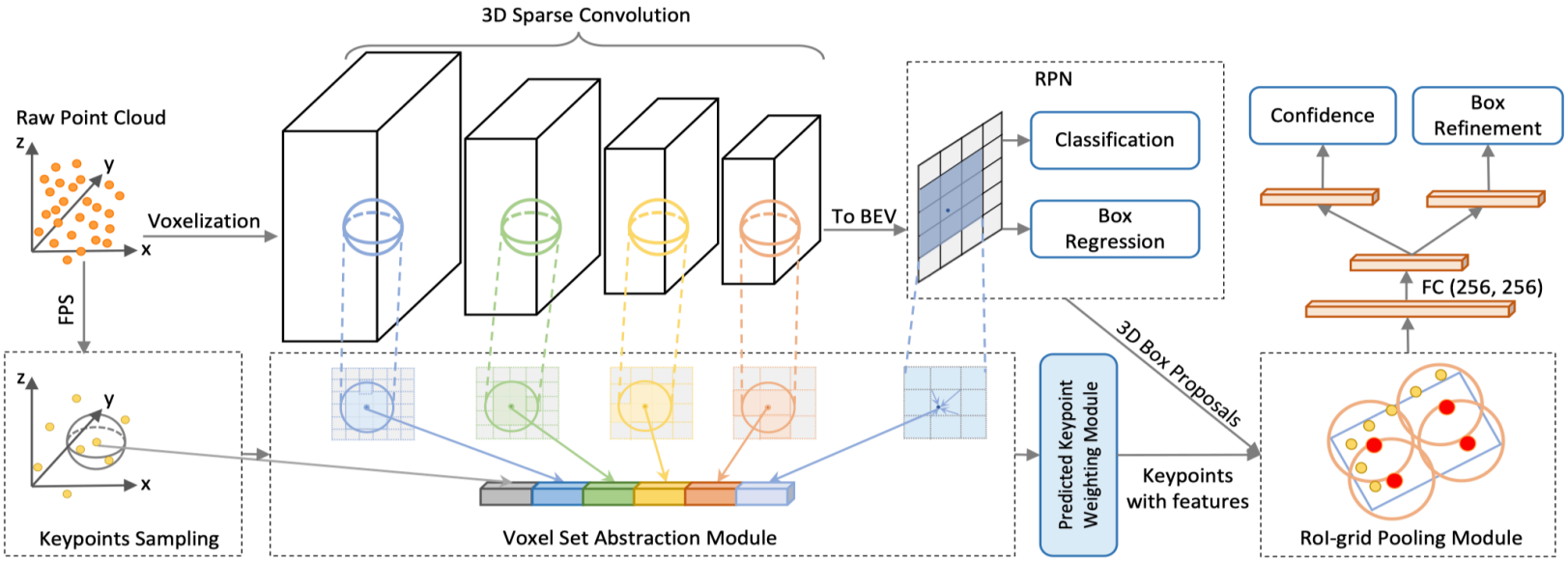


Reference
1. Shaoshuai Shi, et al., From Points to Parts: 3D Object Detection from Point Cloud with Part-aware and Part-aggregation Network

Two-stage detector with 3D box refinement

香港中文大學
The Chinese University of Hong Kong

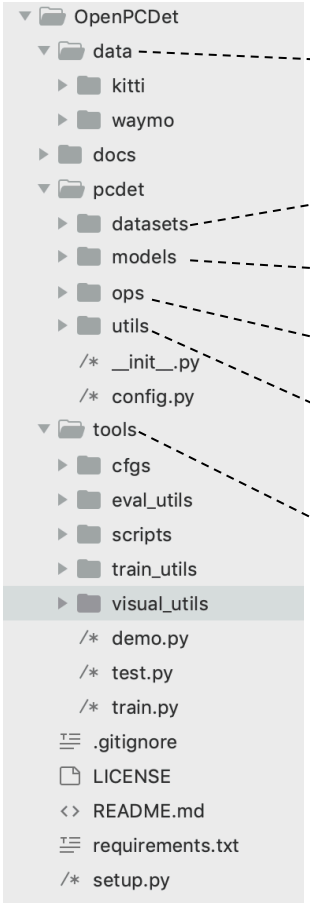
- PV-RCNN: 3D box refinement with RoI-grid pooling



Reference
1. Shaoshuai Shi, et al., PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection

OpenPCDet: 3D Detection Toolbox

香港中文大學
The Chinese University of Hong Kong



Store the data and train/val splits

Data processing/loader and dataset-specific evaluation tools

Model definition and forward/backward codes

Customized operations (C++ / CUDA codes) for 3D detection

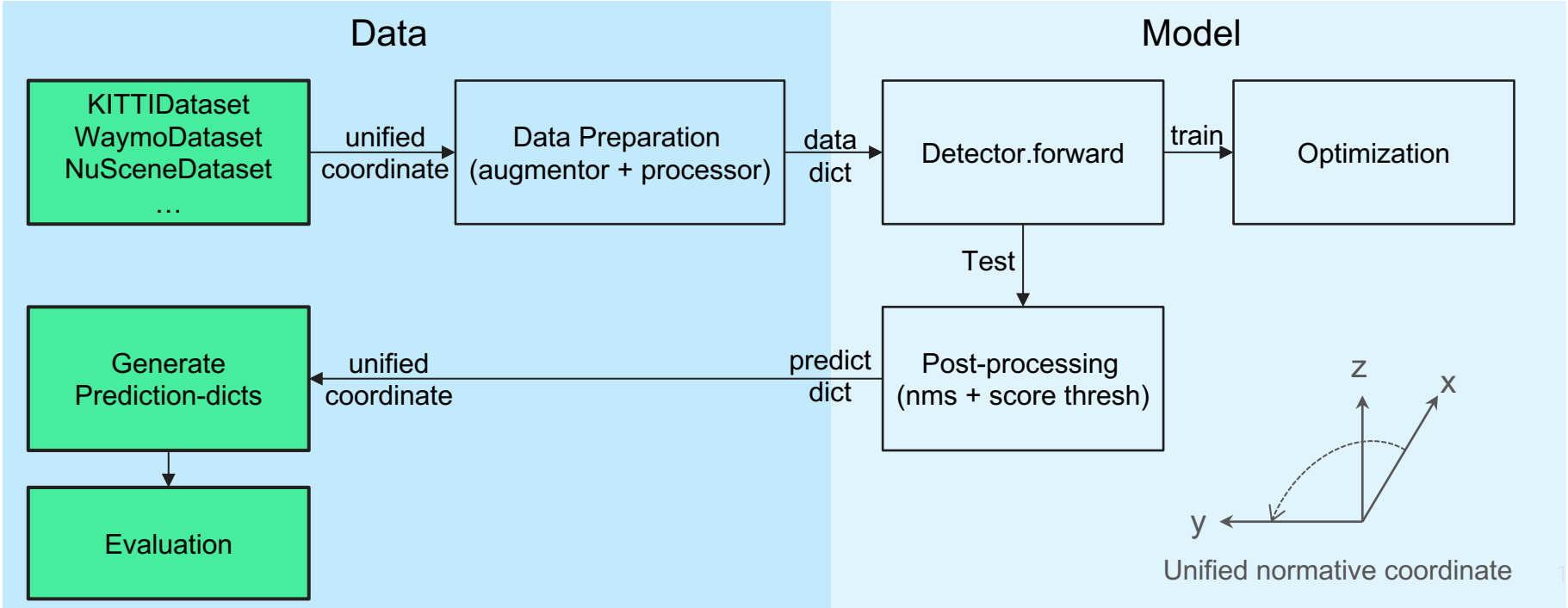
Common utilized codes and loss utilized codes for 3D detection

Training / evaluation / visualization tools and configuration files

OpenPCDet: Architecture

香港中文大學
The Chinese University of Hong Kong

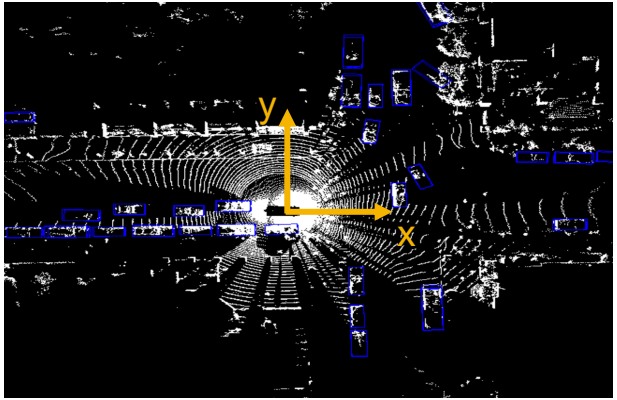
- Data-Model separation with unified coordinate and box definition across different datasets

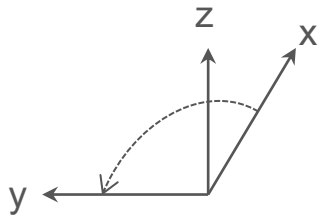


OpenPCDet: Unified coordinate and 3D box definition

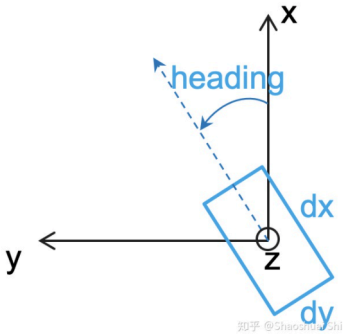
香港中文大學
The Chinese University of Hong Kong

- Unified coordinate:
  - x-axis points towards the front
  - y-axis points towards the left
  - z-axis points towards the up direction
- Unified 3D box definition:
  - (cx, cy, cz, dx, dy, dz, heading)
  - Heading angle: the angle from x-axis to y-axis (heading = 0 for the direction of x-axis)





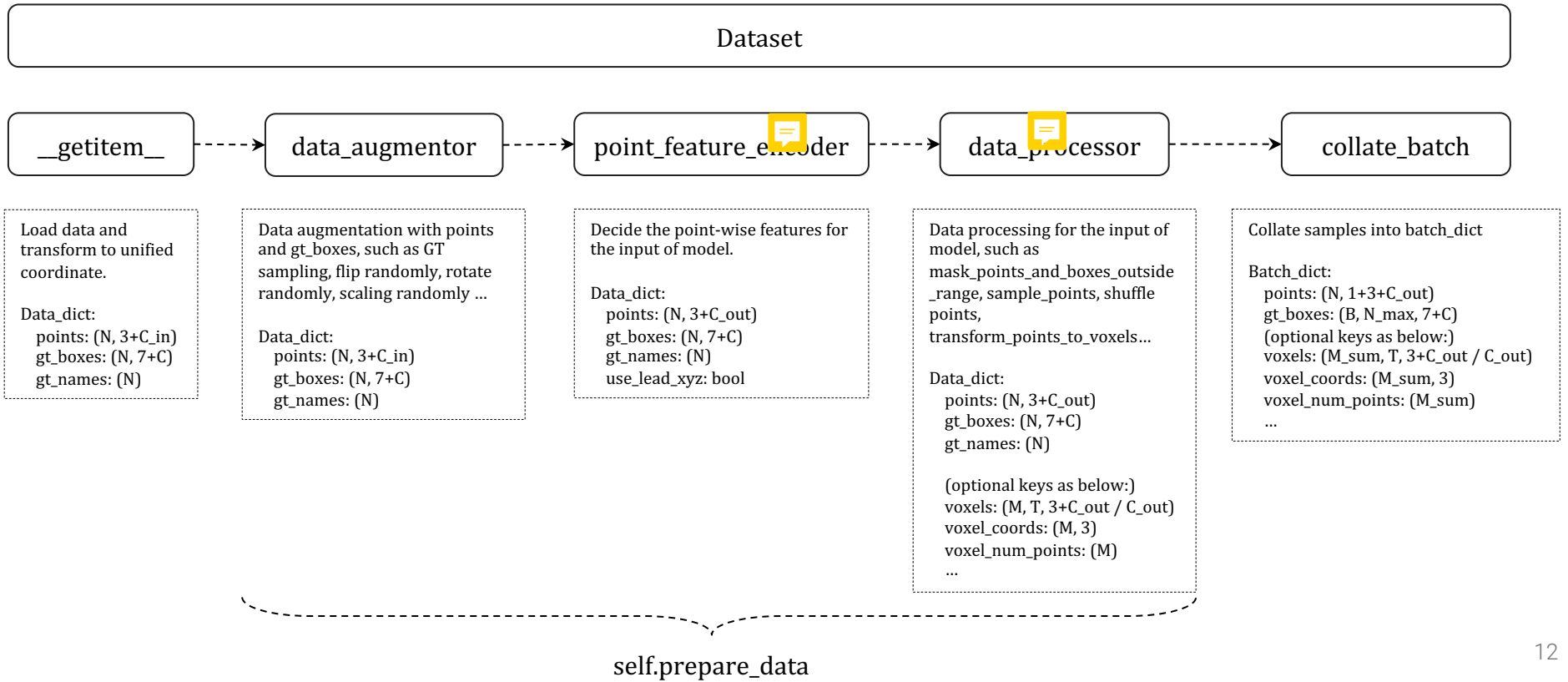
Unified normative coordinate



Unified 3D box definition

OpenPCDet: Data flow

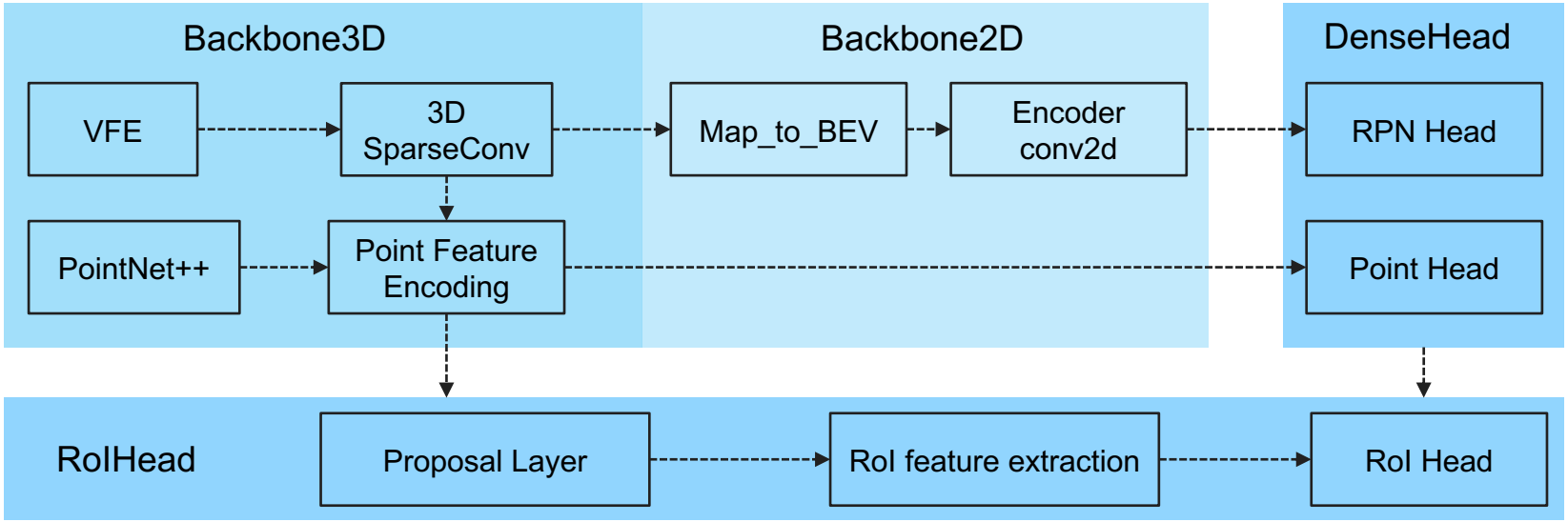
香港中文大學
The Chinese University of Hong Kong



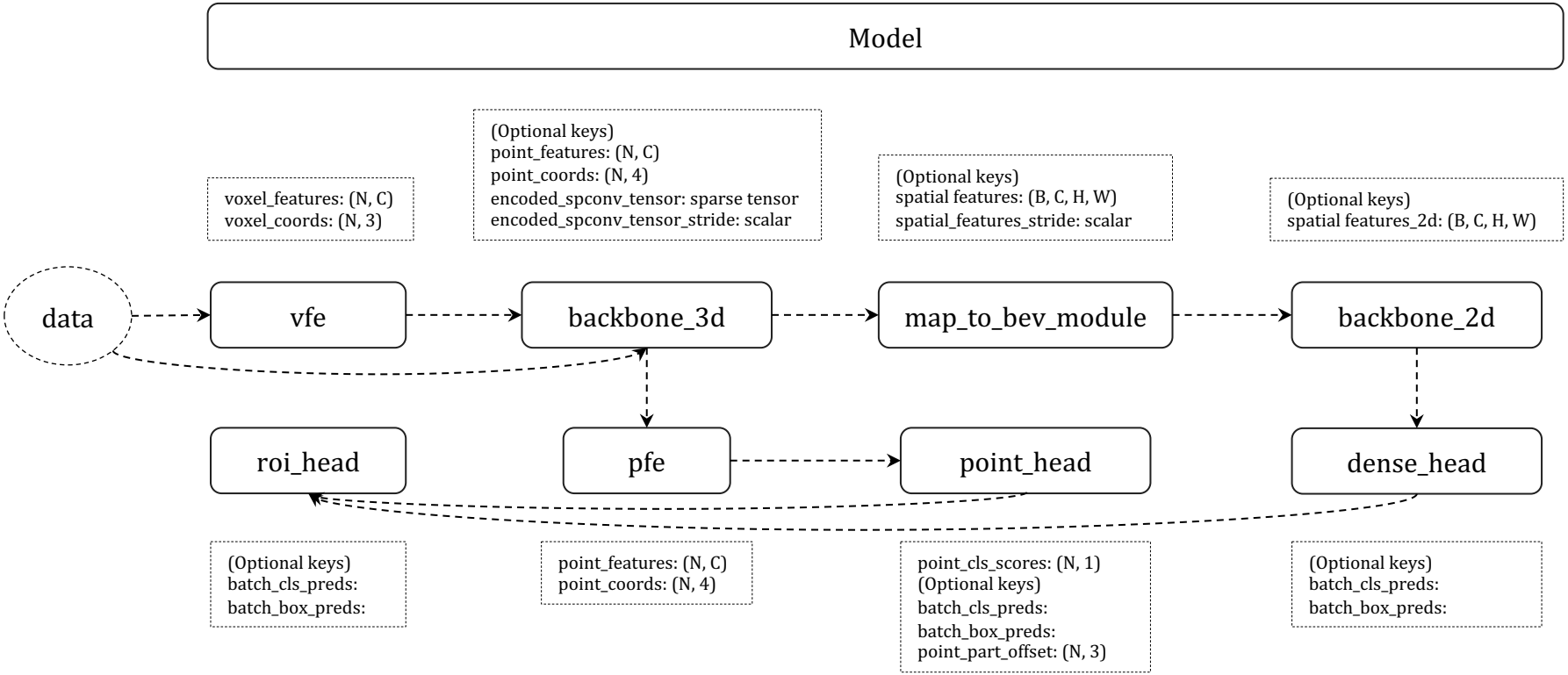
OpenPCDet: Modularization

香港中文大學
The Chinese University of Hong Kong

- Modularization to support various 3D detectors within one framework
  - Backbone: 3D Voxel CNN / Sparse Conv / PointNet++ / 2D CNN
  - DenseHead: Anchor-based / Anchor-free / Point-based
  - RoIHead: proposal generation / RoI Pooling / proposal target assigner



OpenPCDet: Model topology



OpenPCDet: Build detector with configuration dict

香港中文大學
The Chinese University of Hong Kong

- Build your detector with customized configs
  - Detector3DTemplate.build_networks()

```
self.module_topology = [
    'vfe', 'backbone_3d', 'map_to_bev_module', 'pfe',
    'backbone_2d', 'dense_head',  'point_head', 'roi_head'
]
```

```
def build_networks(self):
    model_info_dict = {
        'module_list': [],
        'num_rawpoint_features': self.dataset.point_feature_encoder.num_point_features,
        'num_point_features': self.dataset.point_feature_encoder.num_point_features,
        'grid_size': self.dataset.grid_size,
        'point_cloud_range': self.dataset.point_cloud_range,
        'voxel_size': self.dataset.voxel_size
    }
    for module_name in self.module_topology:
        module, model_info_dict = getattr(self, 'build_%s' % module_name)(
            model_info_dict=model_info_dict
        )
        self.add_module(module_name, module)
    return model_info_dict['module_list']
```

OpenPCDet: Model forward & optimization

香港中文大學
The Chinese University of Hong Kong

- Call each module sequentially with their topology
- Optimization (Calculate the losses within each head):
  - DETECTOR.get_training_loss()
  - HEAD.get_loss()

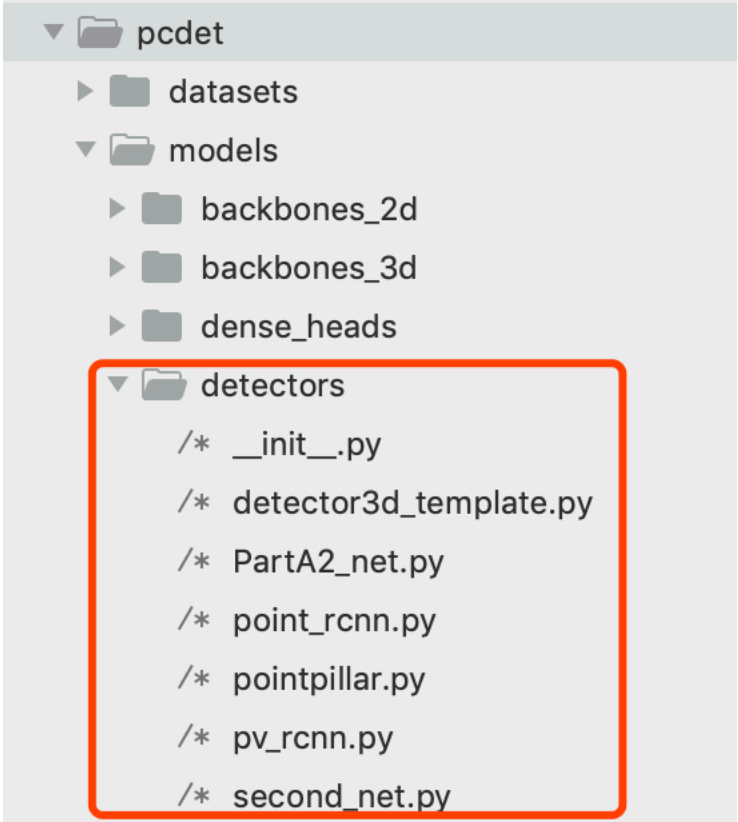
```python
def forward(self, batch_dict):
    batch_dict = self.vfe(batch_dict)
    batch_dict = self.backbone_3d(batch_dict)
    batch_dict = self.map_to_bev_module(batch_dict)
    batch_dict = self.backbone_2d(batch_dict)
    batch_dict = self.dense_head(batch_dict)
```

```python
from .detector3d_template import Detector3DTemplate


class PointRCNN(Detector3DTemplate):
    def __init__(self, model_cfg, num_class, dataset):
        super().__init__(model_cfg=model_cfg, num_class=num_class, dataset=dataset)
        self.module_list = self.build_networks()

    def forward(self, batch_dict):
        for cur_module in self.module_list:
            batch_dict = cur_module(batch_dict)

        if self.training:
            loss, tb_dict, disp_dict = self.get_training_loss()

            ret_dict = {
                'loss': loss
            }
            return ret_dict, tb_dict, disp_dict
        else:
            pred_dicts, recall_dicts = self.post_processing(batch_dict)
            return pred_dicts, recall_dicts

    def get_training_loss(self):
        disp_dict = {}
        loss_point, tb_dict = self.point_head.get_loss()
        loss_rcnn, tb_dict = self.roi_head.get_loss(tb_dict)

        loss = loss_point + loss_rcnn
        return loss, tb_dict, disp_dict

```

OpenPCDet: DetectorTemplate

香港中文大學
The Chinese University of Hong Kong

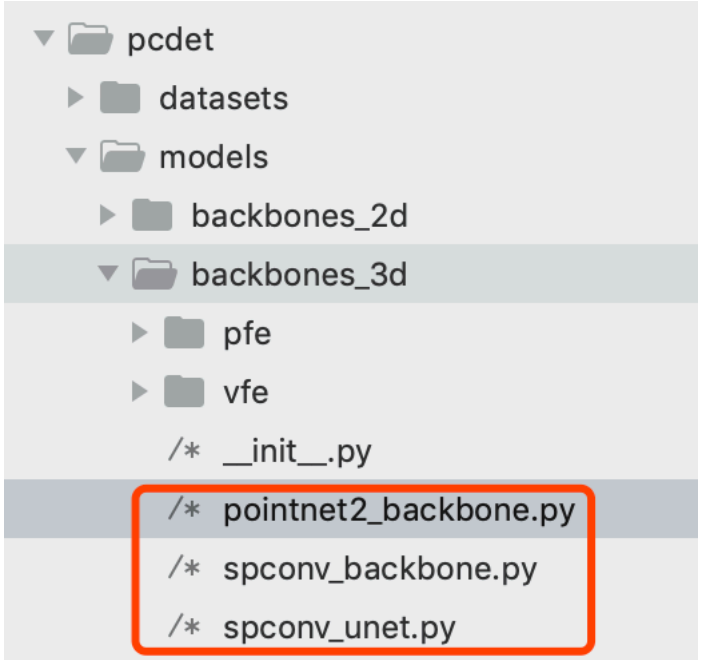
- Task: The top module for a detector
  - Build_networks
  - Forward
  - Loss calculation
  - Post_processing (NMS + score threshold)



OpenPCDet: 3D backbone networks

- Task: extract voxel-wise / point-wise features

- 3D encoder with sparse convolution (with VFE):
  - VoxelBackBone8x
  - VoxelResBackBone8x
- 3D UNet with sparse convolution (with VFE):
  - UNetV2
- Point-wise networks (PointNet++)
  - PointNet2MSG

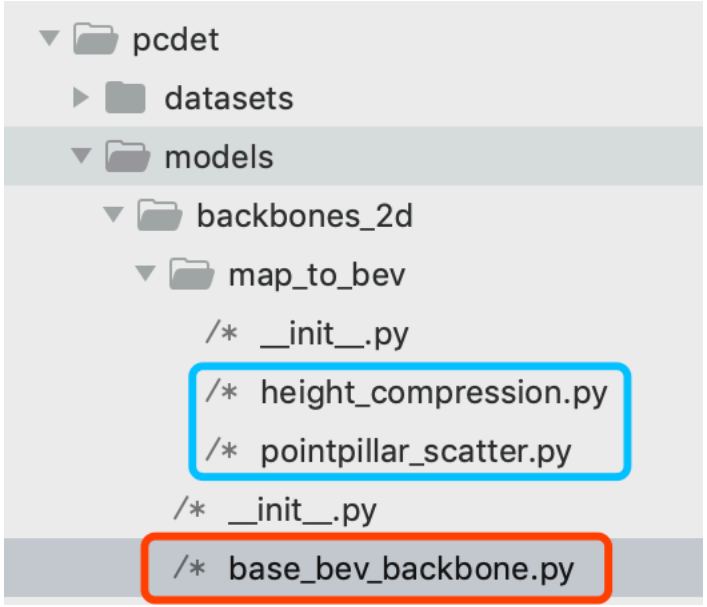


OpenPCDet: 2D backbone networks

香港中文大學
The Chinese University of Hong Kong

- Task: extract 2D feature maps
  - (with map_to_bev_module)

- Map_to_bev_module (map 3D features to 2D maps):
  - HeightCompression
  - PointPillarScatter
- 2D convolution encoder with FPN-like upsampling
  - BaseBEVBackbone

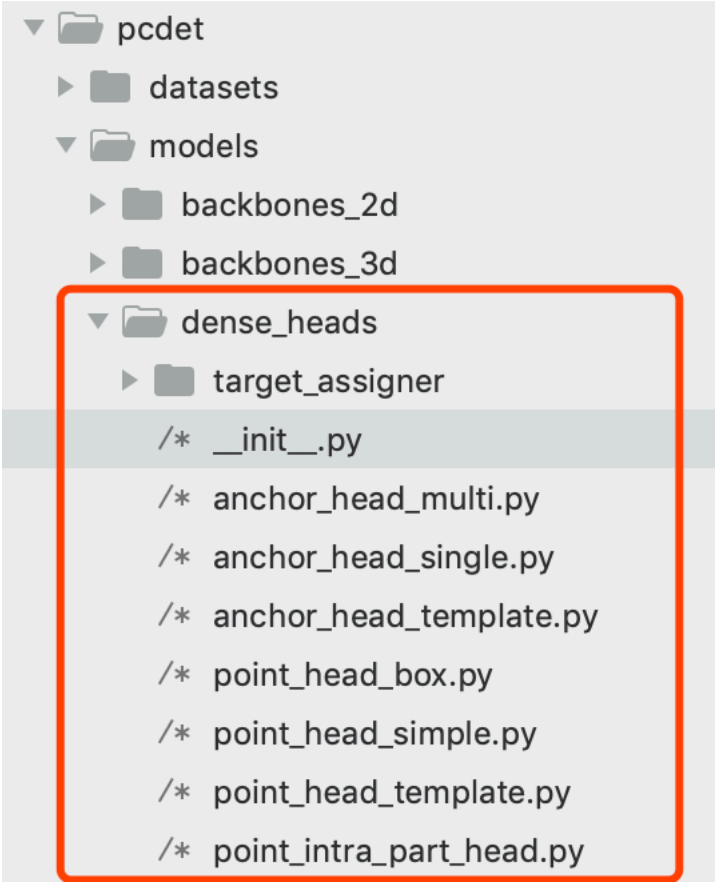


OpenPCDet: Denseheads

香港中文大學
The Chinese University of Hong Kong

- Task: Generate dense 3D boxes
  - Target assigning
  - Head loss calculation

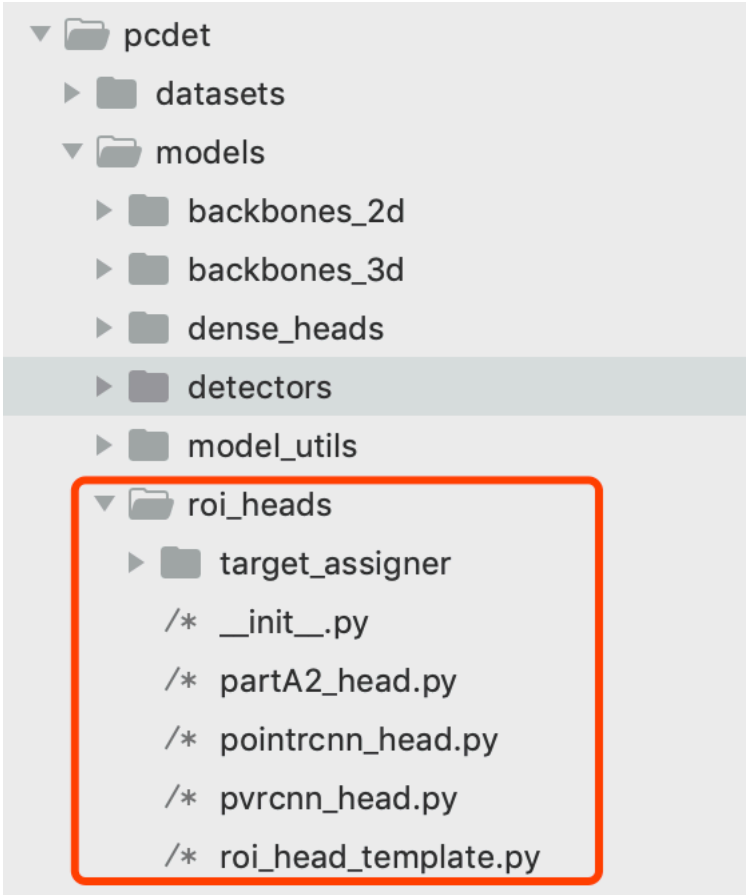
- Dense head with BEV features (AnchorHeadTemplate):
  - AnchorHeadSingle
  - AnchorHeadMulti
  - CenterHead (anchor-free)
- Dense head with point-wise features (PointHeadTemplate):
  - PointHeadSimple
  - PointHeadBox
  - PointIntraPartOffsetHead



OpenPCDet: RoIHeads

- Task: Refine 3D proposals with RoI-aligned features
  - Extract RoI-aligned features
  - proposal_layer
  - ProposalTargetLayer
  - Head loss calculation

- Stage-II RoI refinement (RoIHeadTemplate):
  - PointRCNNHead
  - PartA2FCHead
  - PVRCNNHead



OpenPCDet: Configuration files

香港中文大學
The Chinese University of Hong Kong

- Hierarchical configuration with YAML file
- Each model takes its own config

```yaml
CLASS_NAMES: ['Car', 'Pedestrian', 'Cyclist']

DATA_CONFIG:
    _BASE_CONFIG_: cfgs/dataset_configs/kitti_dataset.yaml


MODEL:
    NAME: SECONDNet

    VFE:
        NAME: MeanVFE

    BACKBONE_3D:
        NAME: VoxelBackBone8x

    MAP_TO_BEV: ...

    BACKBONE_2D: ...

    DENSE_HEAD: ...

    POST_PROCESSING: ...


OPTIMIZATION: ...
```

```python
class DatasetTemplate(torch_data.Dataset):
    def __init__(self, dataset_cfg=None, class_names=None, training=True, root_path=None, logger=None):
        super().__init__()
```

```python
class Detector3DTemplate(nn.Module):
    def __init__(self, model_cfg, num_class, dataset):
```

```python
class AnchorHeadTemplate(nn.Module):
    def __init__(self, model_cfg, num_class, class_names, grid_size,
```

OpenPCDet example: Build one-stage detector

香港中文大學
The Chinese University of Hong Kong

- SECOND / PointPillar

```yaml
CLASS_NAMES: ['Car', 'Pedestrian', 'Cyclist']

DATA_CONFIG:
    _BASE_CONFIG_: cfgs/dataset_configs/kitti_dataset.yaml


MODEL:
    NAME: SECONDNet

    VFE:
        NAME: MeanVFE

    BACKBONE_3D:
        NAME: VoxelBackBone8x

    MAP_TO_BEV: ...

    BACKBONE_2D: ...

    DENSE_HEAD: ...

    POST_PROCESSING: ...


OPTIMIZATION: ...
```

SECOND

```yaml
CLASS_NAMES: ['Car', 'Pedestrian', 'Cyclist']

DATA_CONFIG: ...

MODEL:
    NAME: PointPillar

    VFE: ...

    MAP_TO_BEV: ...

    BACKBONE_2D: ...

    DENSE_HEAD: ...

    POST_PROCESSING: ...


OPTIMIZATION: ...
```

PointPillar

OpenPCDet example: Build two-stage detector

香港中文大學
The Chinese University of Hong Kong

- PointRCNN / PV-RCNN

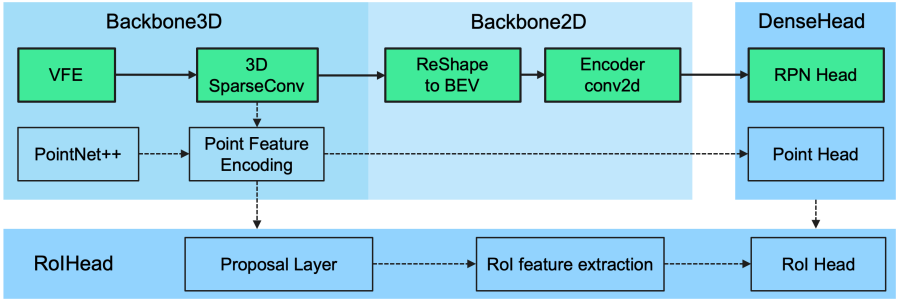
```
CLASS_NAMES: ['Car', 'Pedestrian', 'Cyclist']

DATA_CONFIG: ...

MODEL:
    NAME: PointRCNN

    BACKBONE_3D: ...

    POINT_HEAD: ...

    ROI_HEAD: ...

    POST_PROCESSING: ...


OPTIMIZATION: ...
```

PointRCNN

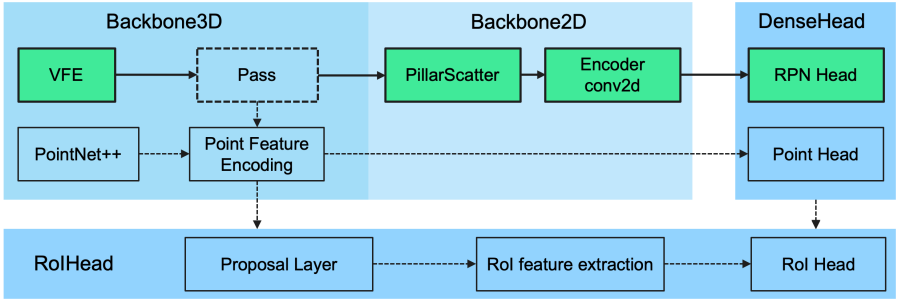
```
CLASS_NAMES: ['Car', 'Pedestrian', 'Cyclist']

DATA_CONFIG: ...

MODEL:
    NAME: PVRCNN

    VFE:
        NAME: MeanVFE

    BACKBONE_3D:
        NAME: VoxelBackBone8x

    MAP_TO_BEV: ...

    BACKBONE_2D: ...

    DENSE_HEAD: ...

    PFE: ...

    POINT_HEAD: ...

    ROI_HEAD: ...

    POST_PROCESSING: ...


OPTIMIZATION: ...
```

PV-RCNN

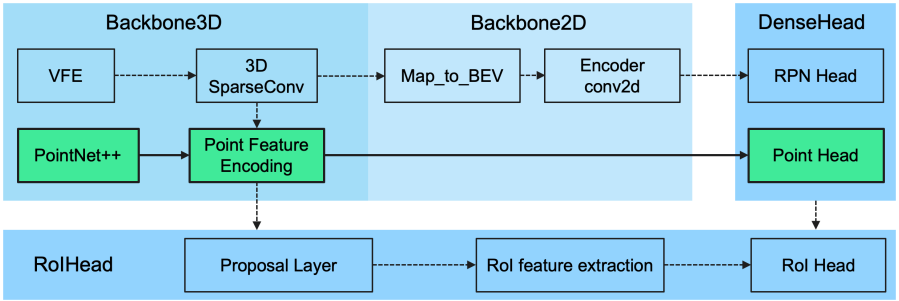
VoxelNet / SECOND / VoxelFPN



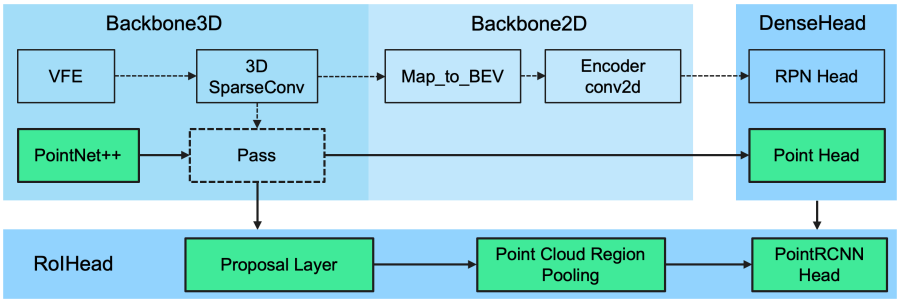
PointPillar / PIXOR / CenterPoint



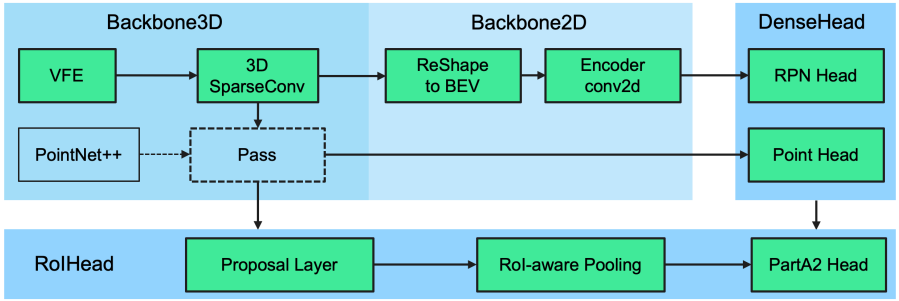
VoteNet / 3DSSD



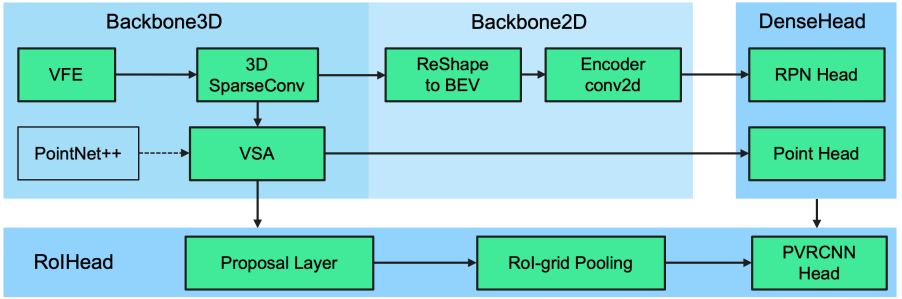
PointRCNN / STD



PartA2-Net



PV-RCNN



OpenPCDet: 3D Detection Toolbox

香港中文大學
The Chinese University of Hong Kong

- How to add customized dataset ?
  1. Write your own DatasetModule that inherited from DatasetTemplate
  2. Overload the self.__getitem__() function to load point clouds / gt_boxes and transform them to the unified coordinate and box definition of OpenPCDet
  3. Call self.prepare_data() to process the data
  4. Overload the self.generate_prediction_dicts() function to transform the predicted results to the format what you like.
  5. Overload the self.evaluation() function to evaluate the results with your own metric.

OpenPCDet: 3D Detection Toolbox

- How to support more models ?
  1. Write your own detector that inherited from DetectorTemplate
  2. Write your own configuration files
  3. Write your own modules to specific directories if we do not provide it
  4. Overload the forward() functionn
  5. Overload the get_training_loss() function

OpenPCDet: Tips with OpenPCDet

香港中文大學
The Chinese University of Hong Kong

- Group the configuration files into different directories
- Multi-gpu training and multi-gpu testing
- View the training process in the tensorboard
- Start a separate evaluation program for fast training
- The following command line parameters are useful:
  - --extra_tag
  - --pretrained_model
  - --set
  - --eval_tag
  - --eval_all

Training script:
bash scripts/dist_train.sh 8 --cfg_file cfgs/kitti_models/pv_rcnn.yaml --extra_tag ex1_tag --set OPTIMIZATION. LR 0.03

Testing script:
sh scripts/dist_test.sh 8 --cfg_file cfgs/kitti_models/pv_rcnn.yaml --extra_tag ex1_tag --eval_all --eval_tag nms02 --set MODEL.POST_PROCESSING.NMS_CONFIG.NMS_THRESH 0.2

Q & A

Welcome to Star / Fork / PR to
OpenPCDet (https://github.com/open-mmlab/OpenPCDet)

Shaoshuai Shi
Ph.D. Student of Multimedia Laboratory
The Chinese University of Hong Kong

智東西 公开课

专注讲解新兴技术创新与应用