

```

_base_ = [
    './_base_/schedules/cyclic_40e.py', './_base_/default_runtime.py',
    './_base_/models/parta2.py'
]

#point_cloud_range = [0, -40, -3, 70.4, 40, 1]

# dataset settings
dataset_type = 'Custom3DDataset'
data_root = 'data/kitti/'
class_names = ['MyPedestrian', 'MyCyclist', 'Car']
input_modality = dict(use_lidar=True, use_camera=False)
train_pipeline = [
    dict(type='LoadPointsFromFile', coord_type='LIDAR', load_dim=4, use_dim=4),
    dict(type='LoadAnnotations3D', with_bbox_3d=True, with_label_3d=True),
    #dict(
        #type='ObjectNoise',
        #num_try=100,
        #translation_std=[1.0, 1.0, 0.5],
        #global_rot_range=[0.0, 0.0],
        #rot_range=[-0.78539816, 0.78539816]),
    dict(type='RandomFlip3D', flip_ratio_bev_horizontal=0.5),
    #dict(
        #type='GlobalRotScaleTrans',
        #rot_range=[-0.78539816, 0.78539816],
        #scale_ratio_range=[0.95, 1.05]),
    dict(type='PointsRangeFilter', point_cloud_range=point_cloud_range),
    dict(type='ObjectRangeFilter', point_cloud_range=point_cloud_range),
    dict(type='ObjectNameFilter', classes=class_names),
    dict(type='PointShuffle'),
    dict(type='DefaultFormatBundle3D', class_names=class_names),
    dict(type='Collect3D', keys=['points', 'gt_bboxes_3d', 'gt_labels_3d'])
]
test_pipeline = [
    dict(type='LoadPointsFromFile', coord_type='LIDAR', load_dim=4, use_dim=4),
    dict(type='DefaultFormatBundle3D', class_names=class_names,
with_label=False),
]
# construct a pipeline for data and gt loading in show function
# please keep its loading function consistent with test_pipeline (e.g. client)
eval_pipeline = [
    dict(type='LoadPointsFromFile', coord_type='LIDAR', load_dim=4, use_dim=4),
    dict(
        type='DefaultFormatBundle3D',
        class_names=class_names,
        with_label=False),
    dict(type='Collect3D', keys=['points'])
]

data = dict(
    samples_per_gpu=1,
    workers_per_gpu=1,
    train=dict(
        type='Custom3DDataset',

```

```
        data_root=data_root,
        ann_file = data_root + 'annotation_smol.pkl',
        pipeline=train_pipeline,
        classes=class_names),
val=dict(
    type=dataset_type,
    data_root=data_root,
    ann_file=data_root + 'kitti_infos_val.pkl',
    pipeline=test_pipeline,
    modality=input_modality,
    classes=class_names,
    box_type_3d='LiDAR',
    test_mode=True),
test=dict(
    type=dataset_type,
    data_root=data_root,
    ann_file=data_root + 'kitti_infos_val.pkl',
    pipeline=test_pipeline,
    modality=input_modality,
    classes=class_names,
    box_type_3d='LiDAR',
    test_mode=True))
```

```
# Part-A2 uses a different learning rate from what SECOND uses.
lr = 0.0001
optimizer = dict(lr=lr)
evaluation = dict(pipeline=eval_pipeline)
find_unused_parameters = True
```