kaggle

Kaggle Winner Presentation (6th)

Google - Fast or Slow? Predict AI Model Runtime

https://www.kaggle.com/competitions/predict-ai-model-runtime

HENG CHER KENG

https://www.kaggle.com/hengck23

Agenda

Agenda

1. Background
2. Summary
3. Feature selection & engineering
4. Training methods
5. Important findings
6. Simple model
7. Code review

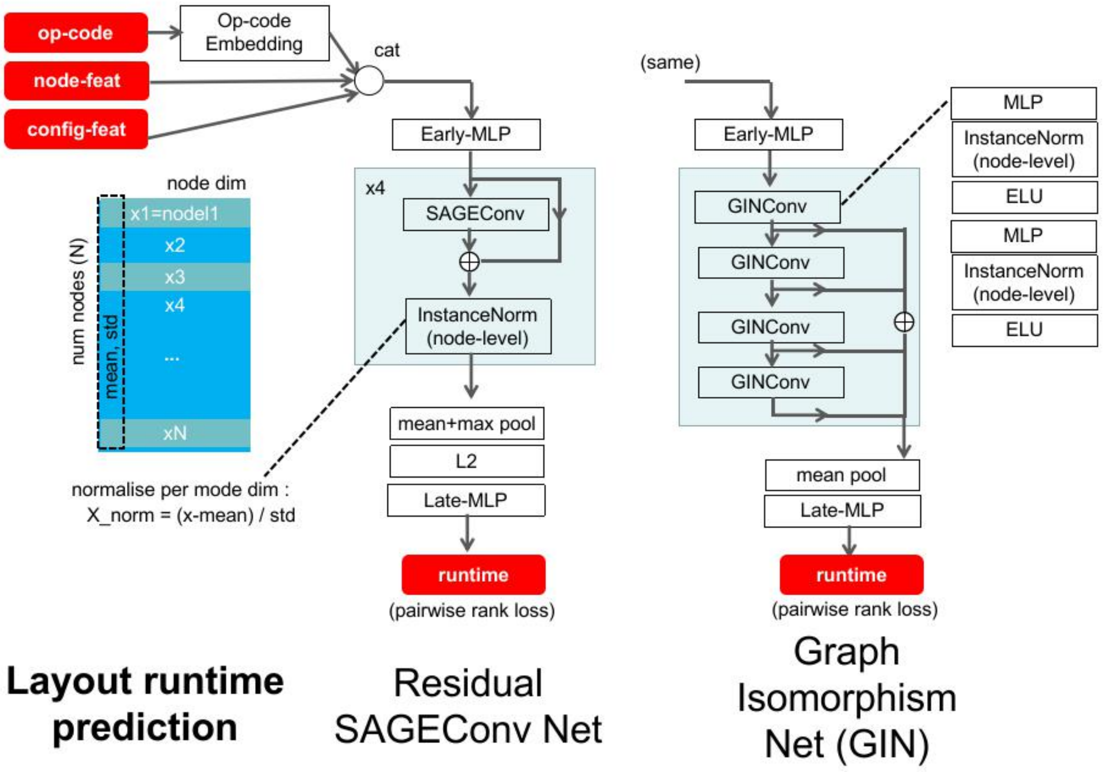
Background

Background

- Contract computer vision and deep learning algorithm engineer.
  - find fracture in x-ray images
  - implement visual slam for robotic navigation.

- Familiar with deep learning and build deep models in my work.
- Experiences in graph neural net GNN from previous Kaggle competitions.

Summary



Layout runtime prediction

Residual SAGEConv Net

Graph Isomorphism Net (GIN)

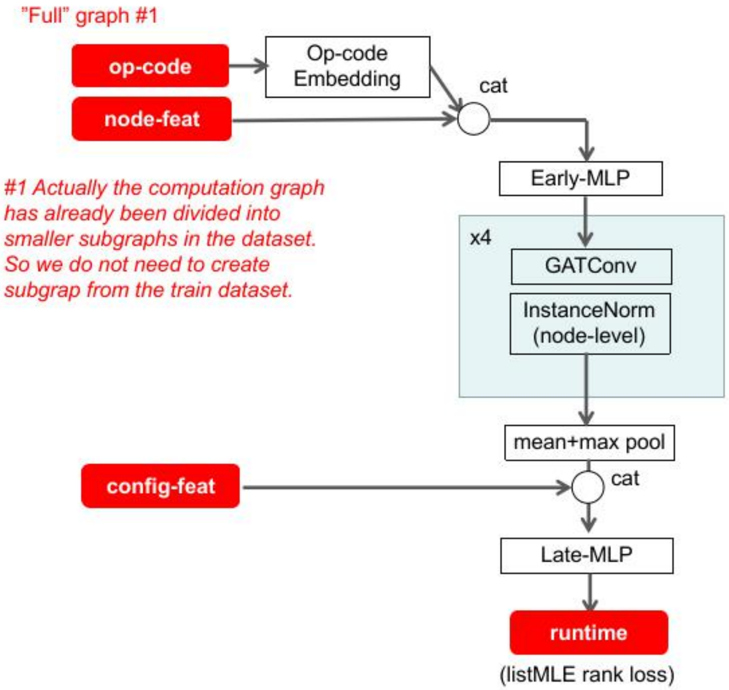
For layout runtime prediction:

- Reduce input graph to subgraph by including only the 5-hop neighbours from the configure node

- GNN model with 4-layer SAGE-conv[2] with residual shortcut

- GNN model with 4-layer GIN-conv[3]

- Graph instance normalisation[1] over node



#1 Actually the computation graph has already been divided into smaller subgraphs in the dataset. So we do not need to create subgrap from the train dataset.

Tile runtime prediction

GATConv Net

For tileruntime prediction:
- GNN model with 4-layer GAT-conv[4]

- Pytorch geometric used for building GNN.
- Training takes abopout 4 to 6 hours for one model in each collection, using Nvidia Quadro RTX 8000 GPU with memory 48 GB.

[1] "Learning Graph Normalization for Graph Neural Networks" - Yihao Chen https://arxiv.org/abs/2009.11746

[2] "Inductive Representation Learning on Large Graphs" - William L. Hamilton https://arxiv.org/abs/1706.02216

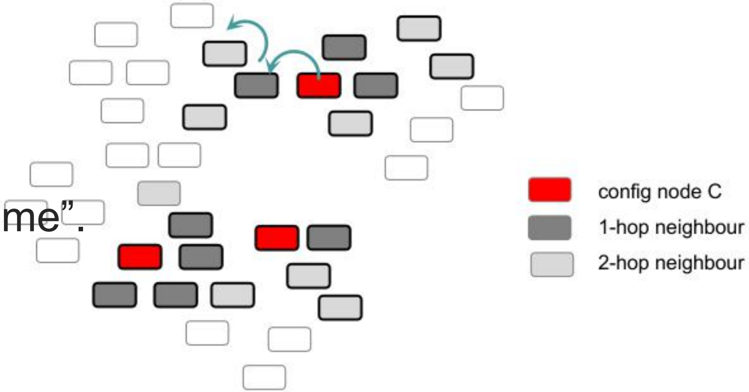
[3] "How Powerful are Graph Neural Networks?" - Keyulu Xu https://arxiv.org/abs/1810.00826

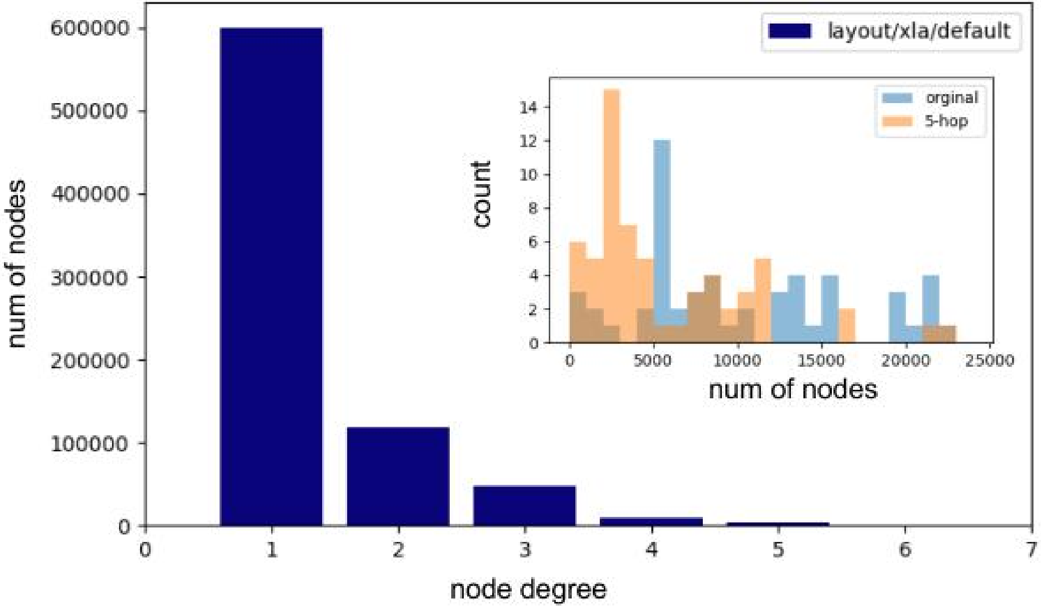
[4] "Graph Attention Networks" - Petar Veličković https://arxiv.org/abs/1710.10903

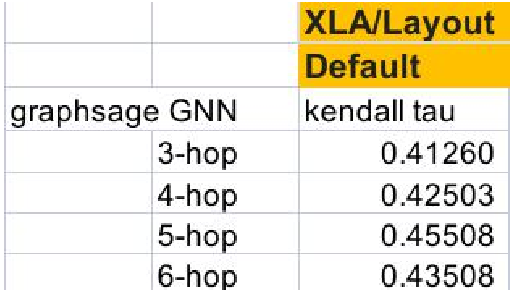
Features Selection/ Engineering

5-hop neighbourhood subgraph as input:
- config node + their neighbours

- relative runtime ranking, just need to consider the "difference of two graph" to predict the "difference in runtime".



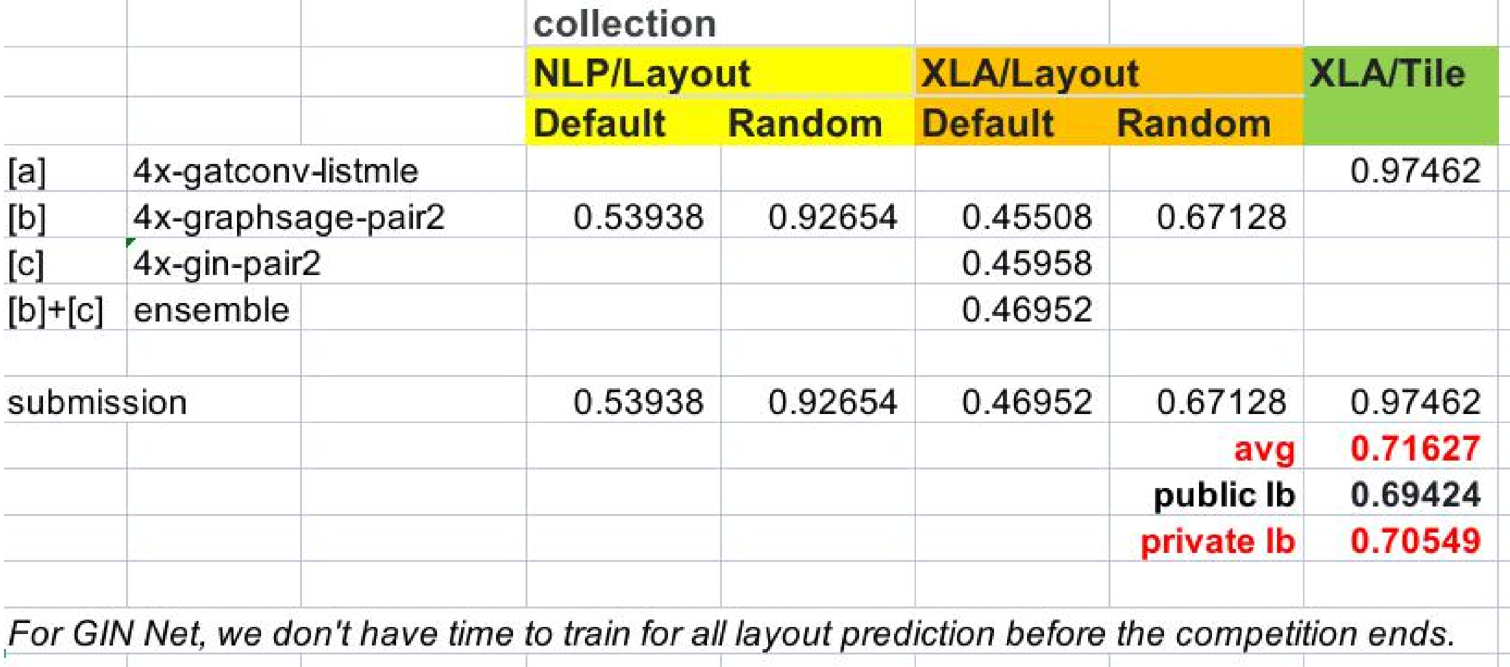




		XLA/Layout
		Default
graphsage GNN		kendall tau
	3-hop	0.41260
	4-hop	0.42503
	5-hop	0.45508
	6-hop	0.43508

Features Selection/Engineering

- Ensemble should imporve results



		collection				
		NLP/Layout		XLA/Layout		XLA/Tile
		Default	Random	Default	Random	
[a]	4x-gatconv-listmle					0.97462
[b]	4x-graphsage-pair2	0.53938	0.92654	0.45508	0.67128	
[c]	4x-gin-pair2			0.45958		
[b]+[c]	ensemble			0.46952		
submission		0.53938	0.92654	0.46952	0.67128	0.97462
					avg	0.71627
					public lb	0.69424
					private lb	0.70549

For GIN Net, we don't have time to train for all layout prediction before the competition ends.

Training Methods

Training Methods

- For layout target kendall tau, use pairwise rank loss.
- Tile top5 slowndown target use listMLE

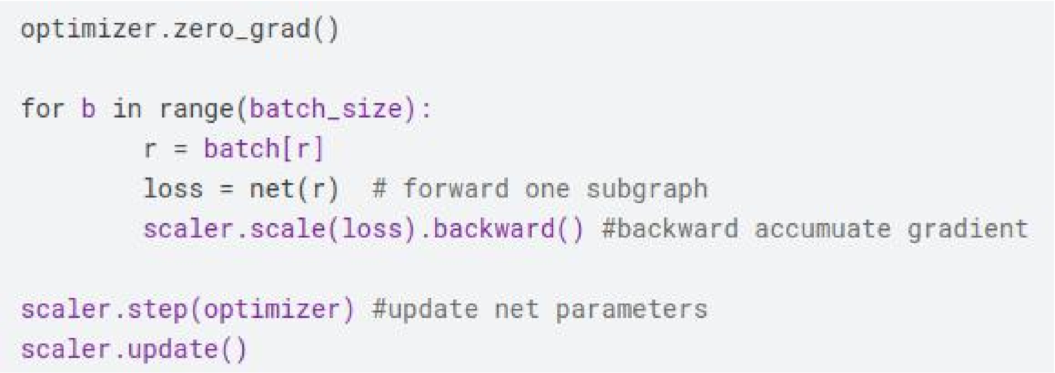
- ADAM optimizer with fixed lr 0.0005
- Use stochastic weight averaging SWA.

    final weights = average last 10 trained weights

  +0.01 improvement over best model

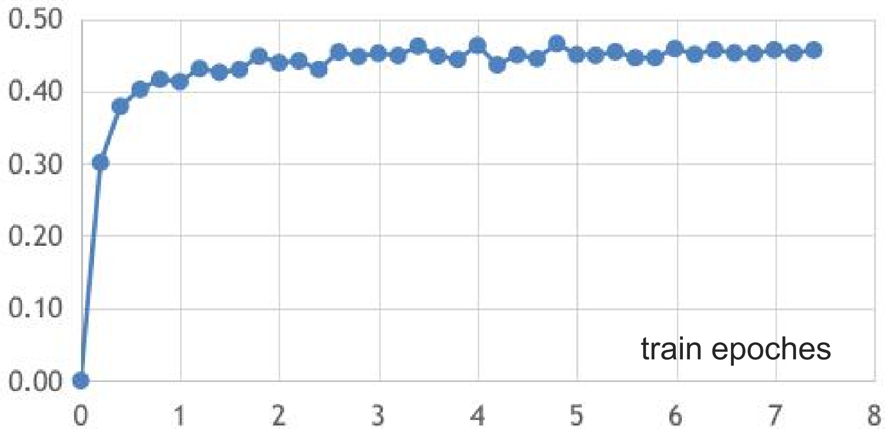
- batch size = 32, sample 80 to 100 configurations per subgraph

- gradient accumation



```
optimizer.zero_grad()

for b in range(batch_size):
        r = batch[r]
        loss = net(r)  # forward one subgraph
        scaler.scale(loss).backward() #backward accumuate gradient

scaler.step(optimizer) #update net parameters
scaler.update()
```

validation kendall tau (layout/xla/default)



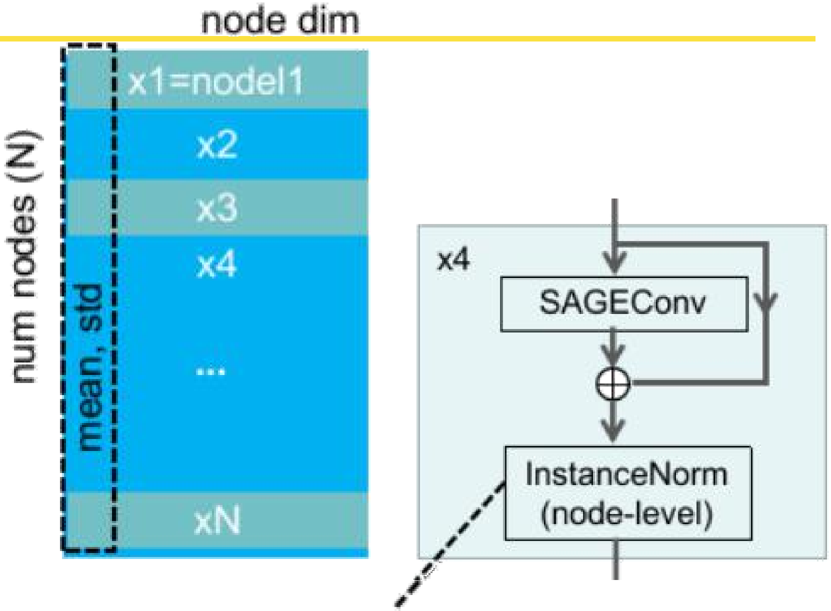
Important and Interesting Findings

Important and Interesting Findings

- Graph instance normalisation gives faster convergence and improve validation kendall tau by 0.02

- Attention pooling at graph readout for layout runtime prediction improves validation 0.01 for some cases

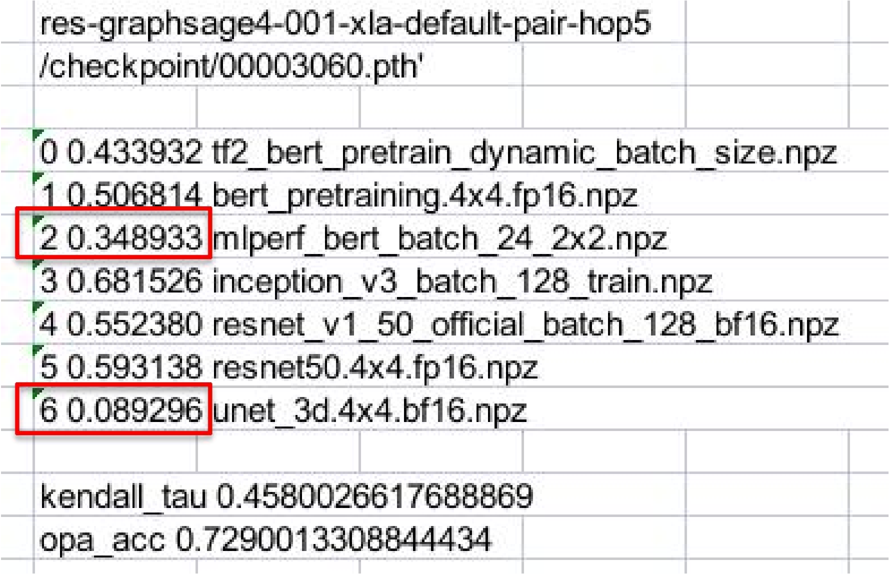
this solution is not selected becuase it did not improve public LB. (But private LB is better)



normalise per mode dim :
X_norm = (x-mean) / std

Important and Interesting Findings

- validation kendall tau has large variance. Careful to select final submission to avoid “shakeup” for private LB.



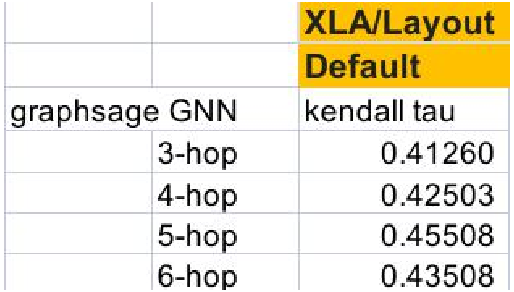
```
res-graphsage4-001-xla-default-pair-hop5
/checkpoint/00003060.pth'

0 0.433932 tf2_bert_pretrain_dynamic_batch_size.npz
1 0.506814 bert_pretraining.4x4.fp16.npz
2 0.348933 mlperf_bert_batch_24_2x2.npz
3 0.681526 inception_v3_batch_128_train.npz
4 0.552380 resnet_v1_50_official_batch_128_bf16.npz
5 0.593138 resnet50.4x4.fp16.npz
6 0.089296 unet_3d.4x4.bf16.npz

kendall_tau 0.4580026617688869
opa_acc 0.7290013308844434
```

Simple Model

Simple Model

- Possible to use even smaller 4-hop neighbourhood subgraph for better speed at the expense of 5 to 10% loss in accuracy

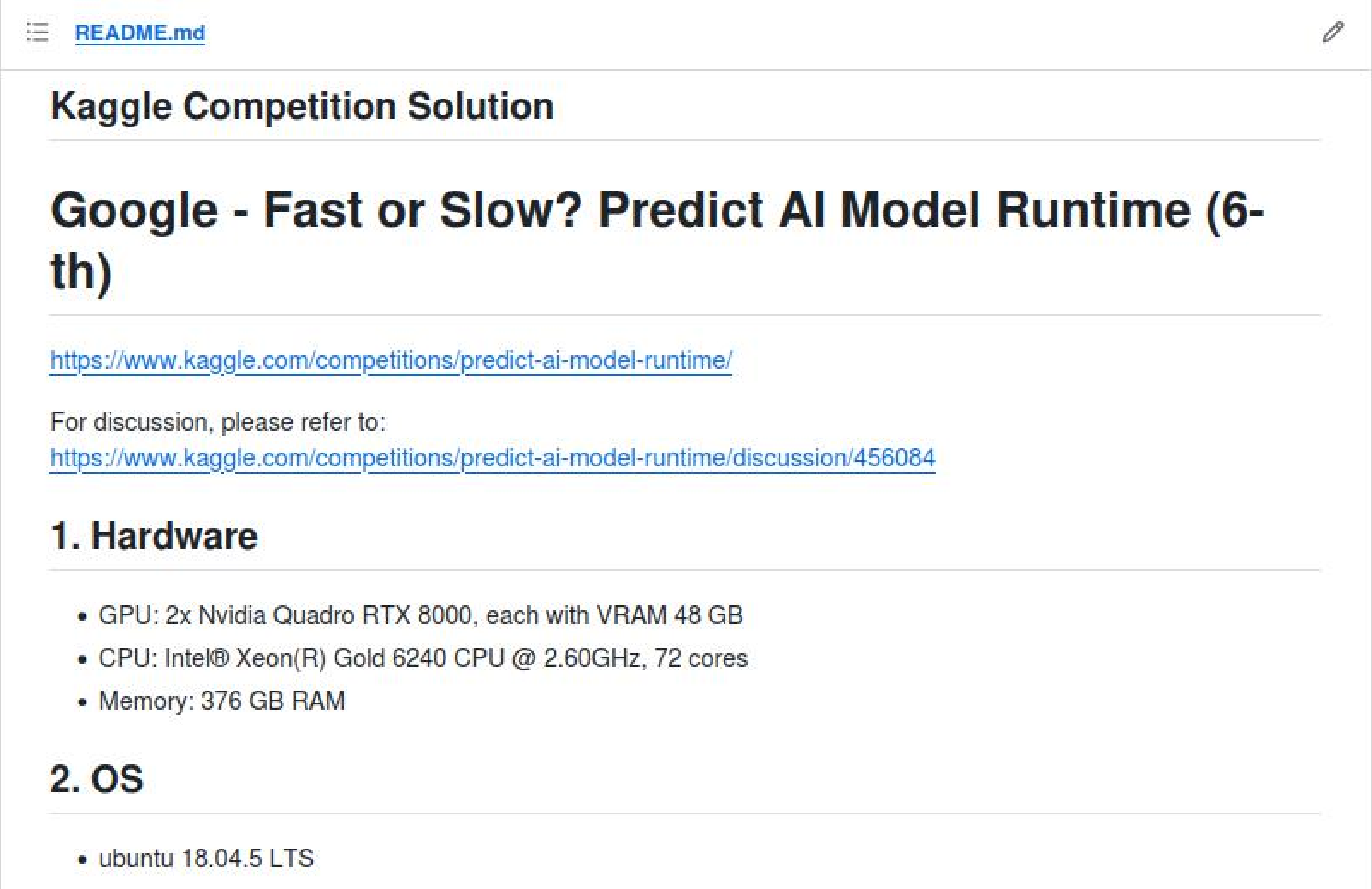
- Number of graph conv layers can also be reduced. We find num of layer = num of hop + 1



		XLA/Layout
		Default
graphsage GNN		kendall tau
	3-hop	0.41260
	4-hop	0.42503
	5-hop	0.45508
	6-hop	0.43508

Code Review

https://github.com/hengck23/solution-predict-ai-model-runtime



README.md

Kaggle Competition Solution

Google - Fast or Slow? Predict AI Model Runtime (6-th)

https://www.kaggle.com/competitions/predict-ai-model-runtime/

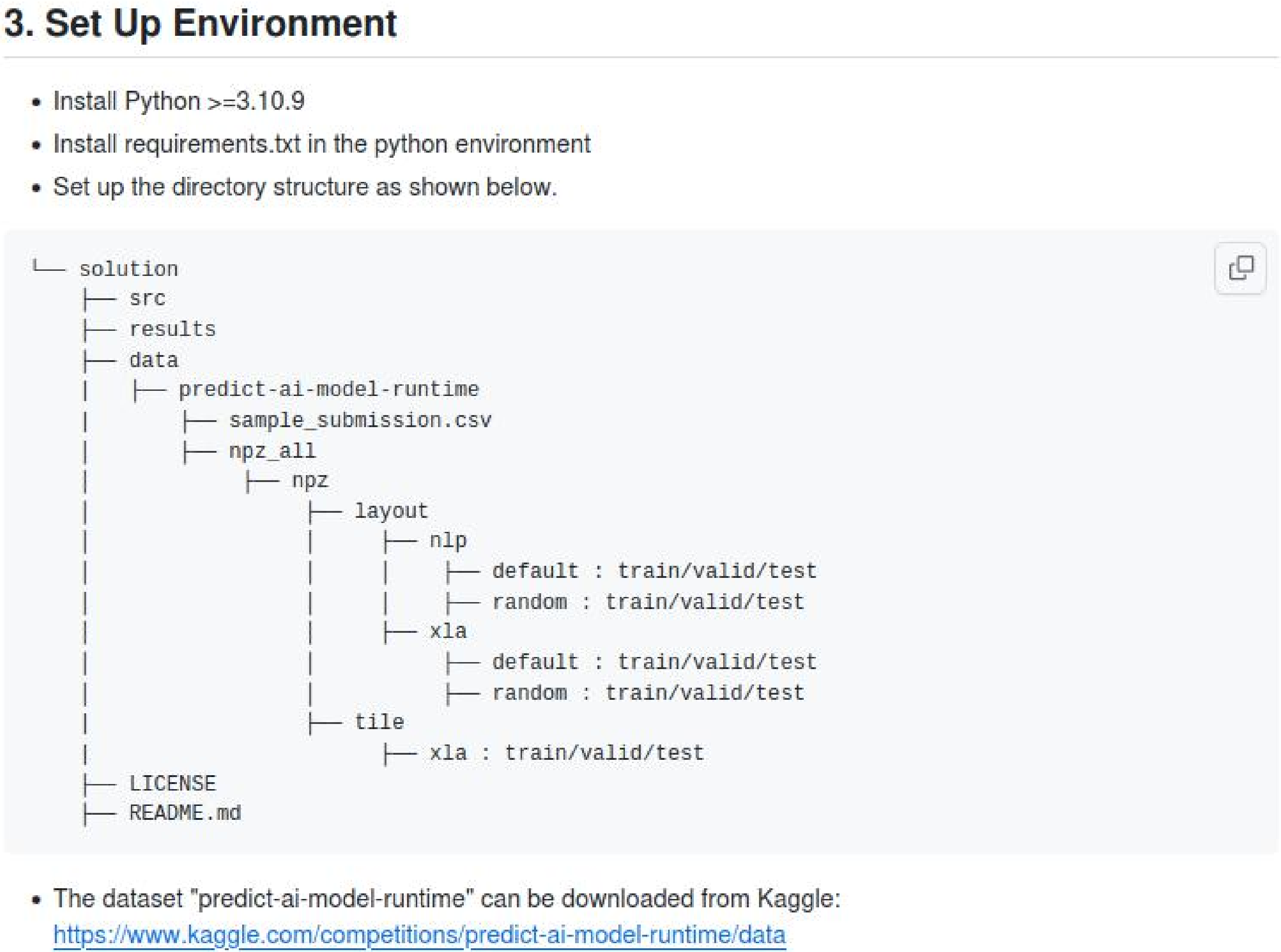
For discussion, please refer to:
https://www.kaggle.com/competitions/predict-ai-model-runtime/discussion/456084

1. Hardware

- GPU: 2x Nvidia Quadro RTX 8000, each with VRAM 48 GB
- CPU: Intel® Xeon(R) Gold 6240 CPU @ 2.60GHz, 72 cores
- Memory: 376 GB RAM

2. OS

- ubuntu 18.04.5 LTS

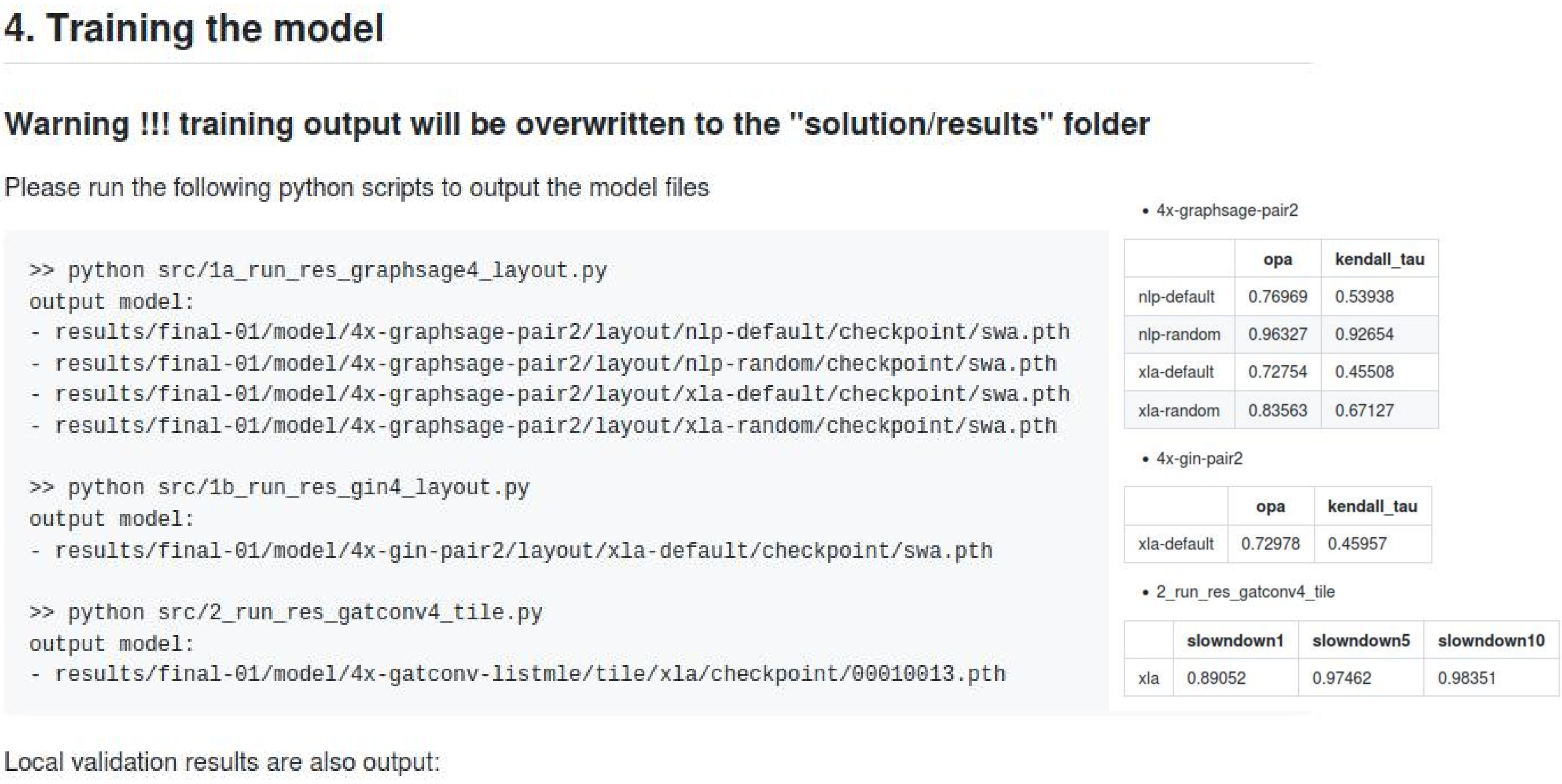


3. Set Up Environment

- Install Python >=3.10.9
- Install requirements.txt in the python environment
- Set up the directory structure as shown below.

```
└── solution
    ├── src
    ├── results
    ├── data
    |   ├── predict-ai-model-runtime
    |       ├── sample_submission.csv
    |       ├── npz_all
    |            ├── npz
    |                 ├── layout
    |                 |     ├── nlp
    |                 |     |    ├── default : train/valid/test
    |                 |     |    ├── random : train/valid/test
    |                 |     ├── xla
    |                 |          ├── default : train/valid/test
    |                 |          ├── random : train/valid/test
    |                 ├── tile
    |                       ├── xla : train/valid/test
    |
    ├── LICENSE
    ├── README.md
```

- The dataset "predict-ai-model-runtime" can be downloaded from Kaggle: https://www.kaggle.com/competitions/predict-ai-model-runtime/data



4. Training the model

Warning !!! training output will be overwritten to the "solution/results" folder

Please run the following python scripts to output the model files

```
>> python src/1a_run_res_graphsage4_layout.py
output model:
- results/final-01/model/4x-graphsage-pair2/layout/nlp-default/checkpoint/swa.pth
- results/final-01/model/4x-graphsage-pair2/layout/nlp-random/checkpoint/swa.pth
- results/final-01/model/4x-graphsage-pair2/layout/xla-default/checkpoint/swa.pth
- results/final-01/model/4x-graphsage-pair2/layout/xla-random/checkpoint/swa.pth

>> python src/1b_run_res_gin4_layout.py
output model:
- results/final-01/model/4x-gin-pair2/layout/xla-default/checkpoint/swa.pth

>> python src/2_run_res_gatconv4_tile.py
output model:
- results/final-01/model/4x-gatconv-listmle/tile/xla/checkpoint/00010013.pth
```

- 4x-graphsage-pair2

	opa	kendall_tau
nlp-default	0.76969	0.53938
nlp-random	0.96327	0.92654
xla-default	0.72754	0.45508
xla-random	0.83563	0.67127

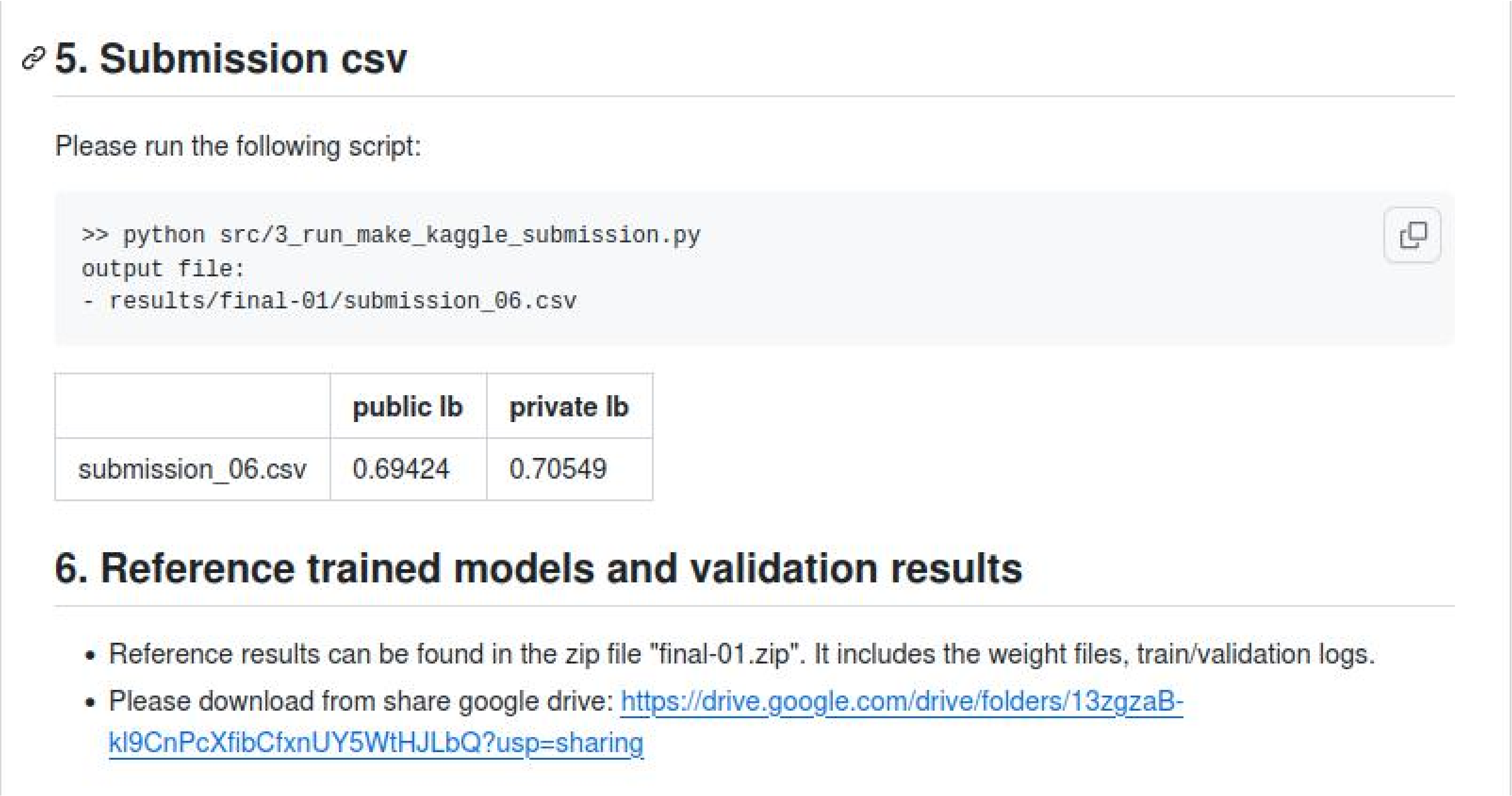
- 4x-gin-pair2

	opa	kendall_tau
xla-default	0.72978	0.45957

- 2_run_res_gatconv4_tile

	slowndown1	slowndown5	slowndown10
xla	0.89052	0.97462	0.98351

Local validation results are also output:



5. Submission csv

Please run the following script:

```
>> python src/3_run_make_kaggle_submission.py
output file:
- results/final-01/submission_06.csv
```

	public lb	private lb
submission_06.csv	0.69424	0.70549

6. Reference trained models and validation results

- Reference results can be found in the zip file "final-01.zip". It includes the weight files, train/validation logs.
- Please download from share google drive: https://drive.google.com/drive/folders/13zgzaB-kl9CnPcXfibCfxnUY5WtHJLbQ?usp=sharing

Question and Answer

kaggle