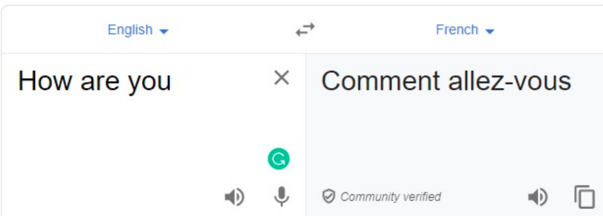
E.T.: Re-Thinking Self-Attention for Transformer Models on GPUs

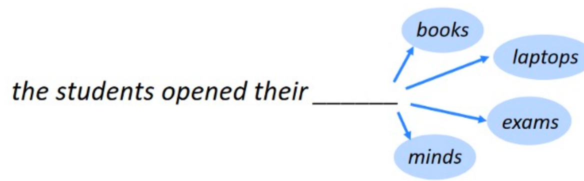
Shaoyi Huang
University of Connecticut

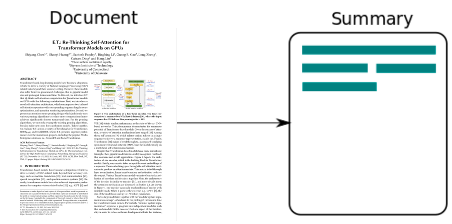
Outline

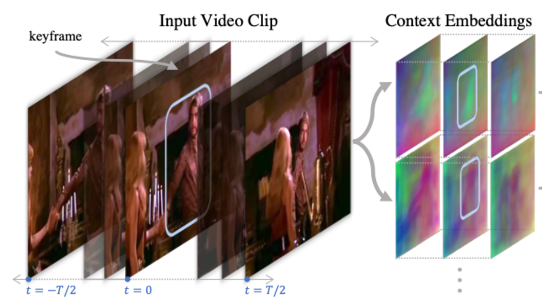
- Motivation
- Challenge #1: Long turnaround time
- Challenge #2: Gigantic model size
- Technique #1: Self-Attention Primitives
- Technique #2: Attention-Aware, Tensor-Core Friendly Pruning
- Evaluation
- Conclusion

Motivation: Sequence-based Problems is Everywhere …









[1]. Girdhar, Rohit, et al. "Video action transformer network." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recog

Attention is All You Need!

Passage Sentence

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity.

Question

What causes precipitation to fall?

Answer Candidate

gravity

Question Answering on SQuAD2.0 dev

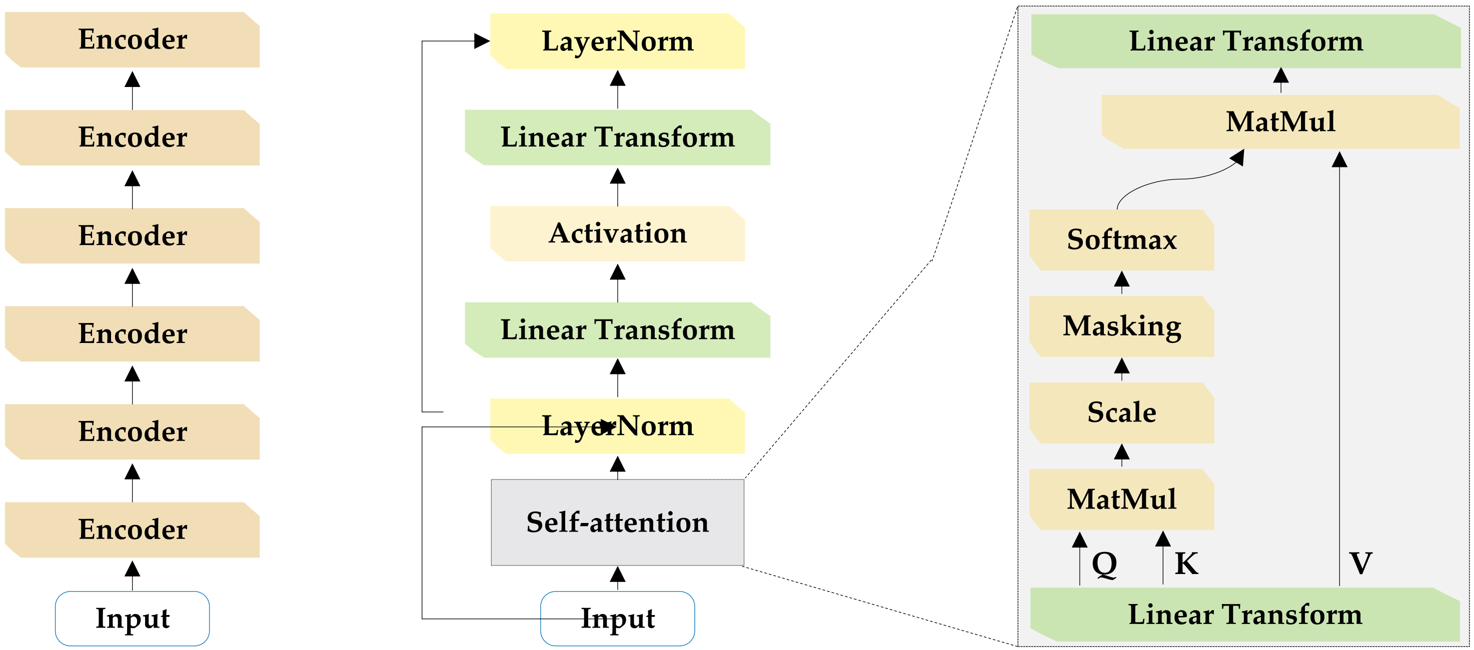
Rank	Model	F1	EM	Extra Training Data	Paper	Code	Result	Year
1	**XLNet** (single model)	90.6	87.9	×	XLNet: Generalized Autoregressive Pretraining for Language Understanding			2019
2	**XLNet+DSC**	89.51	87.65	×	Dice Loss for Data-imbalanced NLP Tasks			2019
3	**RoBERTa** (no data aug)	89.4	86.5	✓	RoBERTa: A Robustly Optimized BERT Pretraining Approach			2019

Semantic Textual Similarity on MRPC

Rank	Model	Accuracy	F1	Paper	Code	Result	Year
1	**SMART-RoBERTa Large**	93.7%		SMART: Robust and Efficient Fine-Tuning for Pre-trained Natural Language Models through Principled Regularized Optimization			2019
2	**ALBERT**	93.4%		ALBERT: A Lite BERT for Self-supervised Learning of Language Representations			2019
3	**RoBERTa**	92.3%		RoBERTa: A Robustly Optimized BERT Pretraining Approach			2019

[1]. https://paperswithcode.com/sota/question-answering-on-squad20-dev
[2]. https://paperswithcode.com/sota/semantic-textual-similarity-on-mrpc

Challenge #1. Long Turnaround Time

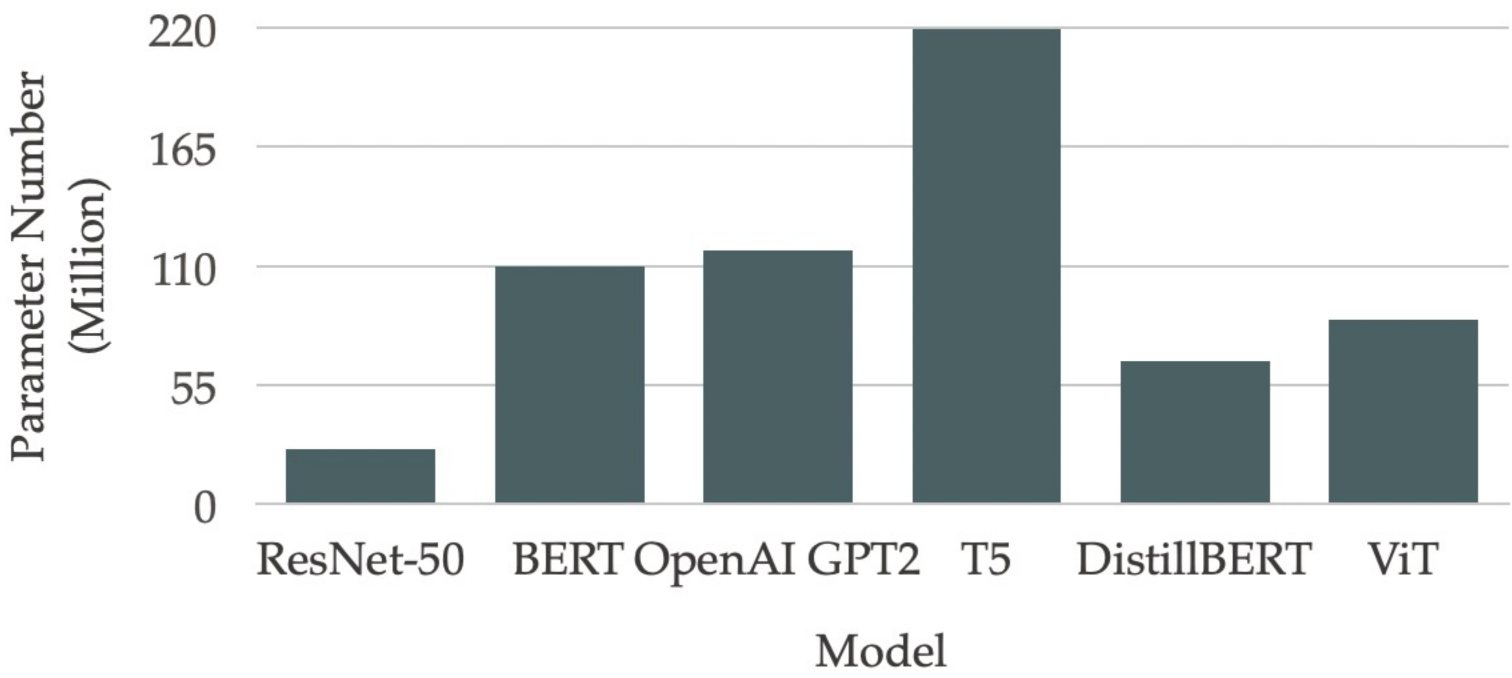


Model workflow Encoder workflow Self-attention workflow

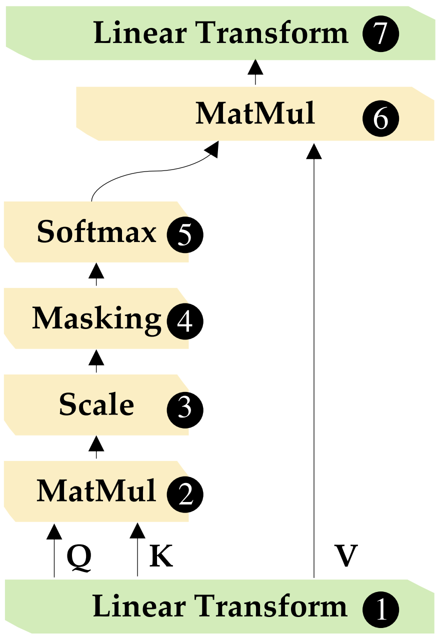
Challenge #1. Long Turnaround Time

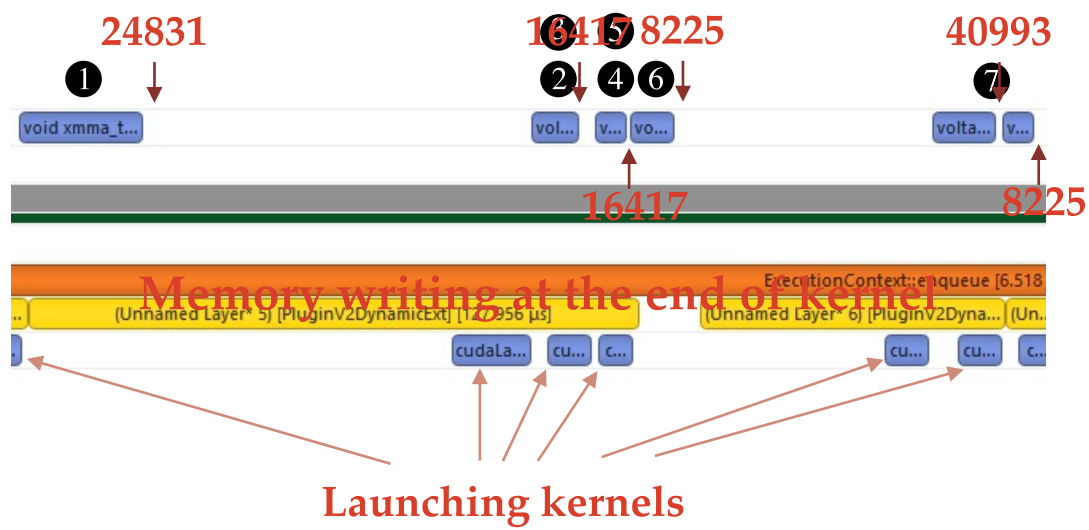
[1]. https://github.com/NVIDIA/TensorRT/tree/master/demo/BERT

Challenge #2. Gigantic model size

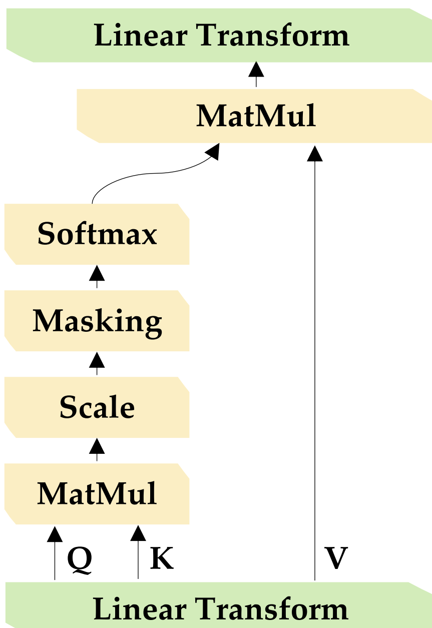


Kernels are not free





Think self-attention as a primitive

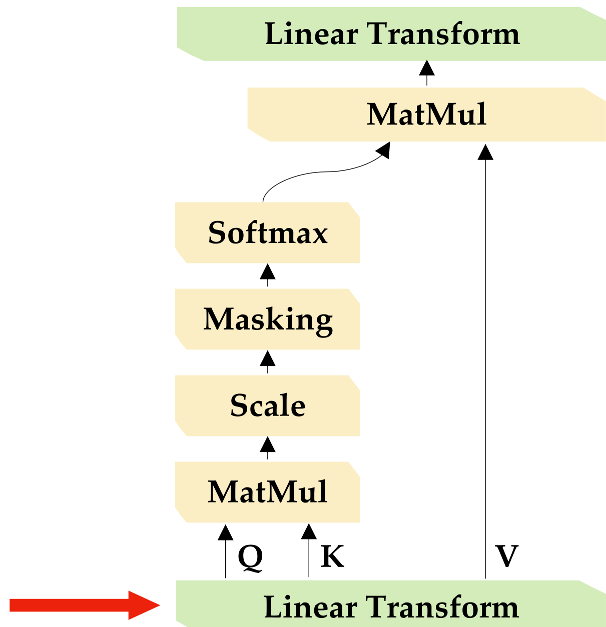


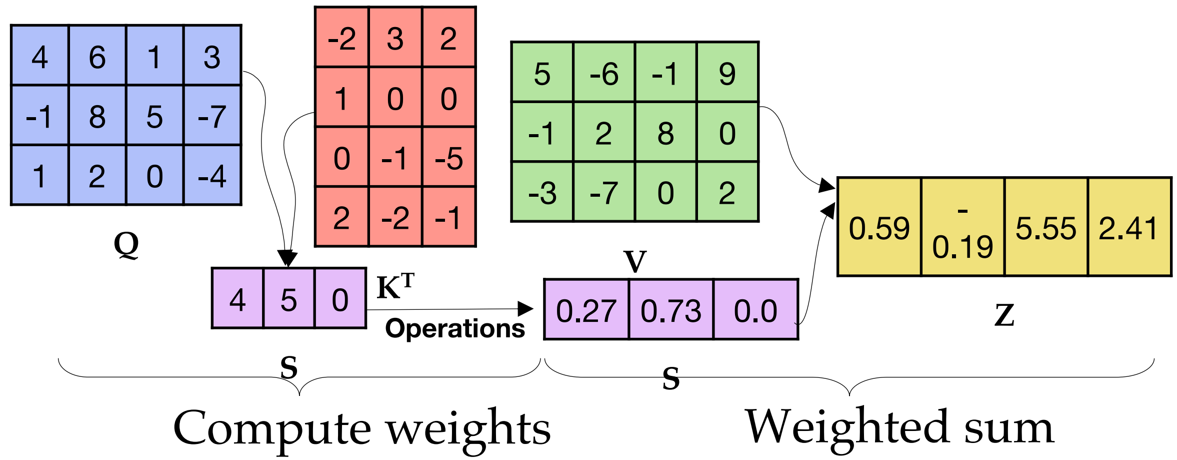
How are you →

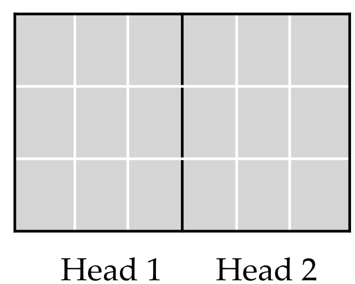
how	5	2	1	4
are	-3	-1	0	0
you	1	-2	-1	-1

Input

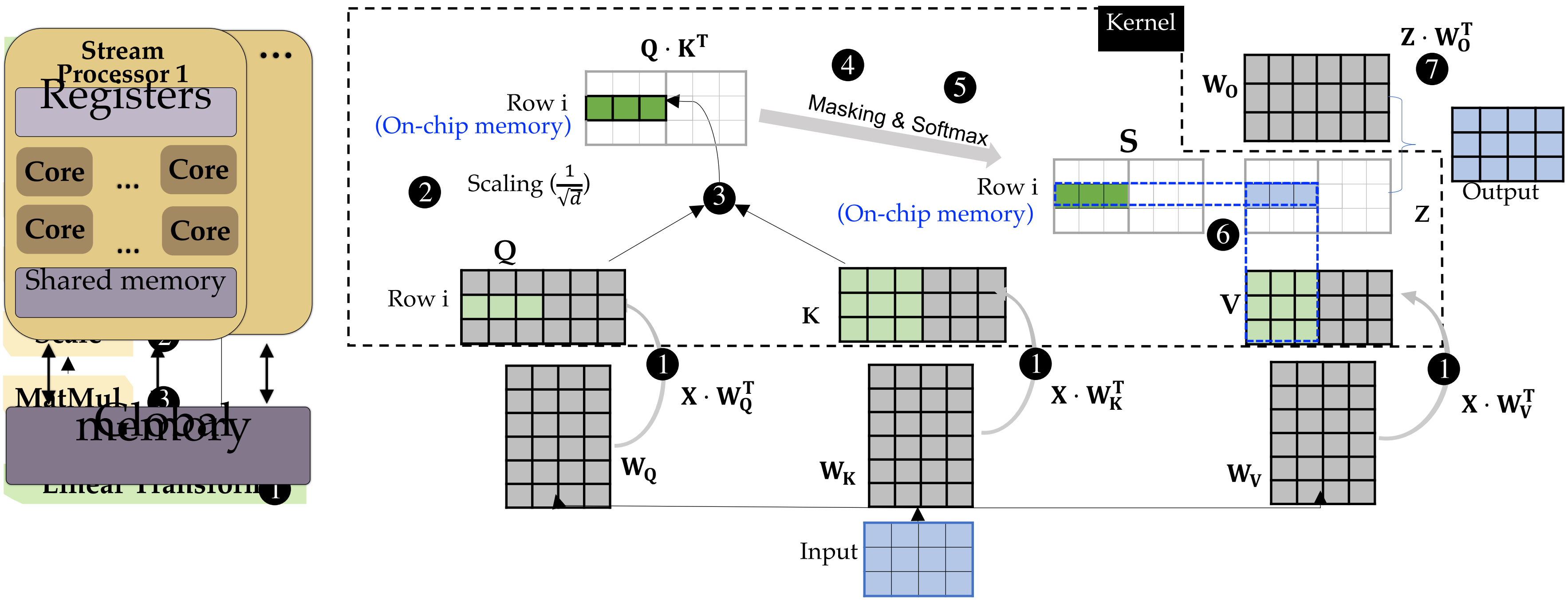
Think self-attention as a primitive



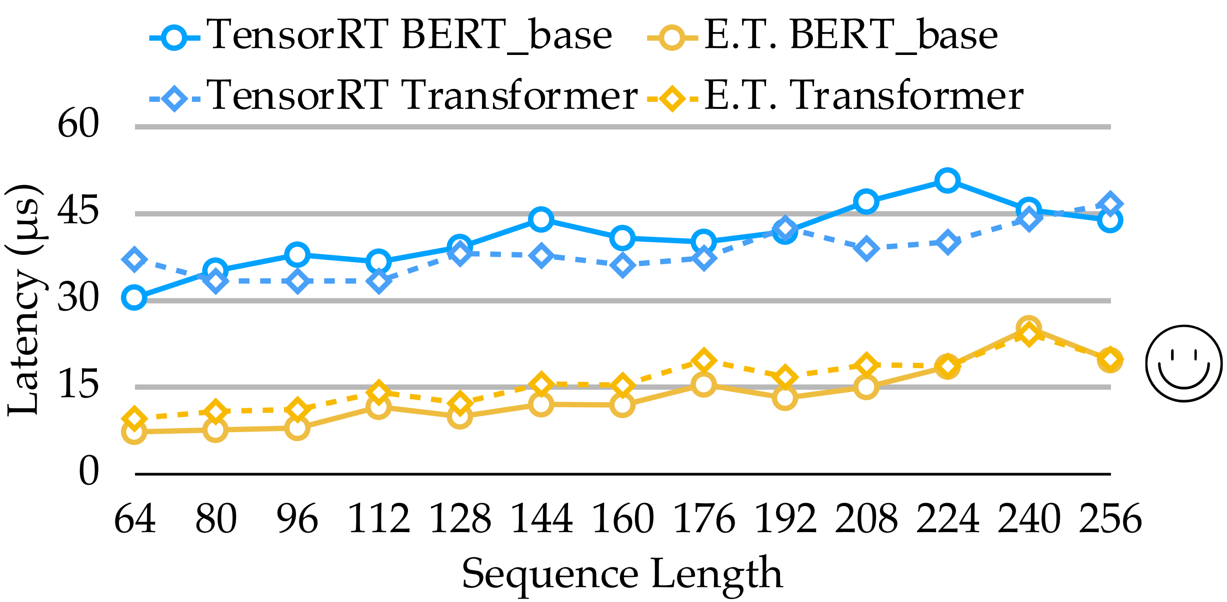




Compute the self-attention on-the-fly



Evaluate on-the-fly-attention



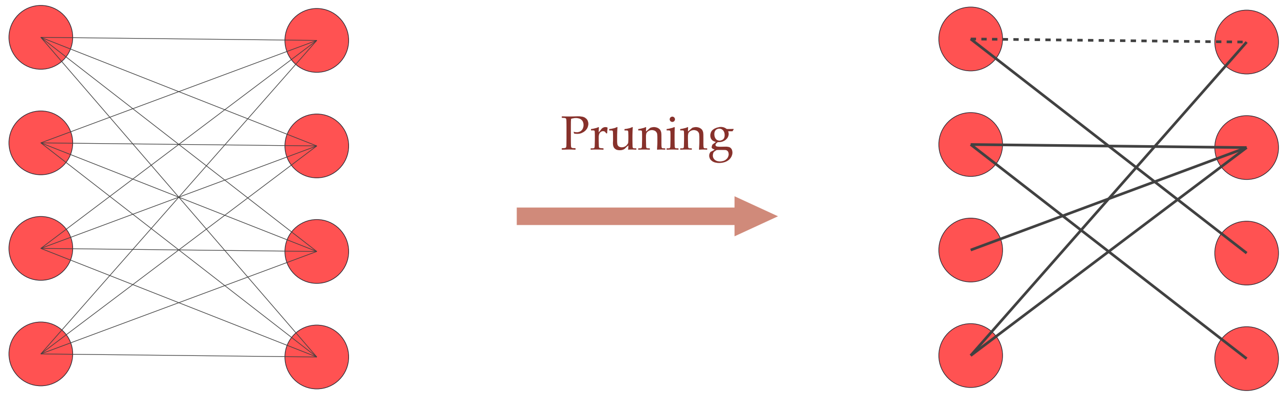
BERT_base:

- Model Size: 768
- Number of heads: 12

Transformer:

- Model Size: 800
- Number of heads: 4

Pruning makes the model small



Dense

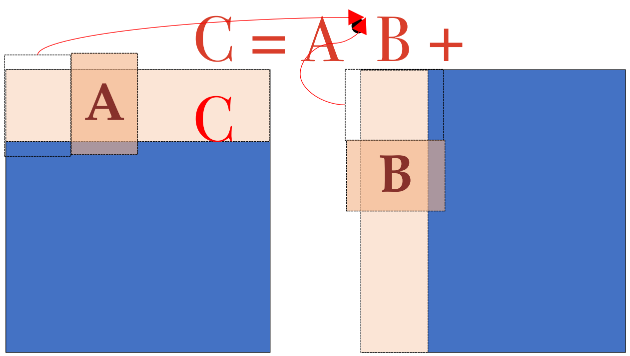
4	3	6	9
8	7	6	2
4	8	3	2
5	2	4	9

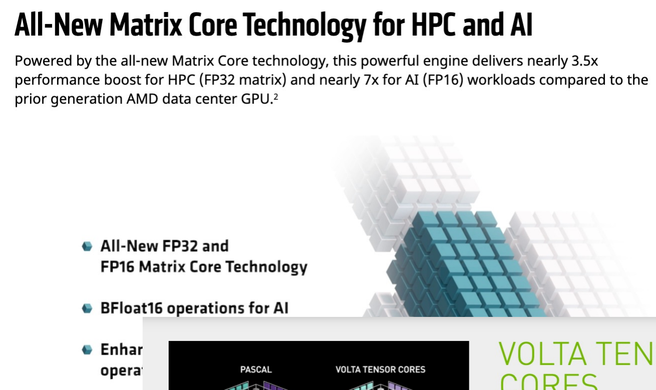
Pruning

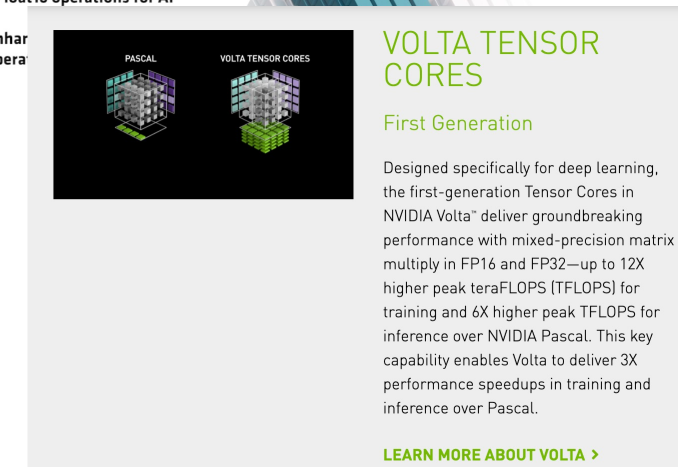
Sparse

0	0	6	0
0	7	0	2
0	8	0	0
5	2	0	0

Using emerging hardware







[1]. https://www.amd.com/en/technologies/cdna
[2]. https://www.nvidia.com/en-us/data-center/tensor-cores

Efficient computing on sparse models

Irregular

0	0	6	0
0	7	0	2
0	8	0	0
5	2	0	0

Row

4	3	6	9
0	0	0	0
4	8	3	2
0	0	0	0

Column pruning

4	0	6	0
8	0	6	0
4	0	3	0
5	0	4	0

Tensor-tile Pruning

0	0	6	9
0	0	6	2
4	8	0	0
5	2	0	0

W

4	3	6	9
0	0	0	0
4	8	3	2
0	0	0	0

→ $\mathbf{W}_{\text{pruned}}$

4	3	6	9
4	8	3	2

X

1	0	-2	0
-3	1	1	1
0	-1	1	0
2	0	-3	1

× $\mathbf{W}^{T}_{\text{pruned}}$

4	4
3	8
6	3
9	2

→

-8	0	-2	0
6	0	1	0
3	0	-5	0
-1	0	1	0

X

1	0	-2	0
-3	1	1	1
0	-1	1	0
2	0	-3	1

W

4	0	6	0
8	0	6	0
4	0	3	0
5	0	4	0

→

4	6
8	6
4	3
5	4

X (adjusted)

1	-2
-3	1
0	1
2	-3

× $\mathbf{W}^{T}_{\text{pruned}}$

4	8	4	5
6	6	3	4

→

1	0	-2	0
-3	1	1	1
0	-1	1	0
2	0	-3	1

Prune the model as fine-tuning

1.1	2.1	1.7	1.2
0.5	2.3	0.4	1.5
0.9	1.8	0.1	3.3
0.8	0.1	1.8	5.1
0.7	0.6	1.5	3.6
1.6	0.7	0.8	0.9
0.6	2.1	2.6	0.5
2.1	1.3	1.4	0.3

L2 norm →

3.3	2.6
2.2	6.3
2.0	4.0
3.3	3.0

→ Change ratio

1.2	1.3
1.1	0.9
0.7	1.5
0.9	0.7

← L2 norm of last iteration

2.6	2.0
2.1	6.6
2.7	2.6
3.5	4.1

$$\underbrace{\min f(\{W^k\}_{k=1}^{N},\{b^k\}_{k=1}^{N})}_{\text{Original loss}} + \underbrace{\lambda \sum_{k=1}^{N}\sum_{i=1}^{p}\sum_{j=1}^{q} \frac{\| W_{ij}^{k} \|_2}{\| W_{ij}^{k-1} \|_2 + \epsilon}}_{\text{Regularizer}}$$

Prune the model as fine-tuning

0.8	0.6	1.1	0.3
1.2	2.1	-0.4	1.2
1.3	-0.5	-0.1	3.6
0.4	1.5	2.1	5.1
1.6	1.1	1.7	1.2
0.6	1.8	0.4	1.5
2.3	1.5	0.1	3.6
0.7	2.1	0.6	1.8

0.8	0.6	1.1	0.3
1.2	2.1	-0.4	1.2
1.3	-0.5	-0.1	3.6
0.4	1.5	2.1	5.1
1.6	1.1	1.7	1.2
0.6	1.8	0.4	1.5
2.3	1.5	0.1	3.6
0.7	2.1	0.6	1.8

L2 norm →

3.3	
	6.4
	4.1
3.3	

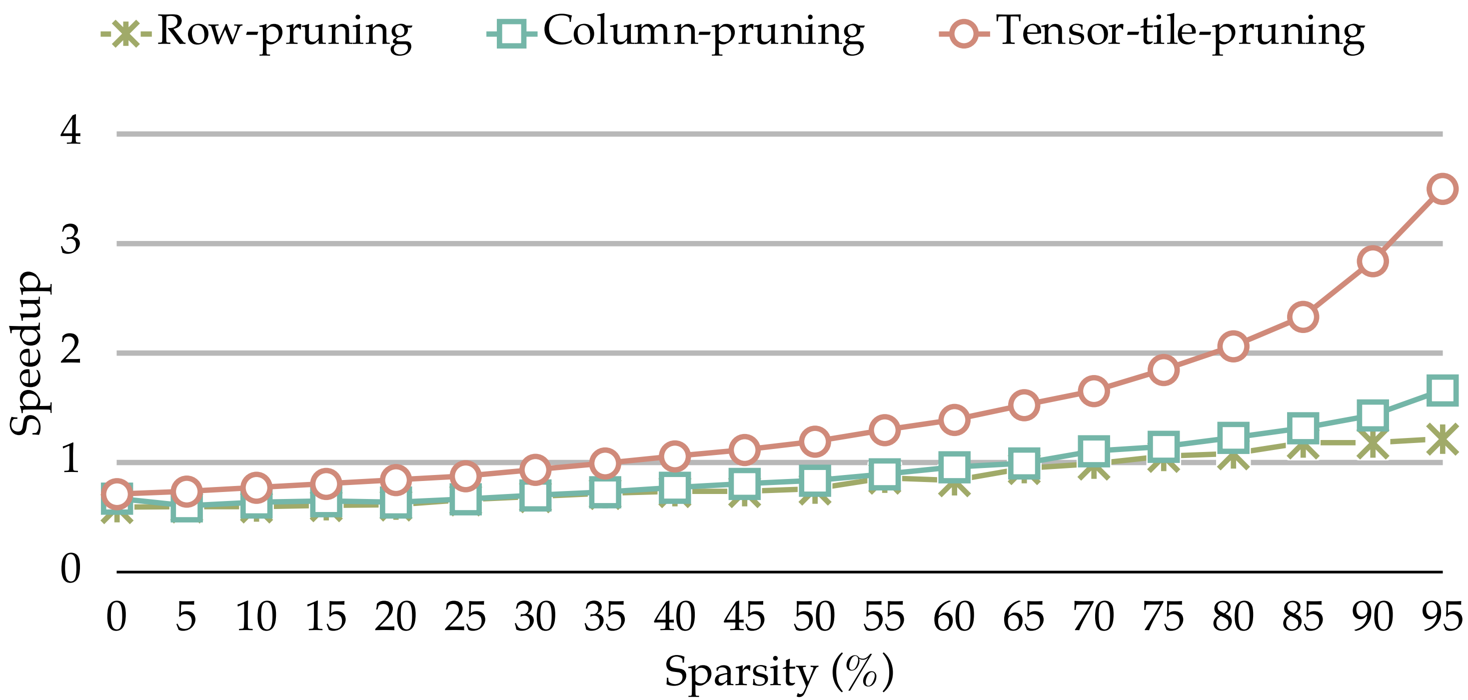
Prune →

0.8	0.6	0	0
1.2	2.1	0	0
0	0	-0.1	3.6
0	0	2.1	5.1
0	0	1.7	1.2
0	0	0.4	1.5
2.3	1.5	0	0
0.7	2.1	0	0

→

0.8	0.6	0	0
1.2	2.1	0	0
0	0	-0.1	3.6
0	0	2.1	5.1
0	0	1.7	1.2
0	0	0.4	1.5
2.3	1.5	0	0
0.7	2.1	0	0

Performance gain from pruning



Efficient computing on sparse models

Irregular

0	0	6	0
0	7	0	2
0	8	0	0
5	2	0	0

Row

4	3	6	9
0	0	0	0
4	8	3	2
0	0	0	0

Column pruning

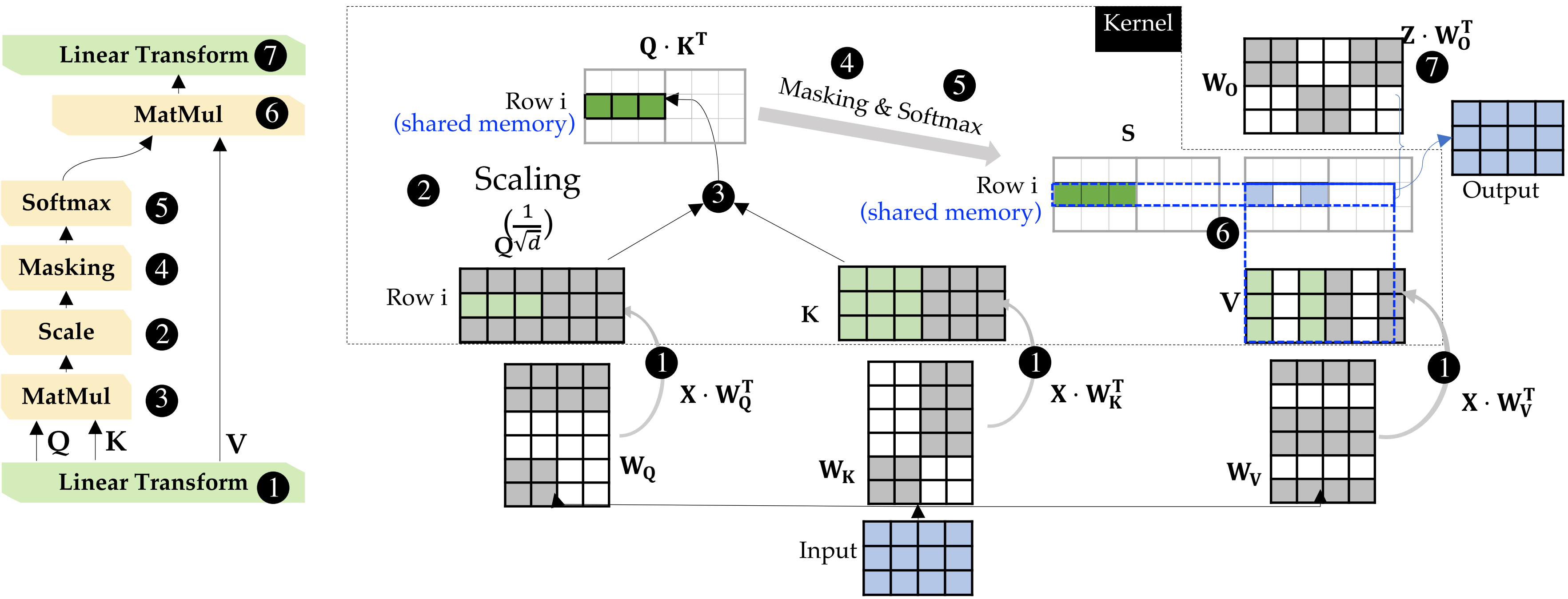
4	0	6	0
8	0	6	0
4	0	3	0
5	0	4	0

Tensor-tile Pruning

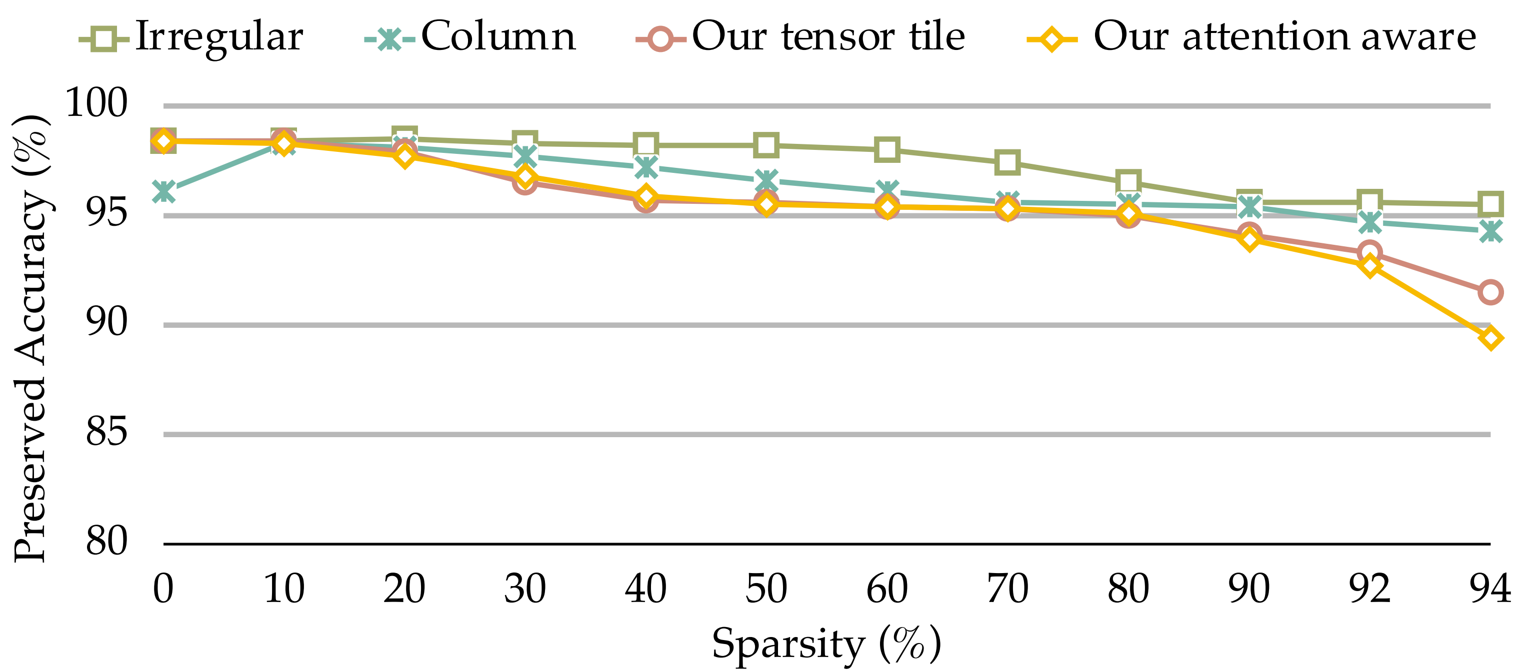
0	0	6	9
0	0	6	2
4	8	0	0
5	2	0	0

	Irregular	Row	Column	Tensor-tile
Accuracy	🙂	☹	😑	😑
Latency	☹	😑	😑	🙂

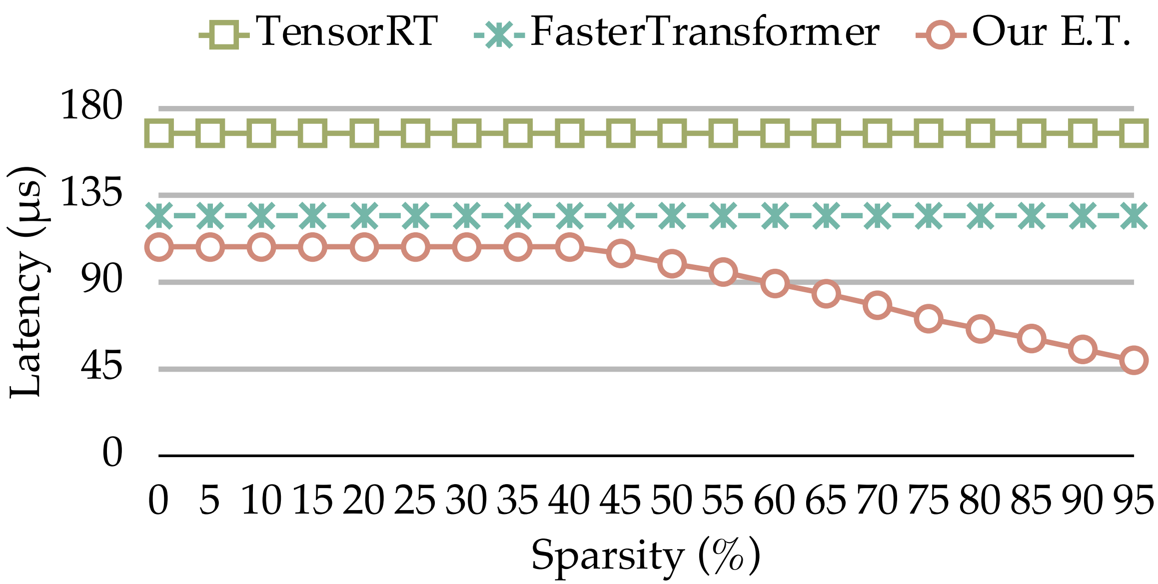
Attention-aware pruning



Evaluate pruning algorithms



Compare with state-of-the-art



Conclusion

- We design a novel self-attention architecture with 2.5x speedup compared with TensorRT
- We introducing tensor-tile pruning algorithms and model-aware pruning.
- E.T. is available at:
  - https://github.com/cctry/E.T.

Conclusion

Thank You & Questions?