Efficient Training of BERT by Progressively Stacking

Linyuan Gong, **Di He**, Zhuohan Li, Tao Qin, Liwei Wang, Tie-Yan Liu

Peking University & Microsoft Research Asia

ICML | 2019

BERT: Effective Model with Huge Costs

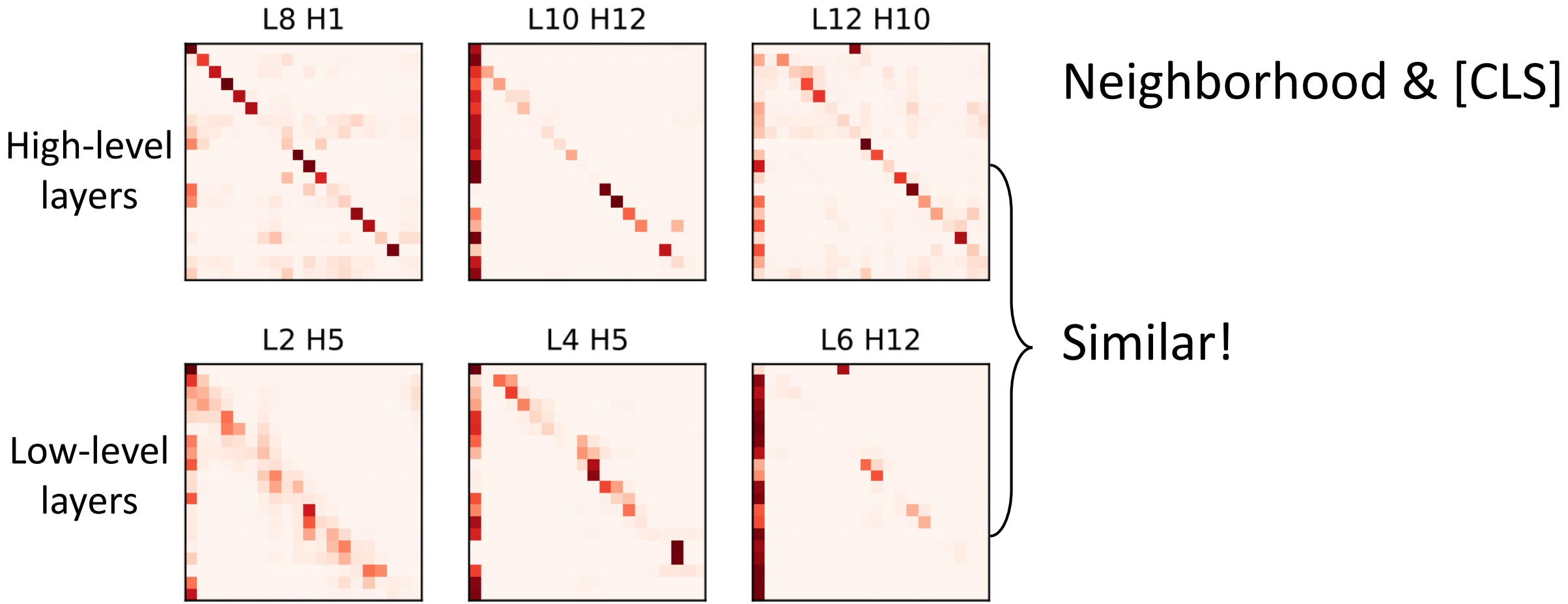
Model
110M/330M parameters

Data
3.4B words (enwiki + book)

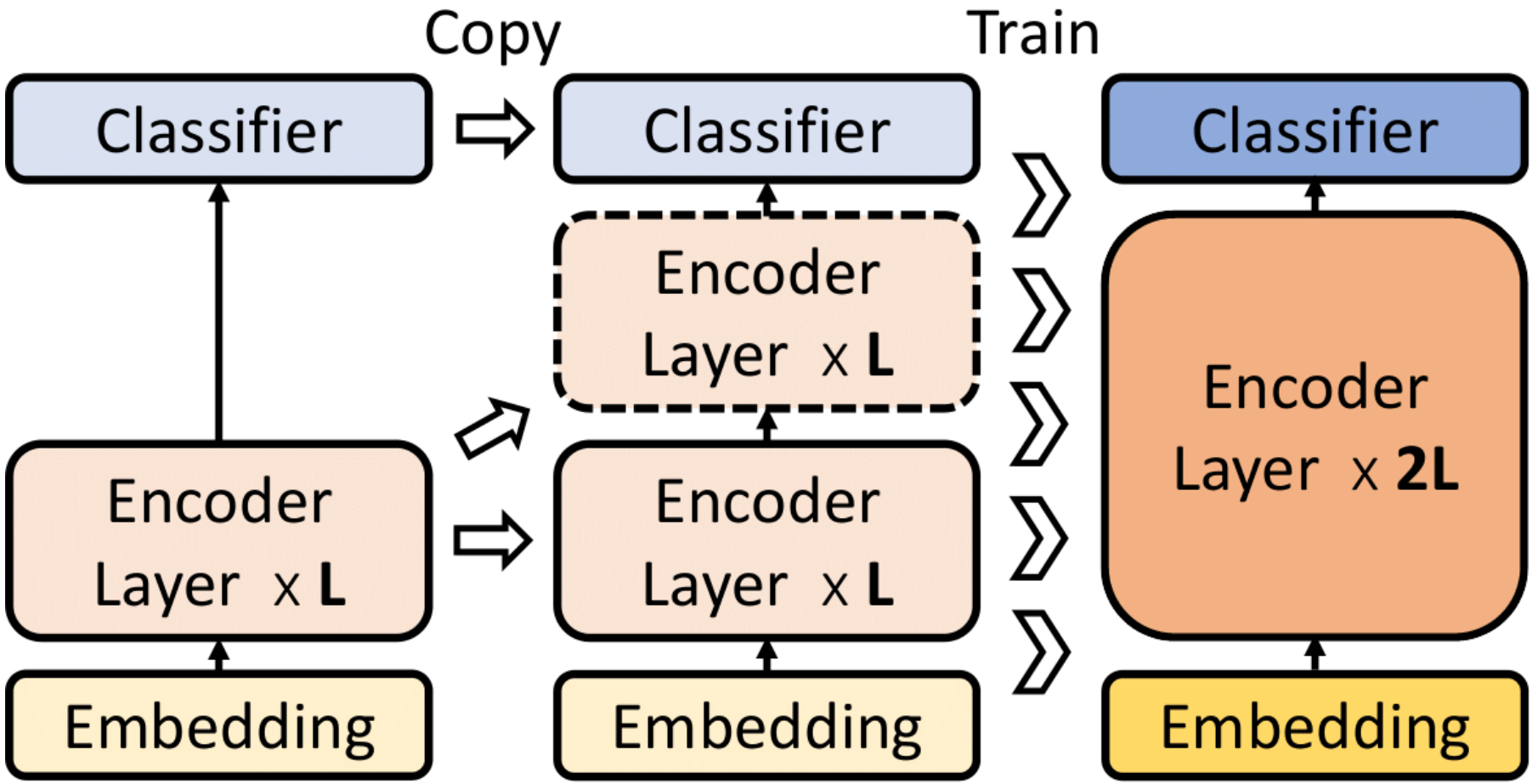
Training
128K tokens * 1M updates

4 Days on 4 TPUs or 23 Days on 4 Tesla P40 GPUs

Attention Distributions of BERT



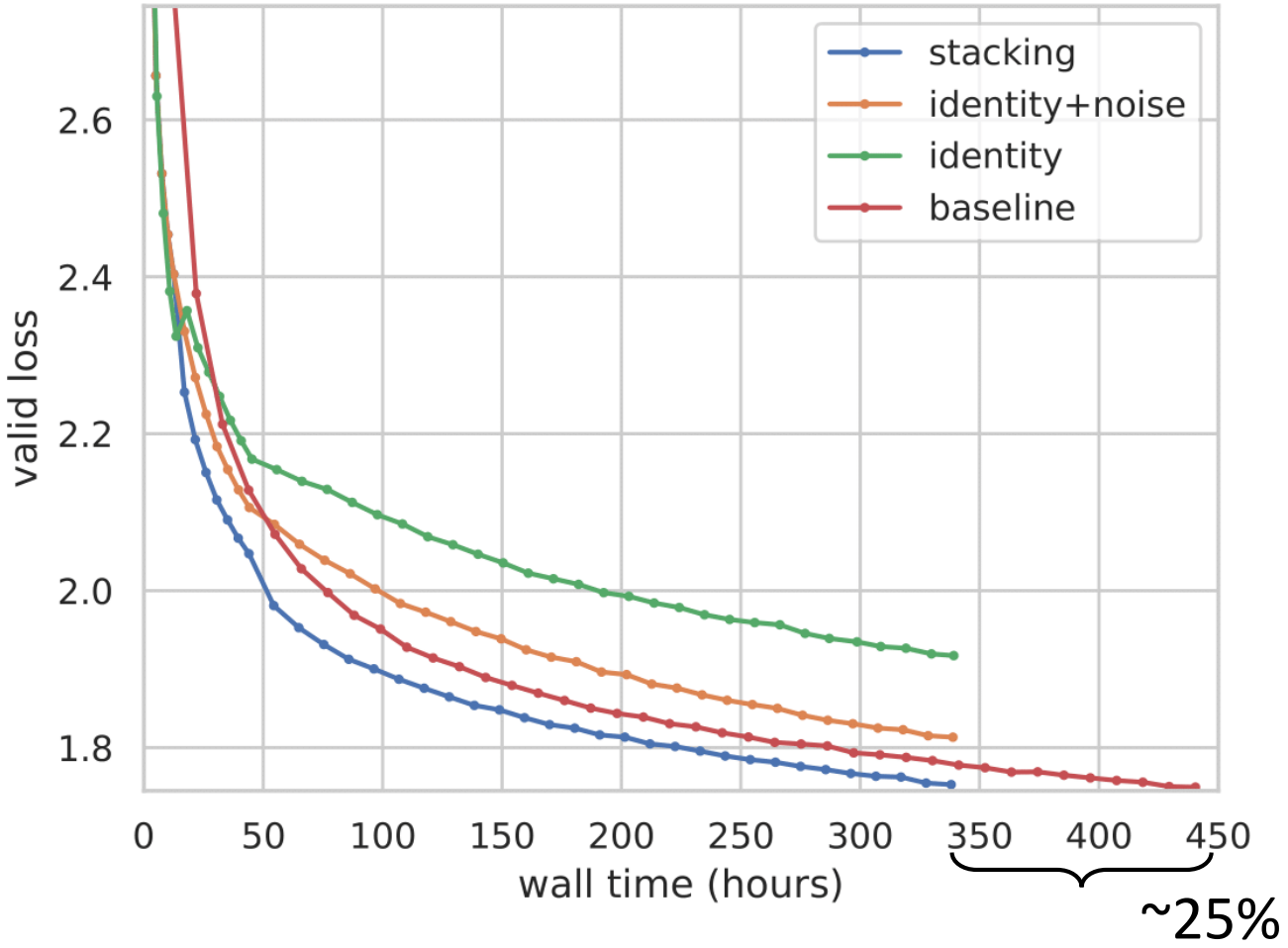
Stacking


Copy
Train
Classifier
Classifier
Classifier
Encoder
Layer x L
Encoder
Layer x L
Encoder
Layer x L
Encoder
Layer x 2L
Embedding
Embedding
Embedding

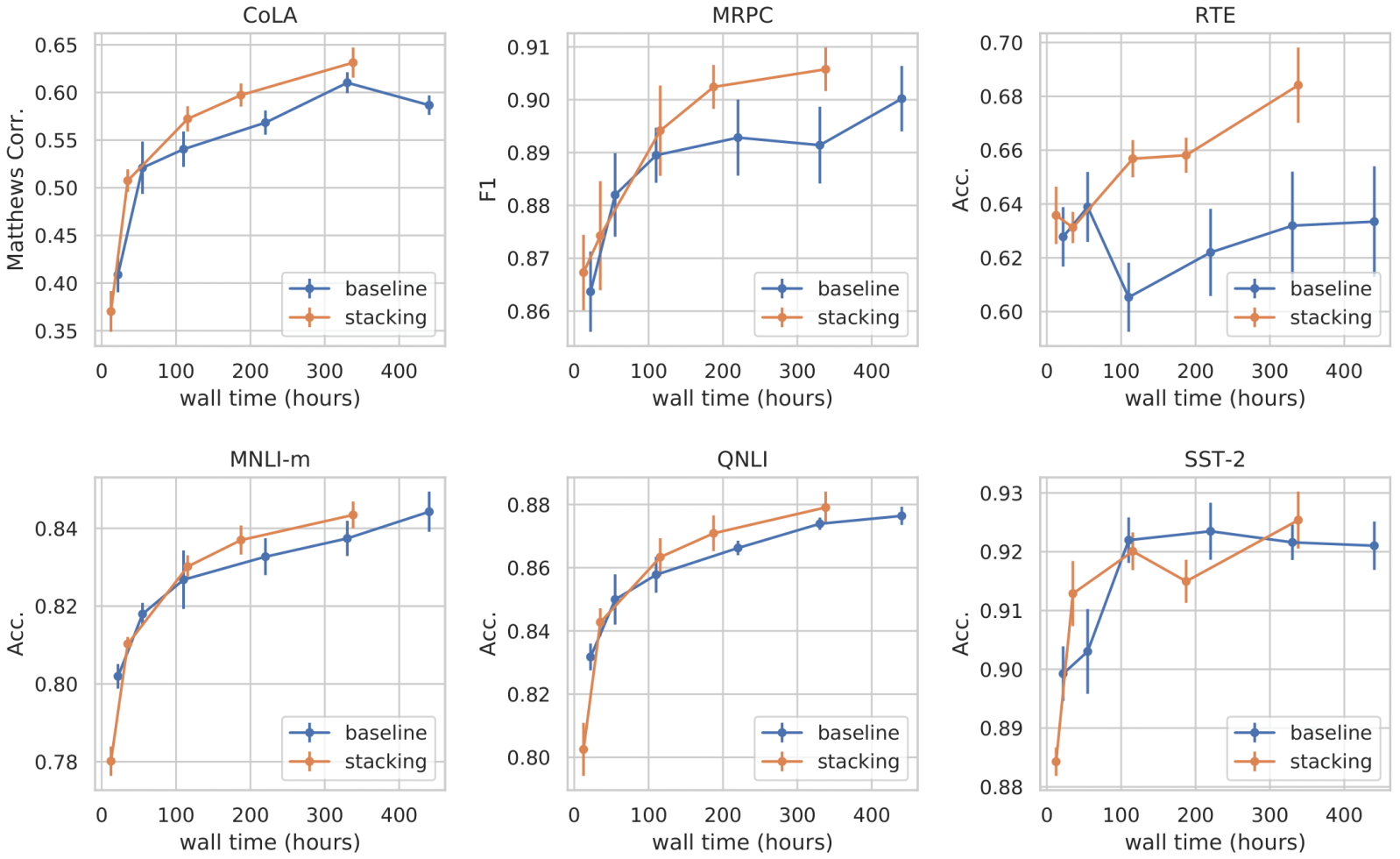
Stacking Progressively

$M_0' \leftarrow \text{InitBERT}(L/2^k)$
$M_0 \leftarrow \text{Train}(M_0')$ {Train from scratch.}
for $i \leftarrow 1$ to k **do**
  $M_i' \leftarrow \text{Stack}(M_i)$ {Doubles the number of layers.}
  $M_i \leftarrow \text{Train}(M_i')$ {M_i has $L/2^{k-i}$ layers.}
end for
return M_k

Result


stacking
identity+noise
identity
baseline
valid loss
2.6
2.4
2.2
2.0
1.8
0
50
100
150
200
250
300
350
400
450
wall time (hours)
~25%

Result


CoLA
Matthews Corr.
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0
100
200
300
400
wall time (hours)
baseline
stacking
MRPC
F1
0.91
0.90
0.89
0.88
0.87
0.86
RTE
Acc.
0.70
0.68
0.66
0.64
0.62
0.60
MNLI-m
0.84
0.82
0.80
0.78
QNLI
0.88
0.86
0.84
0.82
0.80
SST-2
0.93
0.92
0.91
0.90
0.89
0.88

Result

	CoLA	SST-2	MRPC	STS-B	QQP	MNLI	QNLI	RTE	GLUE
BERT-Base	52.1	93.5	**88.9/ 84.8**	**87.1/ 85.8**	**71.2/ 89.2**	**84.6/** 83.4	**90.5**	66.4	78.3
Stacking	**56.2**	**93.9**	88.2/ 83.9	84.2/ 82.5	70.4/ 88.7	84.4/ **84.2**	90.1	**67.0**	**78.4**

Take aways

- Progressively stacking training for BERT is efficient
  - https://github.com/gonglinyuan/StackingBERT
  - Poster **#50**

- Towards a better understanding of Transformer
  - *Understanding and Improving Transformer From a Multi-Particle Dynamic System Point of View,* https://arxiv.org/pdf/1906.02762.pdf
  - *Codes and model ckpts @* https://github.com/zhuohan123/macaron-net