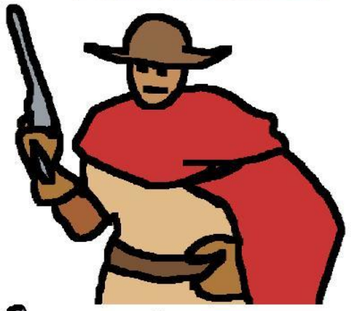
Efficient PLMs

from the perspective of Token

Increasingly Verbose

It's high noon



The current time is twelve o' clock

I am informing you that the current time is twelve o' clock in the afternoon which the sun is at its highest elevation in thy sky.

Us homo sapiens refer the present passage of continued progress of existence and events that occur in irreversible succesion from the pass through the future as twelve o' clock in the afternoon where the sun in which this Earth revolves around is at its highest elevation in the sky and homo sapiens can usually be found eating a light meal which is typically eaten after breakfast and before dinner

The sequence is TOOOOO long!

Where should I pay attention to?

I'm OUT OF MEMORY!

Attention is All We Need?

Attention is good, but not effective for long docs, $O(n^2)$ is not efficient

Rethinking: Are all the tokens (words) necessary? Not really!

How?

1. Skimming [1]
2. Token pruning [2]
3. Token pruning + early exiting [3]
4. Efficient Attention

1. Block-Skim: Efficient Question Answering for Transformer
   [AAAI 2022] SJTU, University of Rochester

2. Learned Token Pruning for Transformers
   [arXiv 2021] UC Berkeley, Samsung

3. Magic Pyramid: Accelerating Inference with Early Exiting and Token Pruning
   [NIPS Workshop 2021] Monash University, Amazon

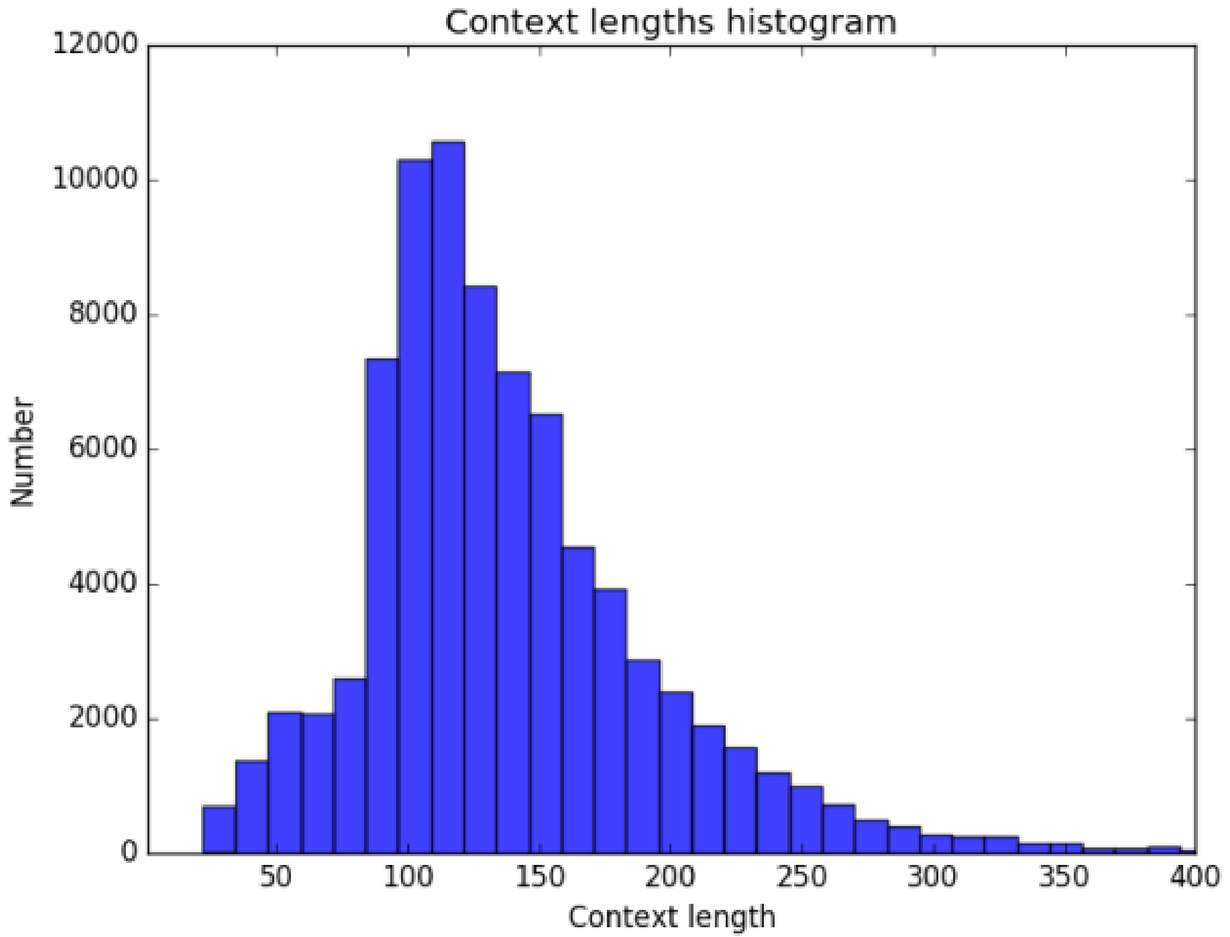
Method 1 — Block-skim [1]

[CLS] Who played quarterback for the Broncos after Peyton Manning was benched ? [SEP] Following their loss in the divisional round of the previous season 's playoffs , the Denver Broncos underwent numerous coaching changes, including a mutual parting with head coach John Fox (who had won four divisional championships in his four years as Broncos head coach), and the hiring of Gary Kubiak as the new head coach. under Kubiak, the Broncos planned to install a run - oriented offense with zone blocking to blend in with quarterback Peyton Manning's shotgun passing skills, but struggled with numerous changes and injuries to the offensive line, as well as manning having his worst statistical season since his rookie year with the Indianapolis Colts in 1998, due to a plantar fasciitis injury in his heel that he had suffered since the summer, and the simple fact that Manning was getting old, as he turned 39 in the 2015 off - season. Although the team had a 7 – 0 start , Manning led the NFL in interceptions. In week 10, Manning suffered a partial tear of the plantar fasciitis in his left foot. He set the NFL's all - time record for career passing yards in this game, but was benched after throwing four interceptions in favor of backup quarterback Brock Osweiler , who took over as the starter for most of the remainder of the regular season. Osweiler was injured, however, leading to Manning's return during the week 17 regular season finale, where the Broncos were losing 13 – 7 against the 4 – 11 San Diego Chargers, resulting in Manning re - claiming the starting quarterback position for the playoffs by leading the team to a key 27 – 20 win that enabled the team to clinch the number one overall AFC seed. Under defensive coordinator Wade Phillips, the Broncos'defense ranked number one in total yards allowed, passing yards allowed and sacks, and like the previous three seasons, the team has continued. [SEP]

- Attention map is effective for locating the answer position
- Use attention to predict what blocks to skim (and to keep)

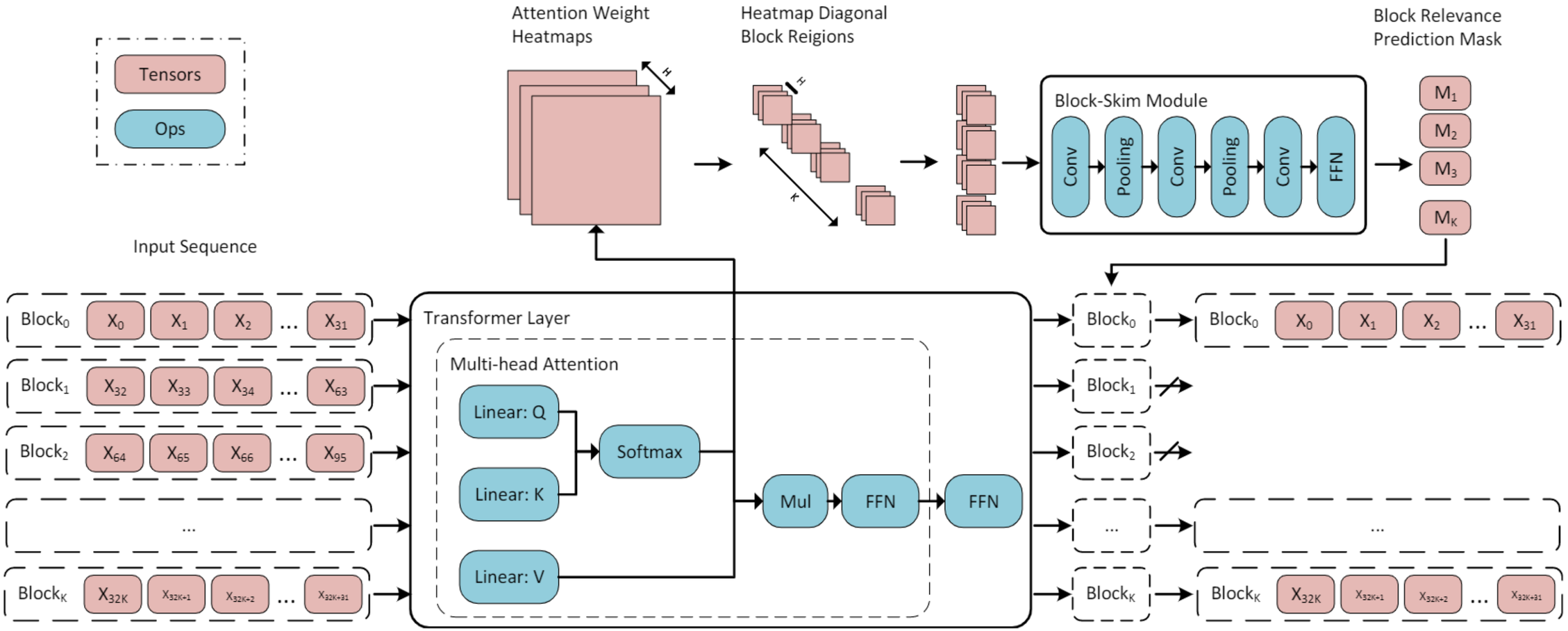
1. Block-Skim: Efficient Question Answering for Transformer [AAAI 2022] SJTU, University of Rochester

QA Context is Lengthy



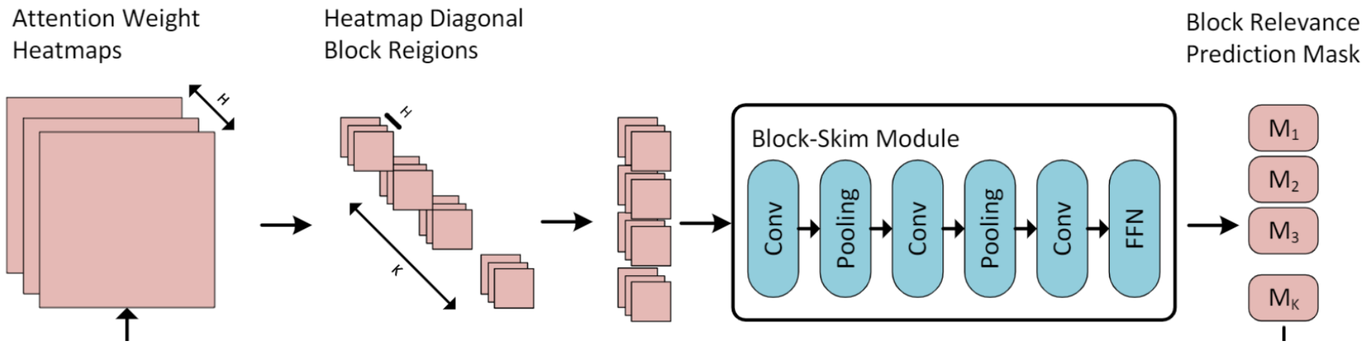
But most context are unnecessary!

Block-skim Architecture

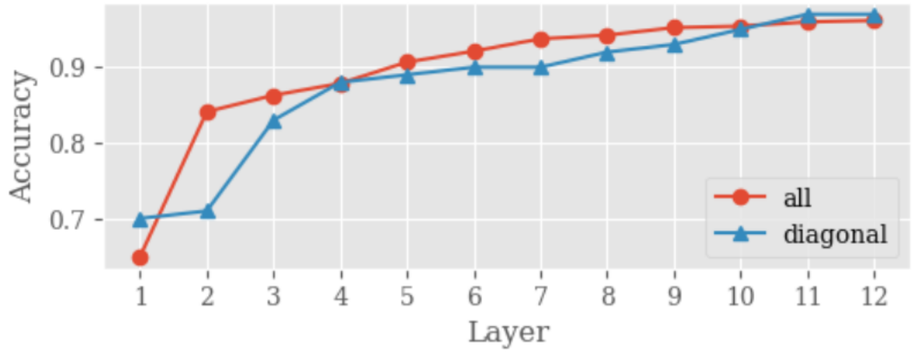


- Use CNN on attention map (`Softmax(QK)`) to calculate block importance
- Do not change Transformer architecture

CNN Based Block Relevance Prediction



Hypothesis: **diagonal region** of attention map contains sufficient information to identify the block relevance.



Rethinking: Diagonal region is intra-block. Are there inter-block features?

Training Block-skim

Single-task multi-objective

$$\mathcal{L}_{\text{BlockSkim}} = \sum_{m_i \in \{\text{blocks}\}} \text{CELoss}\left(m_i, y_i\right)$$

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{QA} + \alpha \sum_{l}^{\#layer} \left(\beta \mathcal{L}^{l,y=1}_{\text{BlockSkim}} + \mathcal{L}^{l,y=0}_{\text{BlockSkim}}\right)$$

1. No skiming when training
2. Does not affect the backbone model calculation
3. Block-skim Loss **improves** the original QA training
4. Seperate $y = 1$ or 0 and balance (β) cuz most blocks don't contain answer

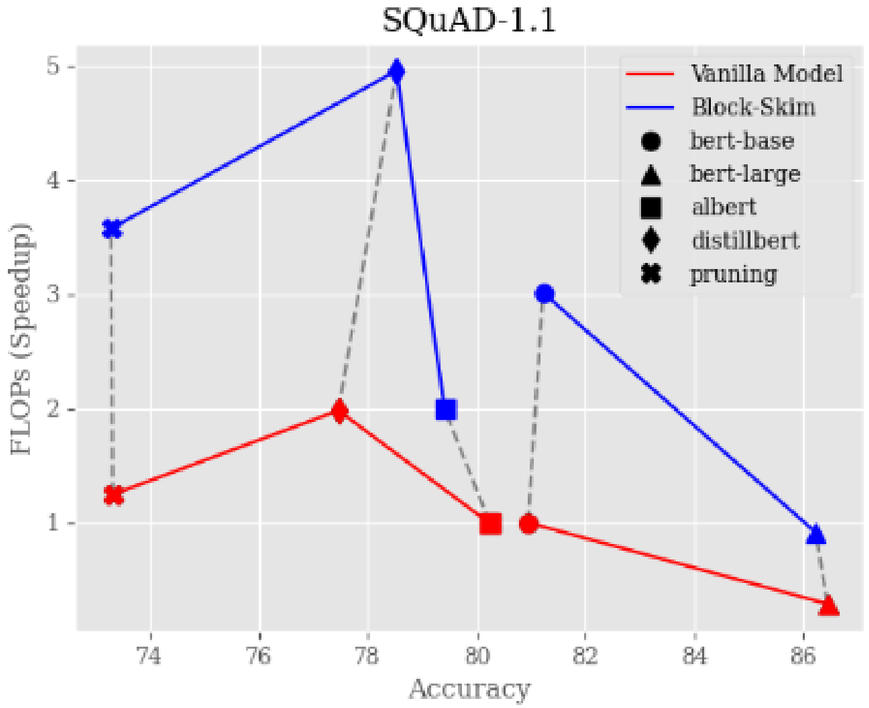
Block-skim Results — Score

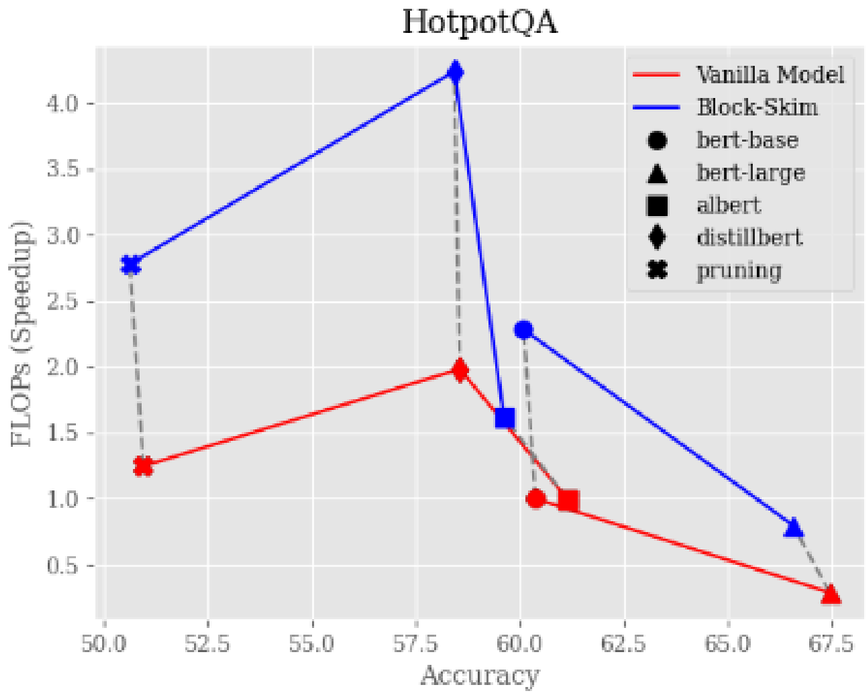
Datasets	SQuAD		HotpotQA		NewsQA		NaturalQuestions		TriviaQA		SearchQA		Avg.	
	F1	Speedup	F1	Speedup	F1	Speedup	F1	Speedup	F1	Speedup	F1	Speedup	F1	Speedup
Balance Factor	20		20		30		30		100		150		-	
Vanilla BERT	88.32	1×	74.39	1×	66.57	1×	78.85	1×	72.61	1×	79.93	1×	76.78	1×
Block-Skim Training	88.92	1×	74.88	1×	67.76	1×	78.98	1×	73.29	1×	80.32	1×	77.36	1×
Block-Skim Inference	88.52	3.01×	74.47	2.28×	65.14	2.53×	78.48	2.56×	72.80	1.81×	79.84	3.17×	76.54	2.56×
Deformer	87.2	3.1×	-	-	-	-	-	-	-	-	-	-	-	-
Length-Adaptive Transformer	88.7	2.22×	-	-	-	-	-	-	-	-	-	-	-	-

- Deformer preprocess and caches the context paragraphs at early layers to reduce the actual inference sequence length
- Length Adaptive Transformer is toekn pruning

Block-Skim objective is consistent with QA objective and improves its accuracy!

Block-skim Results — FLOPs





- Block-skim provides 2~3x reduction in FLOPs
- Accuracy drop is not significant
- Plug-and-play on various Transformer-based models

Block-skim Results — Ablation

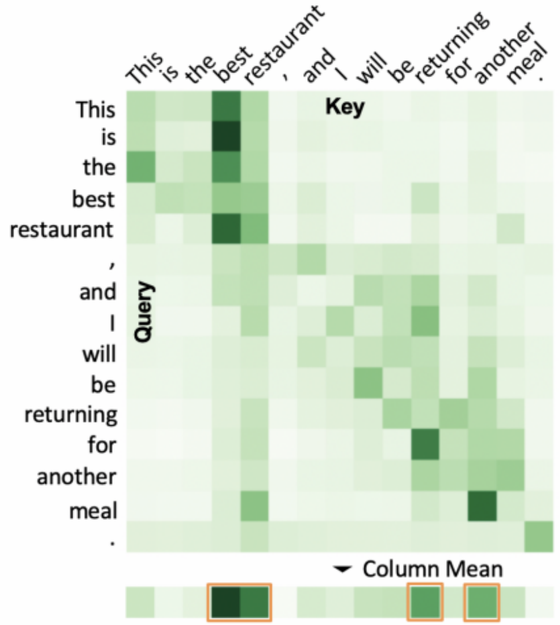
ID	Description	Update Transformer	Skim Training	Block-Skim Module	Block Size	QA	
						EM	F1
			SQuAD				
1	Baseline	✓	-	-	-	80.92	88.32
2	Block-Skim	✓		✓	32	81.52	88.92
3	Freeze Transformer			✓	32	80.92	88.32
4	Skim Traning	✓	✓	✓	32	79.27	86.83
5	Block Size 1	✓		✓	1	81.22	88.60
6	Block Size 8	✓		✓	8	81.25	88.63
7	Block Size 16	✓		✓	16	81.35	88.75
8	Block Size 64	✓		✓	64	81.39	88.65
9	Block Size 128	✓		✓	128	80.90	88.33
			HotpotQA				
10	Baseline	✓	-	-	-	60.37	74.39
11	Block-Skim	✓		✓	32	60.54	74.88
12	Evidence Loss	✓		✓	32	60.78	74.85

- 2 stage training is less effective (ID-3), Block-skim Loss is helpful
- Multi-hop compatability (ID-10~12)
  - ID-12 set supporting facts (i.e., evidence) to 1 in Block-skim Loss.

    Get higher score, but average accuracy of skim predictors (CNN) is worse (?)

    Thus, answer-only objective is enough (?)

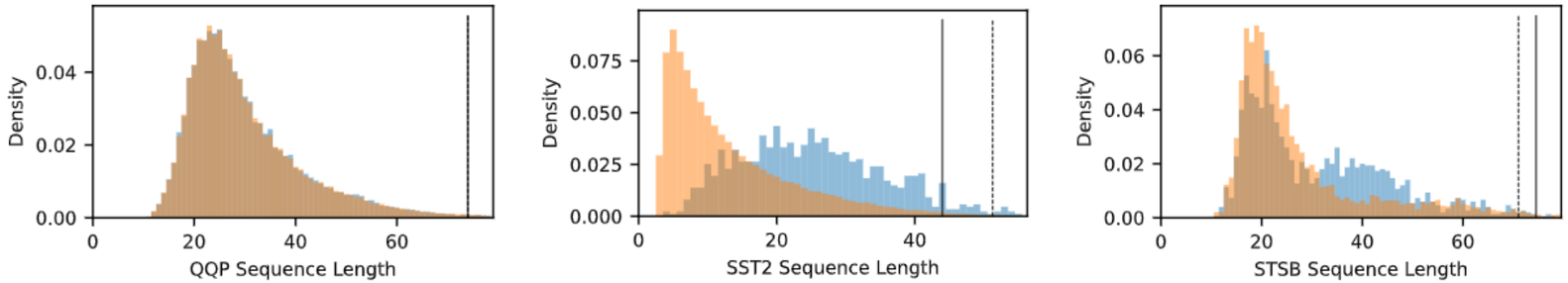
Method 2 — LTP [1]

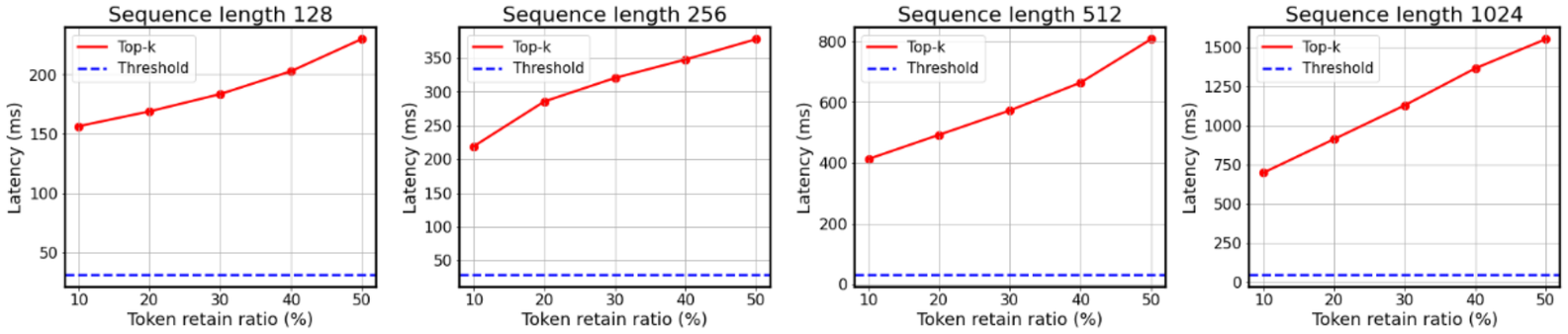


- A **simple, adaptive & robust threshold-based** token pruning method
- 2.10× FLOPs reduction w/ <1% accuracy degradation

1. Learned Token Pruning for Transformers
   [arXiv 2021] UC Berkeley, Samsung

Why Using Threshold?

1. **Adaptive**: PoWER-BERT, Length-Adaptive Transformer (LAT) use a fixed config throughout a dataset, **but sequence length varies a lot!**
   - 

2. **Efficient**: Above 2 & SpAtten, TR-BERT use top-k for token pruning, **which is much expensive!**
   - 

Threshold Token Pruning

Attention of token $\mathbf{x}_i$ and $\mathbf{x}_j$

$$\mathbf{A}^{(h,l)}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \text{softmax}\left(\frac{\left(\mathbf{x}^T \mathbf{W}_q^T\right)\left(\mathbf{W}_k \mathbf{x}\right)}{\sqrt{d}}\right)_{(i,j)}$$

The importance score of token x_i in layer l

$$s^{(l)}\left(\mathrm{x}_i\right) = \frac{1}{\#head}\frac{1}{\#token}\sum_{h=1}^{\#head}\sum_{j=1}^{\#token}\mathbf{A}^{(h,l)}\left(\mathrm{x}_i, \mathrm{x}_j\right)$$

Prune the token if $s^{(l)}\left(\mathrm{x}_i\right) < \theta^{(l)}$

Learning the Threshold

Hard mask

$$M^{(l)}\left(\mathrm{x}_i\right) = \begin{cases} 1 & , \text{if } \boldsymbol{s}^{(l)}\left(\mathrm{x}_i\right) > \boldsymbol{\theta}^{(l)} \\ 0 & , \text{otherwise} \end{cases}$$

No gradient, **Non**-differentiable, **Cannot** estimate gradient ✗

Soft mask

$$\tilde{M}^{(l)}\left(\mathrm{x}_i\right) = \sigma\left(\frac{\boldsymbol{s}^{(l)}\left(\mathrm{x}_i\right) - \boldsymbol{\theta}^{(l)}}{T}\right)$$

Has gradient, **Is** differentiable ✓

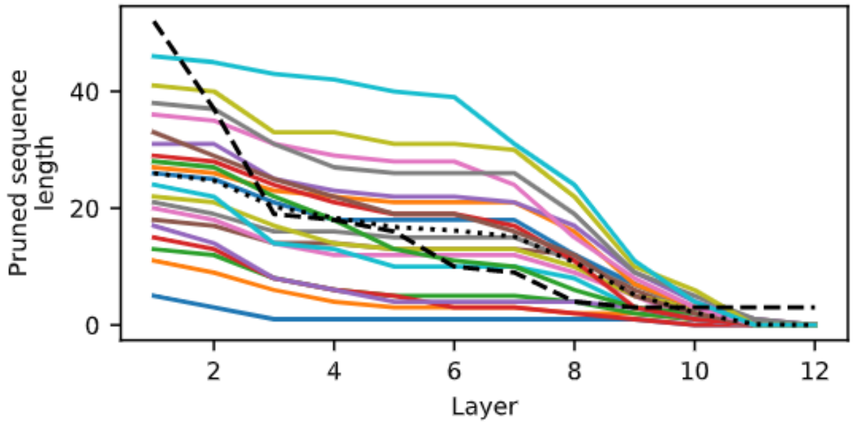
Training LTP

1. Finetune
2. Train finetuned model & thresholds with **SOFT** mask
3. Binarize the mask & fix the thresholds
4. Finetune only the model (**HARD** mask cannot train but OK to inference)

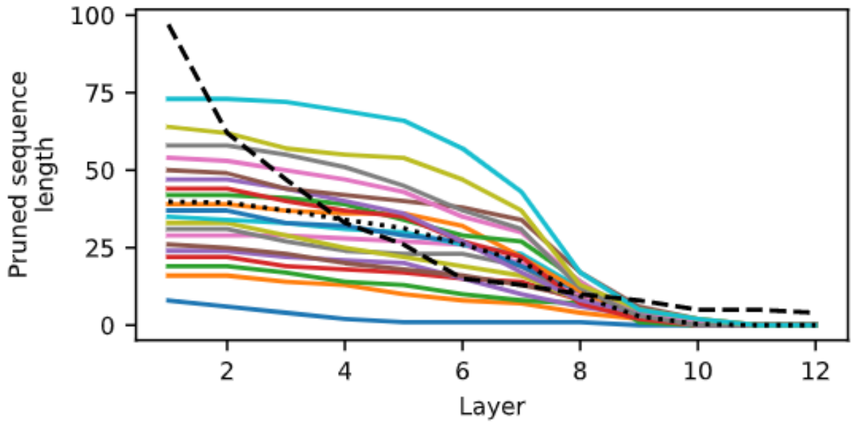
- Trick
  - Add L1 regularization to penalize the network if tokens are unpruned
  - $$\mathcal{L}_{\text{new}} = \mathcal{L} + \lambda \mathcal{L}_{\text{reg}} \text{ where } \mathcal{L}_{\text{reg}} = \frac{1}{\#layer} \sum_{l=1}^{\#layer} \left\| \tilde{M}^{(l)}(\mathbf{x}) \right\|_1$$
  - If much token unpruned, $\sum_{i=1}^{n} \tilde{M}^{(l)}(\mathbf{x}_i)$ is large, $\mathcal{L}_{\text{reg}}$ is large

LTP Results — Main

Task	Accuracy		GFLOPs		Speedup
	RoBERTa	LTP	RoBERTa	LTP	LTP
MNLI-m	87.53	86.53	6.83	3.64	1.88×
MNLI-mm	87.36	86.37	7.15	3.63	1.97×
QQP	90.39	89.69	5.31	2.53	2.10×
QNLI	92.86	91.98	8.94	4.77	1.87×
SST-2	94.27	93.46	4.45	2.13	2.09×
STS-B	90.89	90.03	5.53	2.84	1.95×
MRPC	92.14	91.59	9.33	4.44	2.10×
RTE	77.98	77.98	11.38	6.30	1.81×

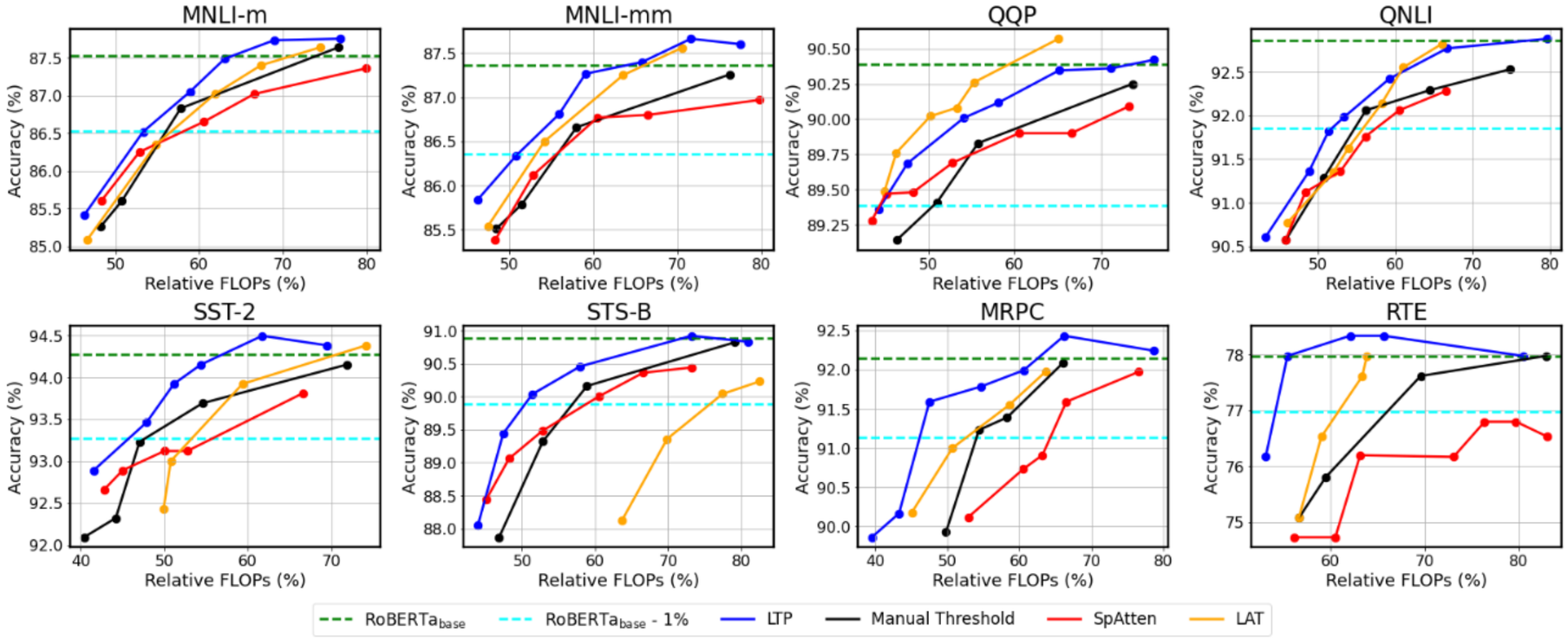


(a) *SST-2*



(b) *MNLI-m*

LTP Results — Comparative



- LAT is good when train set and dev set has similar length distribution (QQP)
- LTP is adaptive, but advantage is not very significant

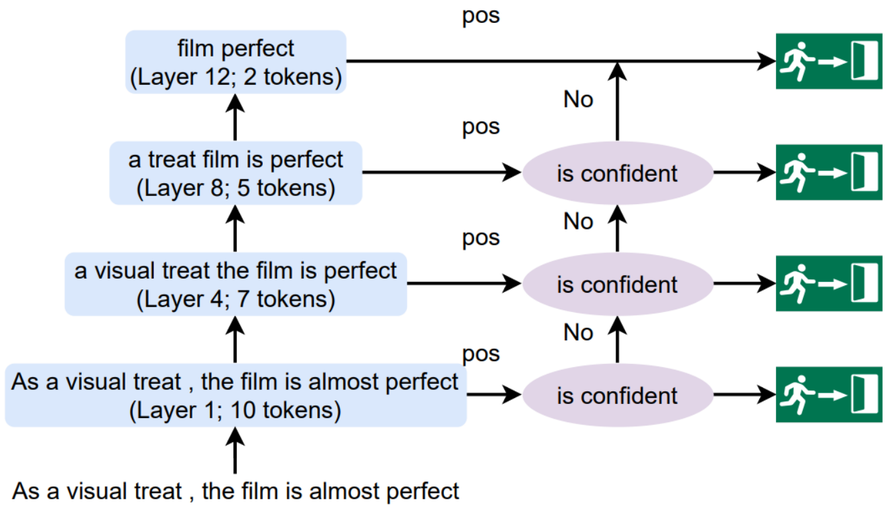
LTP V.S. LAT

Train on short samples (~Q2), eval on all length

	Task	QNLI			QQP		
		~Q2	Q2~Q3	Q3~	~Q2	Q2~Q3	Q3~
LTP	Acc.	91.21	90.02	91.81	89.42	89.51	91.37
	FLOPs	55.89	55.60	56.02	55.18	56.29	58.01
LAT	Acc.	90.87	86.12	75.37	89.20	87.27	82.17
	FLOPs	56.21	46.55	35.89	55.17	46.61	34.14
Diff.	Acc.	+0.34	+3.90	+16.44	+0.22	+2.24	+9.20

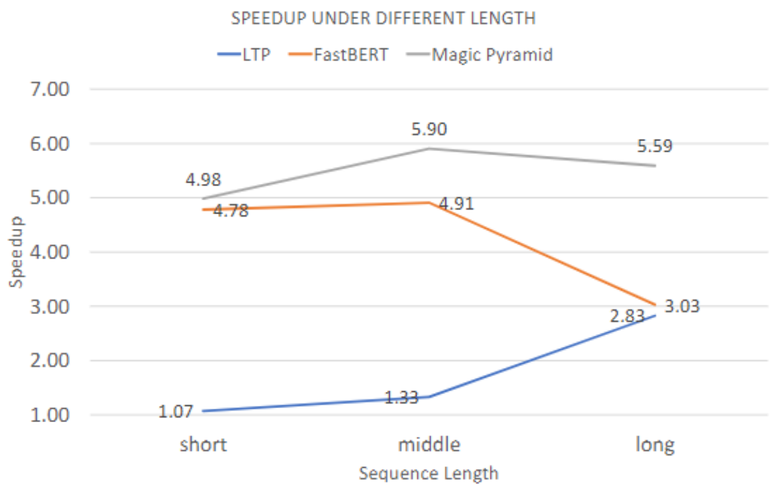
Rethinking: What about LAT training on long samples (Q3~)?

Method 3 — Magic Pyramid [1]



- **Token pruning** is good when sequence is **long**
- But, **early exit** is good when sequence is **short**
- Exploit the synergy!

Sequence Length



(a) Yelp.

(b) AG news

X-Axis	
short	1 ~ 35 tokens
middle	35 ~ 70 tokens
long	> 70 tokens

Recap: correlation between early exit and length - negative but loose! [1]

1. The Right Tool for the Job: Matching Model and Instance Complexities
   [ACL 2020] Allen AI, University of Washington

Training MP

1. Finetuning (vanilla **model**)
2. Soft mask training (**model** & thershold)
3. Hard mask training (**model** w/ hard mask)
4. Exit training (exits KLDiv Loss w/ hard mask)

Datasets	BERT	DistilBERT	LTP	FastBERT	MP (ours)
AG news	3,-,-,-	3,-,-,-	3,1,2,-	3,-,-,2	3,1,2,2
Yelp	3,-,-,-	3,-,-,-	3,1,2,-	3,-,-,2	3,1,2,2
QQP	5,-,-,-,	5,-,-,-,	5,2,5,-	5,-,-,5	5,2,5,5
MRPC	10,-,-,-,	10,-,-,-,	10,10,5,-	10,-,-,5	10,10,5,5
RTE	10,-,-,-,	10,-,-,-,	10,10,5,-	10,-,-,5	10,10,5,5

Table 2: The number of epochs used for regular training, soft pruning, hard pruning, subclassifiers training on different datasets. “-" indicates the corresponding stage is inactive.

Rethinking: training process is complex, is it FAIR? (same question to LTP)

MP Result — Comparative

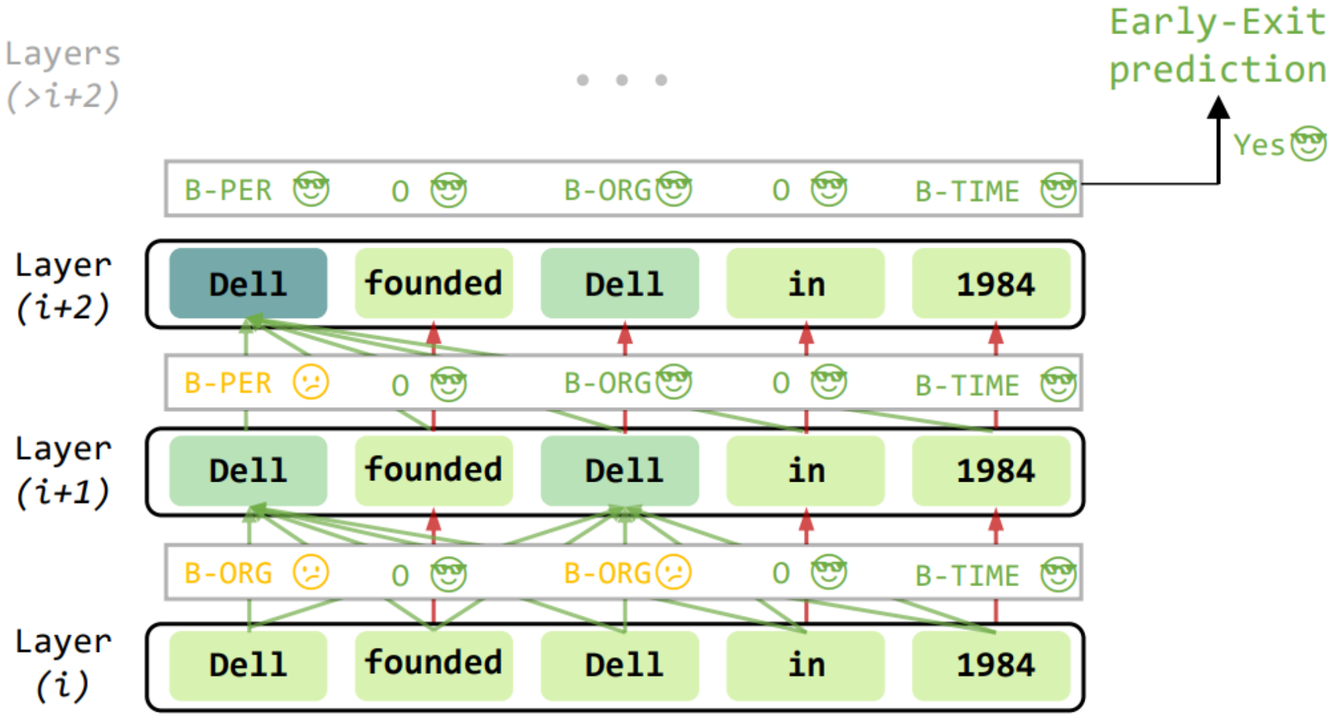
	AG news		Yelp		QQP		MRPC		RTE	
	Acc.	GFLOPs	Acc.	GFLOPs	Acc.	GFLOPs	Acc.	GFLOPs	Acc.	GFLOPs
BERT	94.3	9.0 (1.00x)	95.8	17.2 (1.00x)	91.3	5.1 (1.00x)	85.3	9.2 (1.00x)	68.6	11.2 (1.00x)
distilBERT	94.4	4.5 (2.00x)	95.7	8.6 (2.00x)	90.4	2.6 (2.00x)	84.6	4.6 (2.00x)	58.8	5.6 (2.00x)
LTP	94.3	5.3 (1.72x)	94.7	7.4 (2.32x)	90.6	3.2 (1.60x)	84.8	6.2 (1.48x)	67.8	7.5 (1.50x)
FastBERT	94.3	2.3 (3.97x)	94.8	2.8 (6.18x)	90.7	1.6 (3.20x)	84.3	4.3 (2.13x)	67.6	8.4 (1.33x)
MP (ours)	94.3	1.8 (4.95x)	94.5	2.1 (8.25x)	90.4	1.3 (4.03x)	83.8	3.3 (2.77x)	67.5	6.5 (1.72x)

	AG news			Yelp		
τ	0.1	0.5	0.8	0.1	0.5	0.8
FastBERT	3.97x	10.30x	11.95x	3.15x	6.18x	8.84x
MP (ours)	4.95x	10.53x	11.95x	5.35x	8.25x	10.10x

Table 4: Speedup of FastBERT and MP with different τ.

MP benefits both from token pruning and early exiting

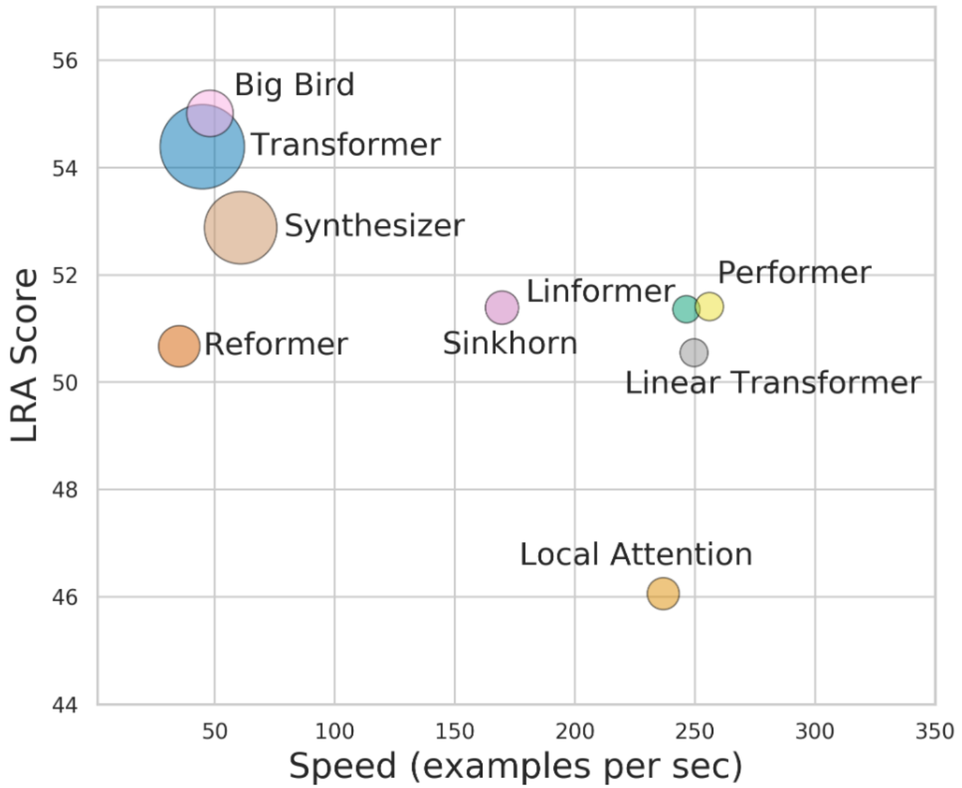
Extend — TOKEE [1]



(c) Token-Level Early-Exit for Sequence Labeling

1. Accelerating BERT Inference for Sequence Labeling via Early-Exit [ACL 2021] FDU

Method 4 — Efficient Attention [1]

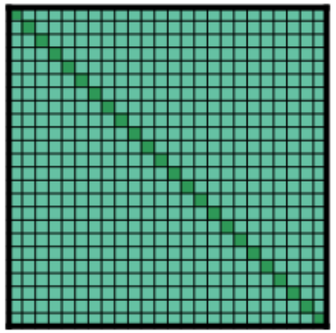


1. X-axis is speed
2. Y-axis is performance
3. Circle is memory footprint

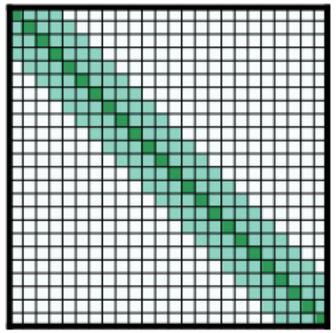
---

1. Long Range Arena: A Benchmark for Efficient Transformers [ICLR 2021] Google

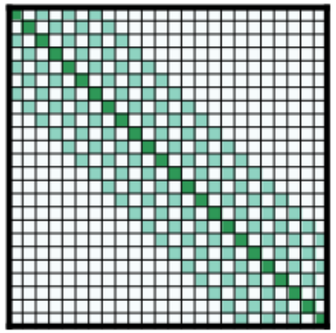
Efficient Attention Illustrated



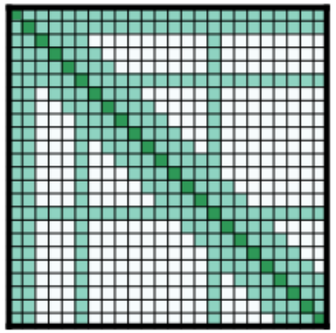
(a) Full n^2 attention



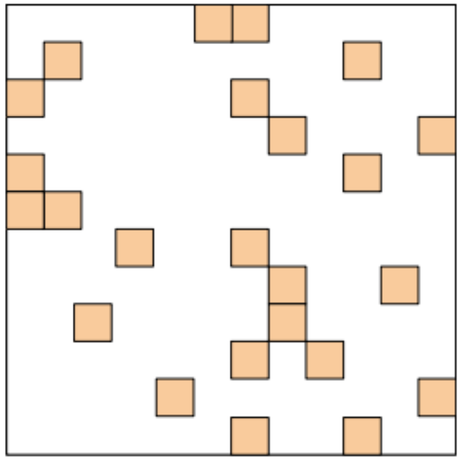
(b) Sliding window attention



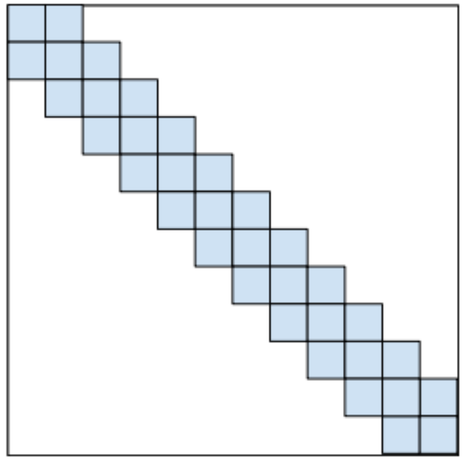
(c) Dilated sliding window



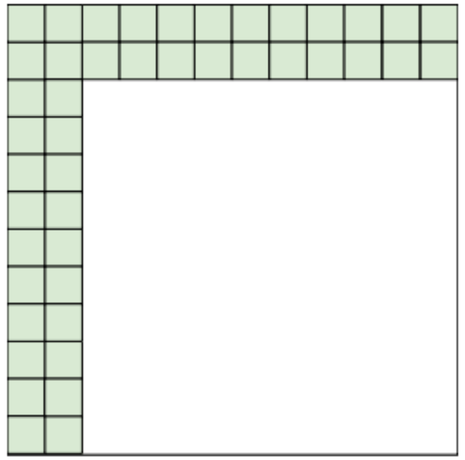
(d) Global+sliding window



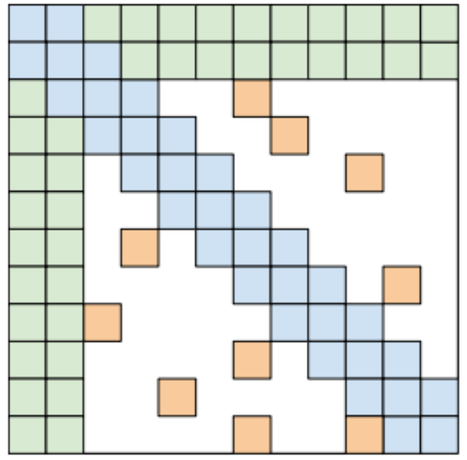
(a) Random attention



(b) Window attention



(c) Global Attention



(d) BIGBIRD

Recap

1. Attention
    1. Redundancy exist, $O(n^2)$ complexity is high
    2. Less effective and efficient on long document
2. Token importance
    1. Attention distribution shows importance
    2. Choosing granularity (QA->block, classification->token)
    3. Supervised learning a mask function
3. Training
    1. Pseudo skiming/pruning when training, real when inferencing
    2. Joint training is generally good