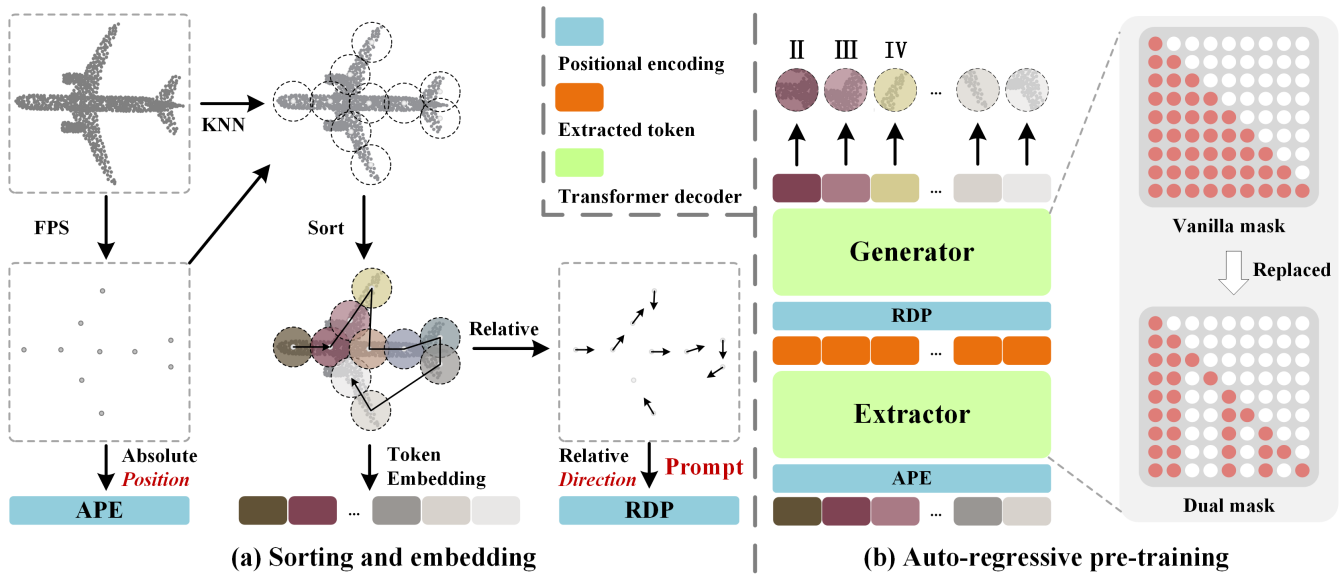
README.md   5/21/2023

PointGPT

PointGPT: Auto-regressively Generative Pre-training from Point Clouds

In this work, we present PointGPT, a novel approach that extends the concept of GPT to point clouds, utilizing a point cloud auto-regressive generation task for pre-training transformer models. In object classification tasks, our PointGPT achieves 94.9% accuracy on the ModelNet40 dataset and 93.4% accuracy on the ScanObjectNN dataset, outperforming all other transformer models. In few-shot learning tasks, our method also attains new SOTA performance on all four benchmarks.

(a) Sorting and embedding   **(b) Auto-regressive pre-training**

1. Requirements

PyTorch >= 1.7.0 < 1.11.0; python >= 3.7; CUDA >= 9.0; GCC >= 4.9; torchvision;

```
pip install -r requirements.txt
```

```
# Chamfer Distance & emd
cd ./extensions/chamfer_dist
python setup.py install --user
cd ./extensions/emd
python setup.py install --user
# PointNet++
pip install
"git+https://github.com/erikwijmans/Pointnet2_PyTorch.git#egg=pointnet2_ops
&subdirectory=pointnet2_ops_lib"
# GPU kNN
pip install --upgrade
https://github.com/unlimblue/KNN_CUDA/releases/download/0.2/KNN_CUDA-0.2-
py3-none-any.whl
```

1 / 3

README.md   5/21/2023

2. Datasets

We use ShapeNet, ScanObjectNN, ModelNet40 and ShapeNetPart for pre-training the PointGPT-S model. See DATASET.md for details.

3. PointGPT Pre-training

To pretrain PointGPT on ShapeNet training set, run the following command.

```
CUDA_VISIBLE_DEVICES=<GPU> python main.py --config cfgs/pretrain.yaml --
exp_name <output_file_name>
```

4. PointGPT Fine-tuning

Fine-tuning on ScanObjectNN, run the following command:

```
CUDA_VISIBLE_DEVICES=<GPUs> python main.py --config
cfgs/finetune_scan_hardest.yaml \
--finetune_model --exp_name <output_file_name> --ckpts <path/to/pre-
trained/model>
```

Fine-tuning on ModelNet40, run the following command:

```
CUDA_VISIBLE_DEVICES=<GPUs> python main.py --config
cfgs/finetune_modelnet.yaml \
--finetune_model --exp_name <output_file_name> --ckpts <path/to/pre-
trained/model>
```

Voting on ModelNet40, run the following command:

```
CUDA_VISIBLE_DEVICES=<GPUs> python main.py --test --config
cfgs/finetune_modelnet.yaml \
--exp_name <output_file_name> --ckpts <path/to/best/fine-tuned/model>
```

Few-shot learning, run the following command:

```
CUDA_VISIBLE_DEVICES=<GPUs> python main.py --config cfgs/fewshot.yaml --
finetune_model \
--ckpts <path/to/pre-trained/model> --exp_name <output_file_name> --way <5
or 10> --shot <10 or 20> --fold <0-9>
```

Part segmentation on ShapeNetPart, run the following command:

2 / 3

README.md   5/21/2023


```
cd segmentation
python main.py --ckpts <path/to/pre-trained/model> --root path/to/data --
learning_rate 0.0002 --epoch 300
```

5. Visualization

Visulization of pre-trained model on ShapeNet validation set, run:

```
python main_vis.py --test --ckpts <path/to/pre-trained/model> --config
cfgs/pretrain.yaml --exp_name <name>
```

Input   **Generation**   **Input**   **Generation**   **Input**   **Generation**

3 / 3