PoNet: Pooling Network for Efficient Token Mixing in Long Sequences

Chao-Hong Tan[1*], Qian Chen[2], Wen Wang[2], Qinglin Zhang[2], Siqi Zheng[2], Zhen-Hua Ling[1]

[1] National Engineering Laboratory for Speech and Language Information Processing, University of Science and Technology of China

[2] Speech Lab, Alibaba Group

[*] Work is done during the internship at Speech Lab, Alibaba Group.

1. Introduction

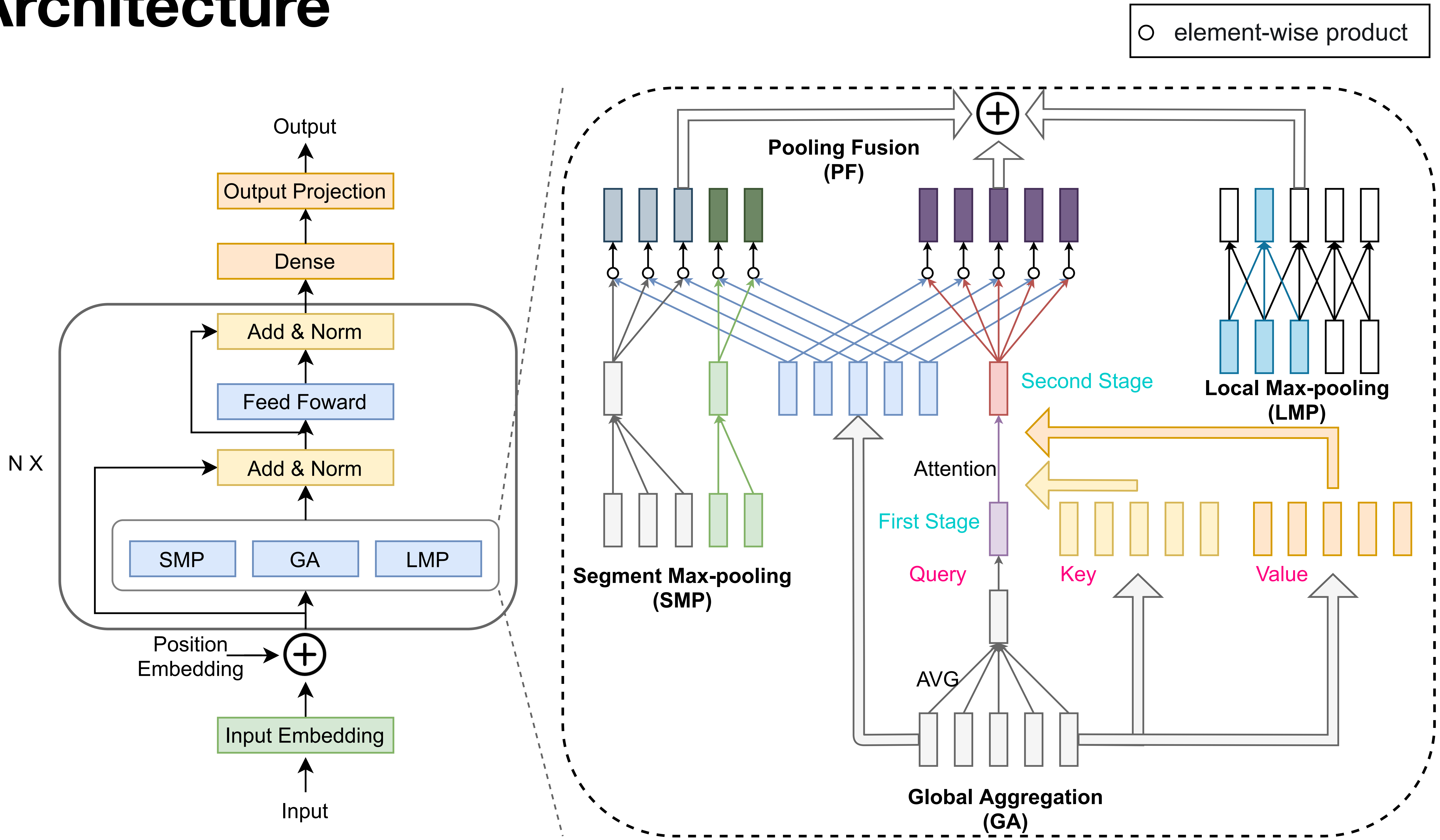
- The self-attention mechanism in transformer
    - Has quadratic time and memory complexity with respect to the sequence length
    - Hinders applications to long sequences
- We propose a novel **Pooling Network** (**PoNet**) for token mixing in long sequences with **linear** complexity
    - **Long sequence modeling** capabilities - Long Range Arena benchmark
        - Significantly outperforms Transformer by **+2.28 absolute** (+3.9% relative) on accuracy
        - Efficiency up to **9** times faster and memory usage **10** times smaller than Transformer on GPU
    - **Transfer learning** capabilities - GLUE
        - PoNet-Base reaches **95.7%** of the accuracy of BERT-Base on the GLUE benchmark

2. Motivation

- Inspired by the **External Attention** (EA) approach (Guo et al., 2021)
- Simplify EA into multi-layer perceptron (MLP) and Softmax
- Softmax infuses the sequence-level information into each token through the denominator term
    - Involves calculations of exponents, still slow
- Our idea: Using **pooling** as an alternative to capture contextual information

3. Model

PoNet Architecture



The right enlarged view shows multi-granularity pooling (GA, SMP, LMP) and pooling fusion.

3. Model

Global Aggregation (GA)

- Capture the most important global information for each token
- Guarantee an overall linear computational complexity
- **First stage**: **Average** at the sequence level

$$\boldsymbol{g} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{h}_{Q_{g_n}} \in \mathbb{R}^d$$

- **Second stage**: **Cross-attention** to provide a more accurate sequence representation

$$\boldsymbol{g}' = Attention\{\boldsymbol{g}, \boldsymbol{H}_{K_g}, \boldsymbol{H}_{V_g}\}$$

3. Model

Segment Max-pooling (SMP)

- **Alleviate information loss** from compressing a long sequence into a single global token
- Introduce an **intermediate level** between tokens and the global token
- Explore prior knowledge of **structure** in the data

$$s_j^k = \max\{h_{s_{0j}}^k, h_{s_{1j}}^k, \ldots, h_{s_{N_k j}}^k\}$$

$$\boldsymbol{s}^k = \{s_1^k, \ldots, s_d^k\} \in \mathbb{R}^d$$

$$\boldsymbol{S} = \{\boldsymbol{s}^1, \ldots, \boldsymbol{s}^K\} \in \mathbb{R}^{K\times d}$$

3. Model

Local Max-pooling (LMP)

- A standard max-pooling over sliding windows
- Capture contextual information from neighboring tokens for each token
- Different from GA and SMP, the window for LMP is **overlapped**

3. Model

Pooling Fusion (PF)

- Interact with tokens through **element-wise product**

$$\boldsymbol{G}_n = \boldsymbol{g}' \circ \boldsymbol{H}_{o_n}$$

$$\boldsymbol{S'}_n = \boldsymbol{S}_{k(n)} \circ \boldsymbol{H}_{o_n}$$

- Add up these three features as the final output of our multi-granularity pooling block

$$\boldsymbol{P} = \boldsymbol{G} + \boldsymbol{S}' + \boldsymbol{L}$$

4. Experiments

LRA benchmark

Model	ListOps(2K)	Text(4K)	Retrieval(4K)	Image(1K)	Pathfinder(1K)	AVG.
Transformer(1)	**36.37**	64.27	57.46	42.44	71.40	54.39
Longformer (1)	35.63	62.85	56.89	42.22	69.71	53.46
BigBird (1)	36.05	64.02	**59.29**	40.83	74.87	**55.01**
Performer (1)	18.01	**65.40**	53.82	**42.77**	**77.05**	51.41
Transformer(2)	**36.06**	61.54	**59.67**	**41.51**	80.38	**55.83**
Linear (2)	33.75	53.35	58.95	41.04	**83.69**	54.16
FNet (2)	35.33	**65.11**	59.61	38.67	77.80	55.30
Transformer(3)	37.10	65.02	79.35	38.20	**74.16**	58.77
Performer(3)	18.80	63.81	78.62	37.07	69.87	53.63
Reformer(3)	19.05	64.88	78.64	43.29	69.36	55.04
Linformer(3)	37.25	55.91	79.37	37.84	67.60	55.59
Nyströmformer(3)	37.15	65.52	79.56	41.58	70.94	58.95
FNet	37.40	62.52	76.94	35.55	FAIL	53.10
PoNet (Ours)	**37.80**	**69.82**	**80.35**	**46.88**	70.39	**61.05**

Results on the **Long Range Arena (LRA)** benchmark (**AVG**: average accuracy across all tasks).
Results with **(1)** are cited from (Tay et al., 2021), with **(2)** are from (Lee-Thorp et al., 2021), with **(3)** are from (Xiong et al., 2021).
We implement our **PoNet** and re-implement **FNet** (Lee-Thorp et al., 2021) based on the PyTorch codebase from (Xiong et al., 2021) and use the **same experimental configurations** to ensure a fair comparison.
For each group, the **best results** for each task and AVG are bold-faced.

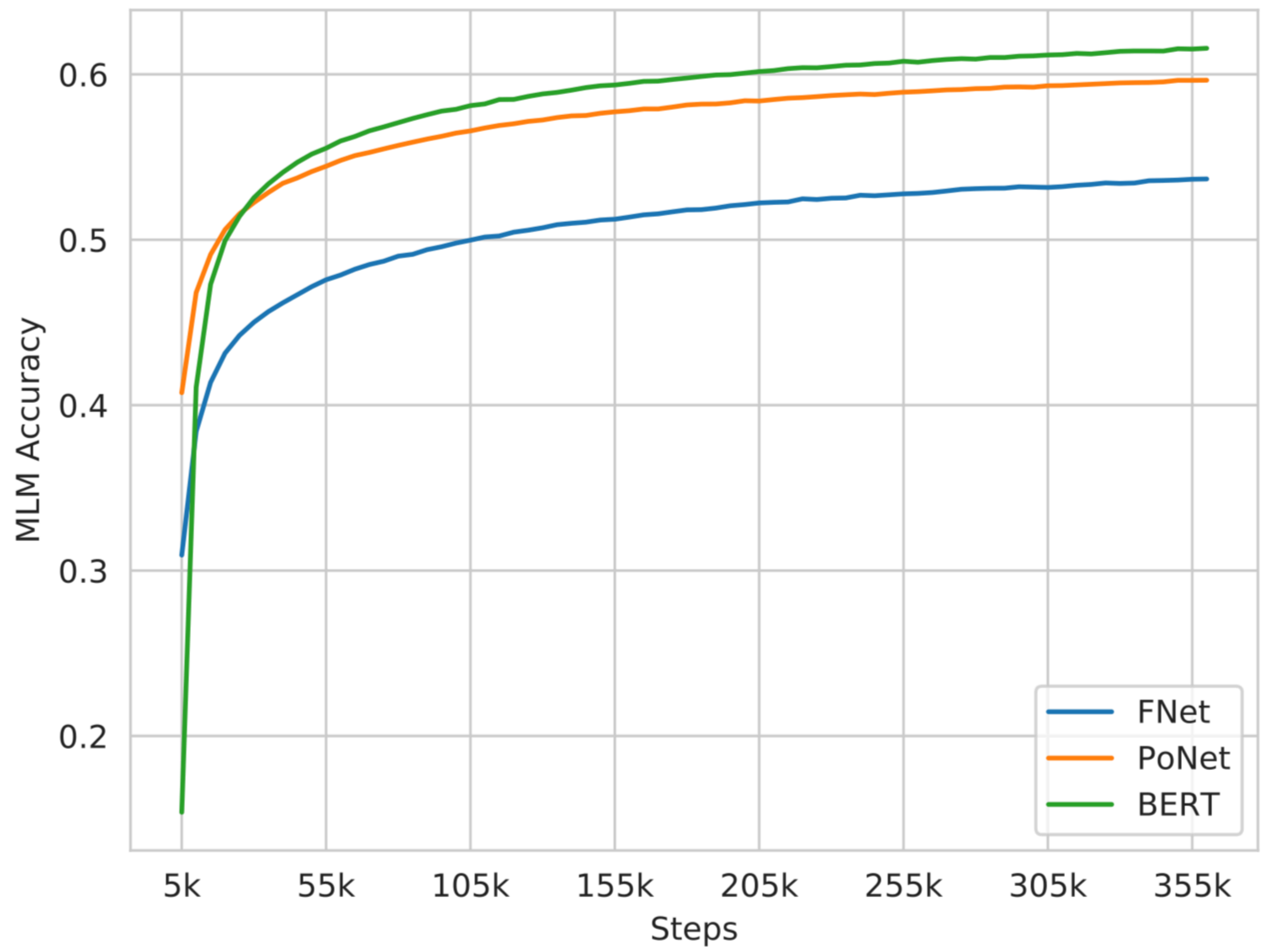
4. Experiments

LRA benchmark

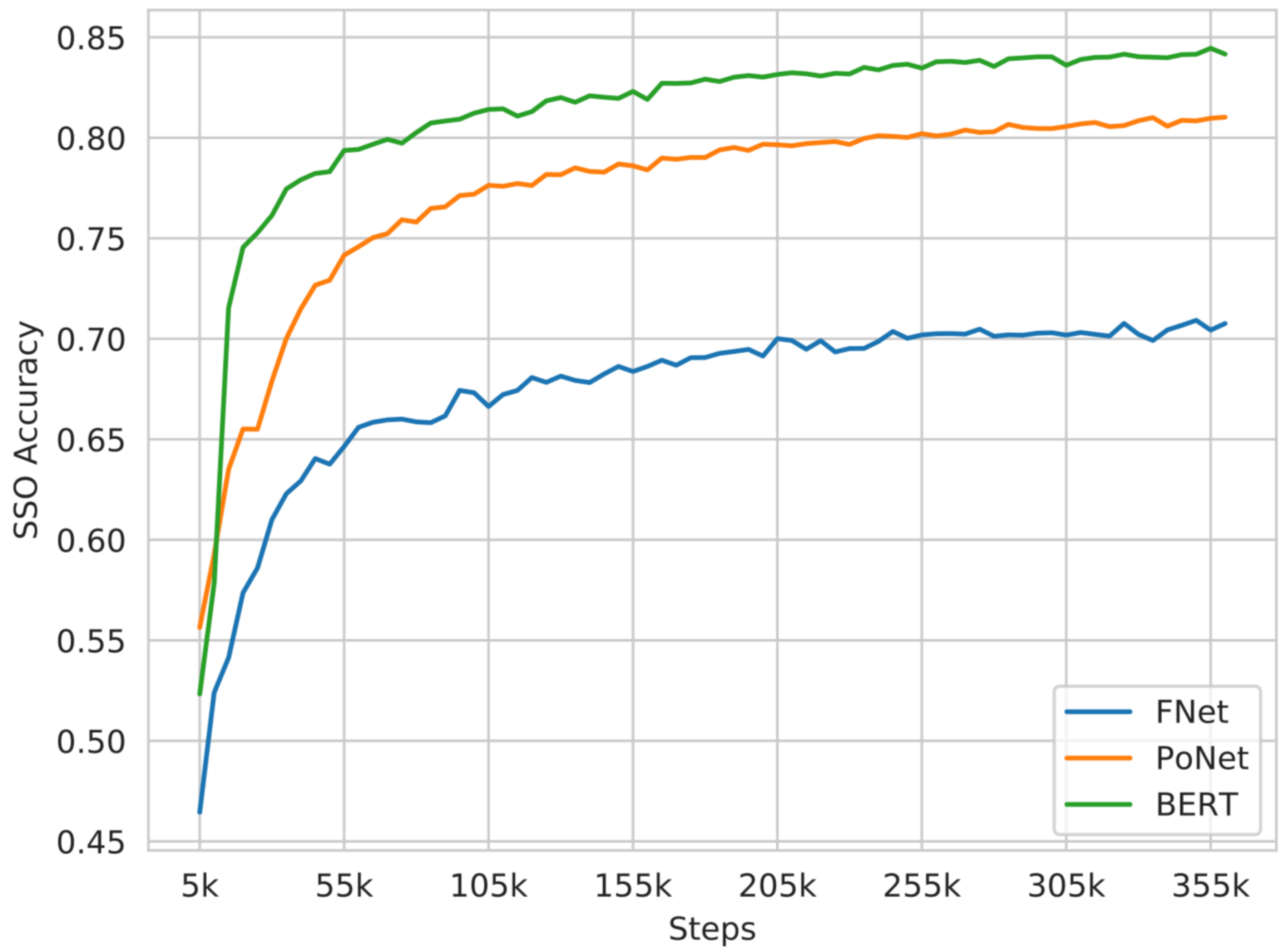
Seq. length	512	1024	2048	4096	8192	16384
	Training Speed (steps/s)↑					
Transformer	45.1	19.4	6.3	1.8	OOM	OOM
Performer	39.4(0.9x)	25.0(1.3x)	14.3(2.3x)	7.8(4.3x)	4.0	2.0
Nyströmformer	39.1(0.9x)	30.3(1.6x)	20.0(3.2x)	11.5(6.4x)	6.1	3.1
FNet	**83.4(1.8x)**	**61.3(3.1x)**	**38.1(6.0x)**	**21.4(11.9x)**	**11.0**	**5.4**
PoNet (Ours)	<u>50.4(1.1x)</u>	<u>40.1(2.1x)</u>	<u>27.8(4.4x)</u>	<u>16.2(9.0x)</u>	<u>8.7</u>	<u>4.5</u>
	Peak Memory Usage (GB)↓					
Transformer	1.4	2.5	6.7	23.8	OOM	OOM
Performer	1.5(1.1x)	2.1(0.8x)	3.1(0.5x)	5.4(0.2x)	9.8	18.7
Nyströmformer	<u>1.2(0.8x)</u>	1.5(0.6x)	1.9(0.3x)	2.8(0.1x)	4.5	8.2
FNet	**1.1(0.8x)**	**1.2(0.5x)**	**1.4(0.2x)**	**1.7(0.1x)**	**2.3**	**3.8**
PoNet (Ours)	**1.1(0.8x)**	<u>1.3(0.5x)</u>	<u>1.7(0.2x)</u>	<u>2.4(0.1x)</u>	<u>3.6</u>	<u>6.5</u>

Comparison of **GPU training speed** and **peak memory consumption** on various input sequence lengths on the LRA text classification task (using the same hyper-parameter setting for this task as in (Xiong et al., 2021)).
The **best results** are bold-faced with the <u>second-best results</u> underlined.

4. Experiments
Transfer Learning — Pre-training Task Accuracy



(a) MLM Accuracy



(b) SSO Accuracy

MLM and **SSO** validation accuracy against the numbers of training steps from **BERT-Base**, **FNet-Base**, and our **PoNet-Base**.
All models are uncased.
MLM as used in BERT (Devlin et al., 2019).
SSO (Sentence Structural Objective) as used in StructBERT (Wang et al., 2020b).

4. Experiments
Transfer Learning — GLUE Fine-tuning Results

Model	MNLI(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	AVG.
BERT-Base	81.35/80.98	88.89	88.01	91.17	47.66	87.83	86.66	69.31	**80.21**
FNet-Base	73.13/73.66	85.75	80.50	88.65	40.61	80.62	80.84	57.40	73.46
PoNet-Base (Ours)	76.99/77.21	87.55	84.33	89.22	45.36	84.57	81.76	64.26	<u>76.80</u>

GLUE Validation results from our PoNet-Base, BERT-Base, and FNet-Base.
All models are uncased and pre-trained with the same configurations using **5GB data** (Wikitext-103 and BooksCorpus) with **340K steps**.
We report the best GLUE results for each model from multiple hyper-parameter configurations.
We report the mean of accuracy and F1 for QQP and MRPC, matthews correlations for CoLA, spearman correlations for STS-B, and accuracy for other tasks.
MNLI(m/mm) means match/mismatch splits.

4. Experiments

Transfer Learning — Extra GLUE Fine-tuning Results

Model	MNLI(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	AVG.
BERT-Base(1)	84/81	87	91	93	73	89	83	83	83.3
Linear-Base(1)	74/75	84	80	94	67	67	83	69	77.0
FNet-Base(1)	72/73	83	80	95	69	79	76	63	76.7
BERT-Base(2)	85/85	89.77	91.78	92.66	58.88	89.28	89.31	70.76	**83.52**
FNet-Base(2)	75/76	86.72	83.23	90.13	35.37	81.43	80.34	59.92	74.23
PoNet-Base(Ours)(2)	78/78	87.76	85.17	89.00	47.24	85.86	83.39	63.53	<u>77.54</u>
BERT-Base(3)	83/83	89.48	90.65	91.74	51.19	89.28	88.73	67.51	**81.63**
FNet-Base(3)	75/76	86.17	82.52	88.42	40.57	83.64	80.90	61.73	74.99
PoNet-Base(Ours)(3)	79/78	87.92	86.31	89.79	45.18	87.17	84.27	66.43	<u>78.29</u>

Extra GLUE Validation results.
Results with **(1)** are from (Lee-Thorp et al., 2021). Results with **(2)** and **(3)** are the best results from searching 20 sets of hyper-parameter configurations based on Table 6 for fine-tuning the pre-trained models.
For BERT-Base **(2)** and FNet-Base **(2)**, we use the official checkpoints provided by authors while for PoNet-Base **(2)**, we pre-train the PoNet model on **5GB data** (Wikitext-103 and BooksCorpus).
For a fair comparison on model capacity by pre-training with more data, BERT-Base**(3)**, FNet-Base**(3)**, and PoNet-Base**(3)** are all pre-trained on the same **16GB data** (Wikipedia and BooksCorpus), trained with MLM+SSO tasks for **1M steps**.
Note that our BERT-Base(3) has a lower performance than the official BERT-Base(2), which is mainly due to the different batch size.

4. Experiments

Transfer Learning — Long-Text Classification

Model	HND(F_1)	IMDb(F_1/Acc)	Yelp-5(F_1)	Arxiv(F_1)
#Example (#Classes)	500 (2)	25000 (2)	650000 (5)	30043 (11)
#Wordpieces avg. (95thpctl.)	734 (1,974)	312 (805)	179 (498)	16,210 (32,247)
RoBERTa-Base (Zaheer et al., 2020)	87.8	95.3/95.0	71.75	87.42
Longformer (Beltagy et al., 2020)	94.8	**95.7**/−−	−−	−−
BigBird (Zaheer et al., 2020)	92.2	−−/**95.2**	**72.16**	**92.31**
BERT-Base	88.0	94.1/94.1	69.59	85.36
FNet-Base	86.3	90.4/90.5	65.49	79.90
PoNet-Base (Ours)	**96.2**	93.0/93.0	69.13	86.11

Fine-tuning results (in F_1 and Acc) on long-text classification datasets.
For the **third group of results**, we use the **official checkpoints of BERT-Base and FNet-Base**.
PoNet-Base reaches **99%** of BERT-Base's F1 on IMDb and Yelp-5

5. Ablation Analysis

Model	Pre-trained tasks MLM	Pre-trained tasks SST	Downstream tasks CoLA	Downstream tasks STS-B
PoNet($340K$ steps)	59.44	80.75	45.36	84.57
PoNet w/o SS-GA	59.33	76.92	46.18	78.38
PoNet w/o GA	56.64	74.36	49.51	64.61
PoNet w/o SMP	56.96	78.41	44.21	84.89
PoNet w/o LMP	56.53	80.27	41.44	85.55
PoNet w/o (SMP&LMP)	43.61	76.72	11.36	84.93
PoNet using $\mathcal{L}_{\mathrm{MN}}$	62.53	79.28	50.91	75.32
PoNet using $\mathcal{L}_{\mathrm{OM}}$	63.11	−−	51.26	69.83

Results of ablation study as accuracy for pre-training MLM and SST (Sentence Structure Task) tasks, matthews correlations for CoLA, and spearman correlations for STS-B.
L_{MN} denotes MLM and NSP loss. L_{OM} denotes only MLM loss.
SST denotes **NSP** when using **L_{MN}** and the **SSO** task otherwise.
All pre-training experiments run 340K steps with 5GB data.

5. Ablation Analysis

- Removing GA
  - Degrades accuracy on SST pre-training task and downstream STS-B
  - Sentence-pair tasks heavily rely on the global information
- Removing SMP or LMP
  - Drastic degradation on MLM and CoLA accuracy
- **All three poolings are important for the modeling capabilities of PoNet**
- Weakening SST loss (L_{MN}, L_{OM})
  - Weakens GA representation learning
  - Strengthens SMP and LMP learning
- **Fine-tuning performance of PoNet on sentence-pair tasks highly relies on sentence structural tasks in pre-training**

6. Conclusion

- A novel **Pooling Network (PoNet)** to replace self-attention with a **multi-granularity pooling block**
- **Linear** time and memory complexity
- **Competitive long-range dependency modeling** capacity and **strong transfer learning** capabilities
- Future work include
  - Further optimization of model structure and pre-training
  - Applying PoNet to a broader range of tasks including generation tasks (e.g., summarization, machine translation)

github.com/lxchtan/PoNet

Thanks for listening!

Reference

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. NAACL-HLT 2019.
- Wei Wang, Bin Bi, Ming Yan, Chen Wu, Jiangnan Xia, Zuyi Bao, Liwei Peng, and Luo Si. StructBERT: incorporating language structures into pre-training for deep language understanding. ICLR 2020.
- Meng-Hao Guo, Zheng-Ning Liu, Tai-Jiang Mu, and Shi-Min Hu. Beyond self-attention: External attention using two linear layers for visual tasks. CoRR, abs/2105.02358, 2021.
- Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long Range Arena : A benchmark for efficient transformers. ICLR 2021.
- James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontañón. FNet: mixing tokens with fourier transforms. CoRR, abs/2105.03824, 2021.
- Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. AAAI 2021.