FACT: FFN-Attention Co-optimized Transformer Architecture with Eager Correlation Prediction

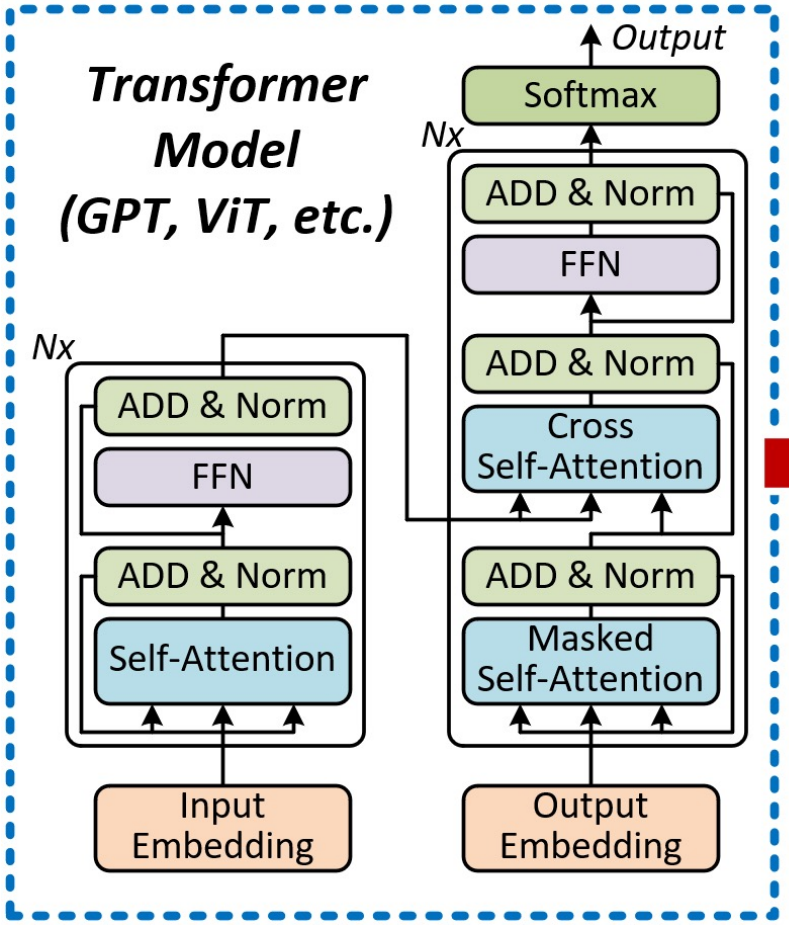
Yubin Qin, Yang Wang, Dazheng Deng, Zhiren Zhao, Xiaolong Yang,
Leibo Liu, Shaojun Wei, Yang Hu, Shouyi Yin

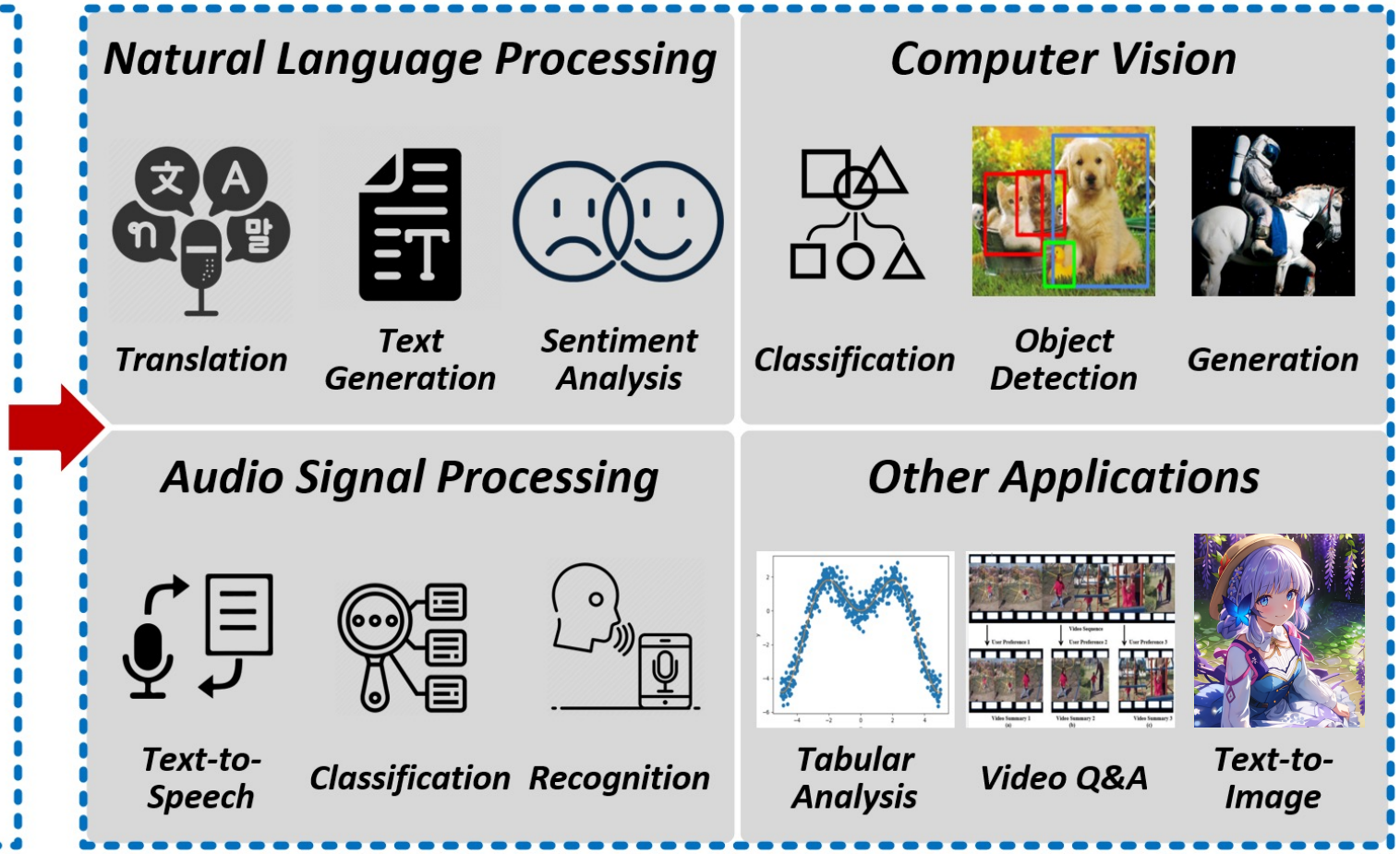
School of Integrated Circuits
BNRist, Tsinghua University

Outline

- **Background and Motivation**
- Eager Prediction Mechanism
- Hardware Innovation
- Evaluation

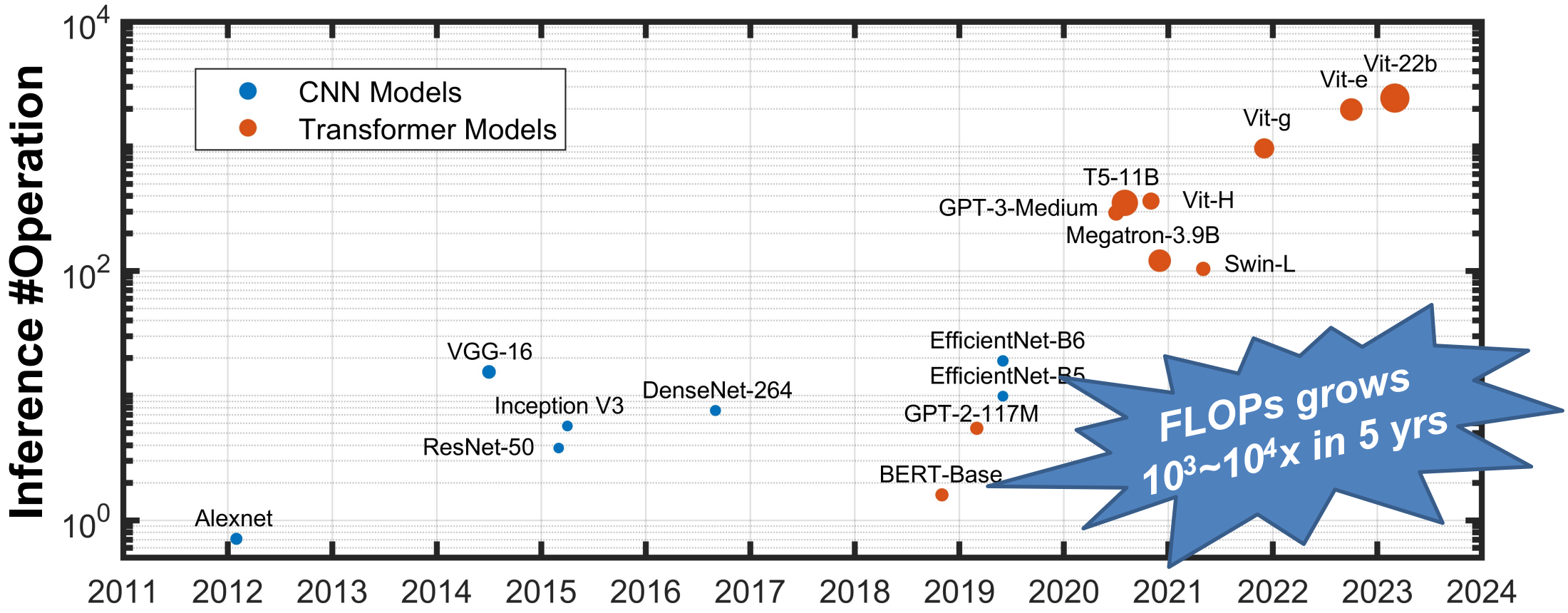
The Outstanding Transformer Model





Transformers beat traditional models in various fields.

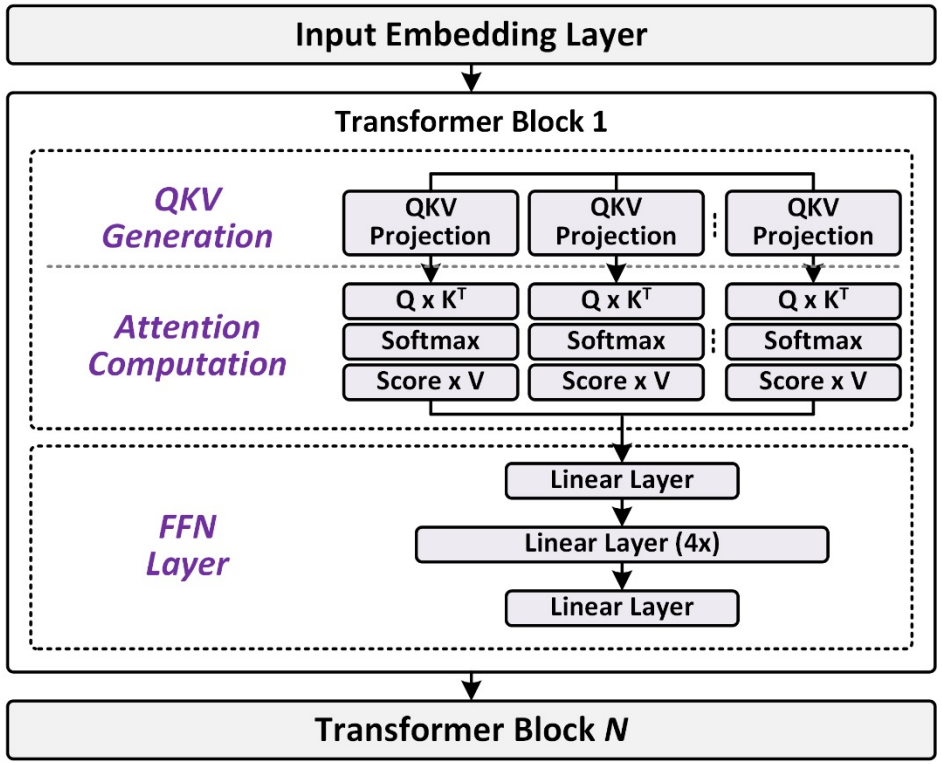
The Outstanding Transformer Model

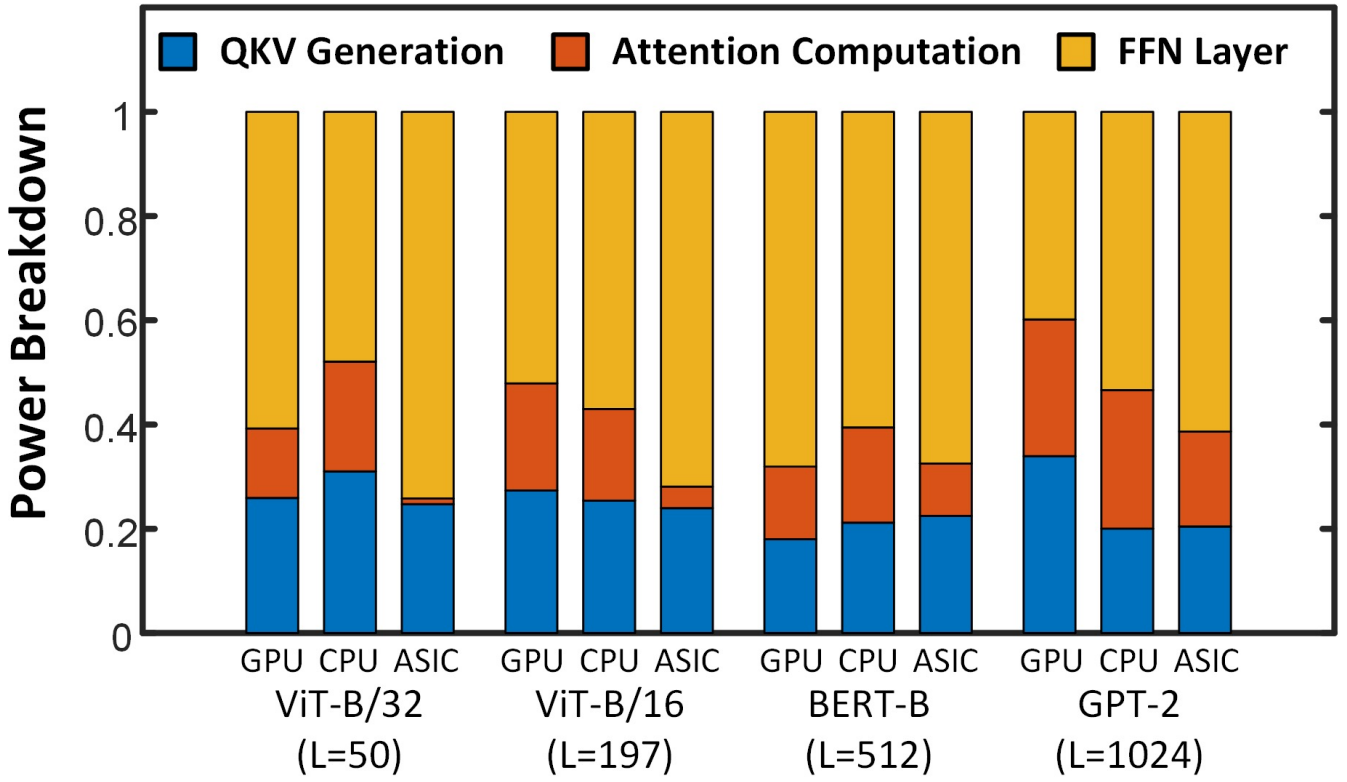


At the cost of explosive computation power

Power Breakdown of Transformer

- The power consumption of linear layer cannot be ignored!

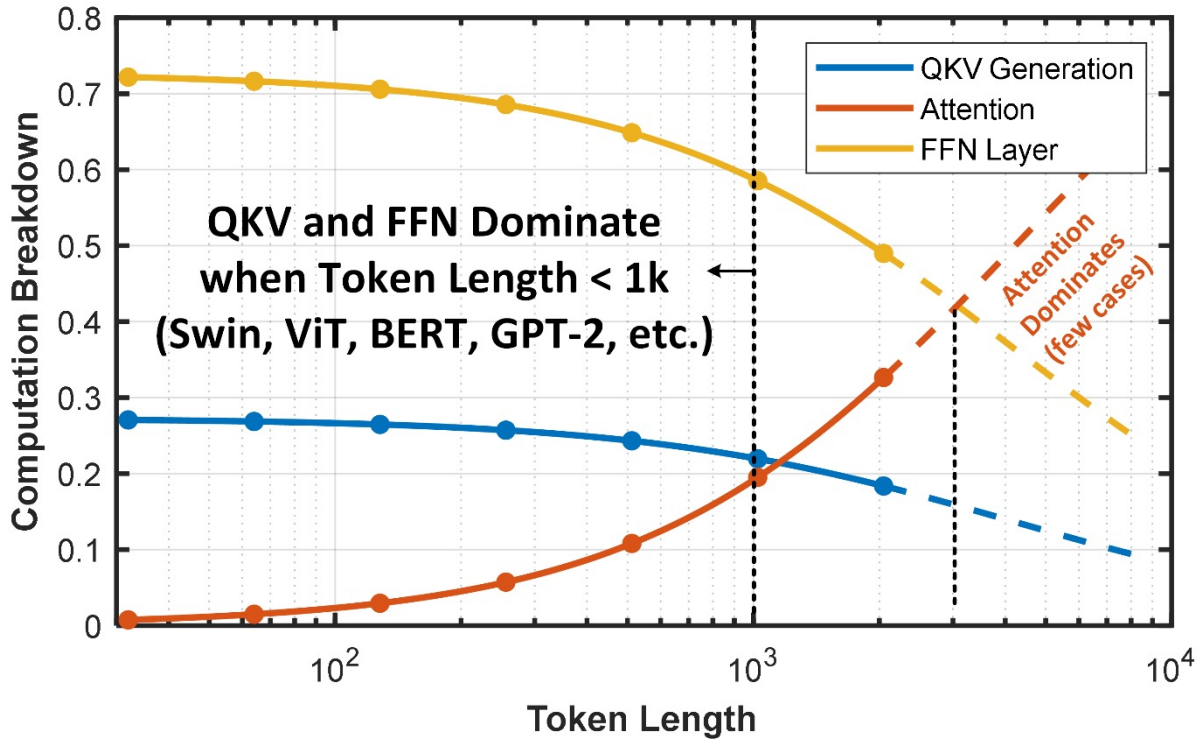




Power Breakdown of Transformer

- **Attention dominates only for very long token length**
- **The optimization of FFN and QKV is a must for most cases**

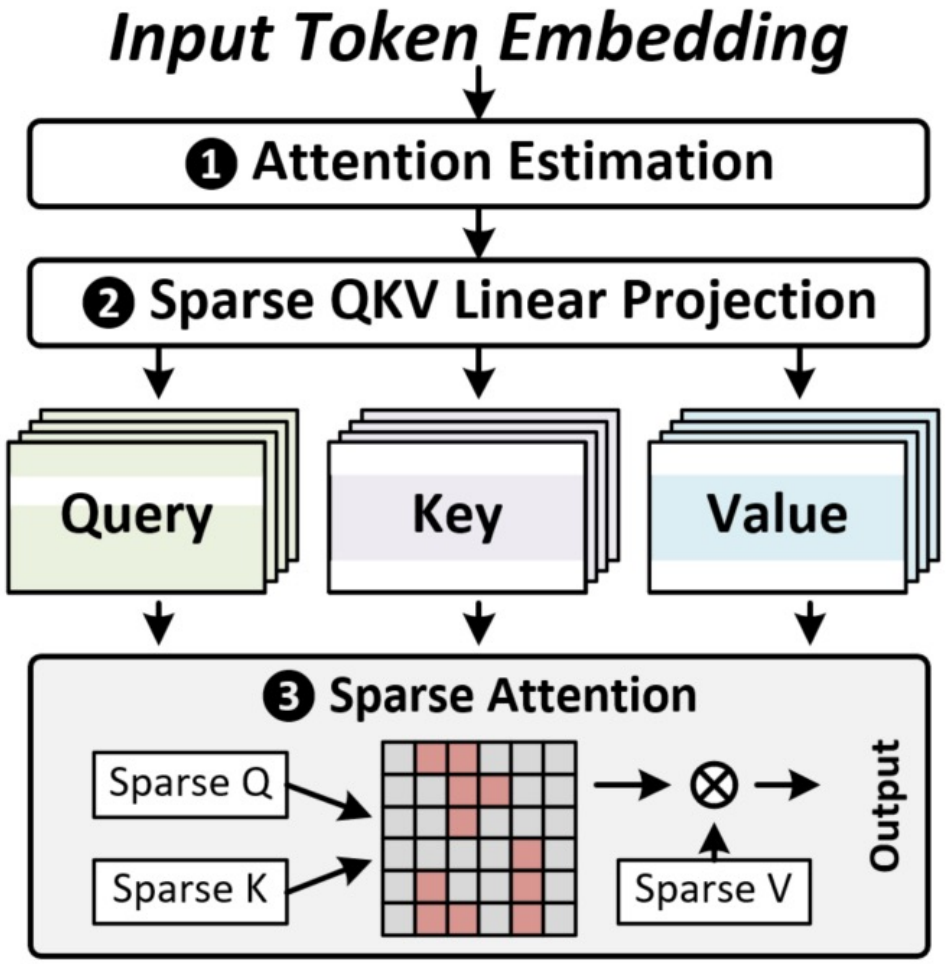
Layer Type	Num of Operation
Embedding	$4n_{token}d_{model}$
QKV Generation	$6n_{layer}n_{token}d_{model}^2$
Attention	$4n_{layer}n_{token}^2d_{model}$
Feed Forward	$18n_{layer}n_{token}d_{model}^2$
De-embedding	$2n_{token}d_{model}n_{vocab}$



Outline

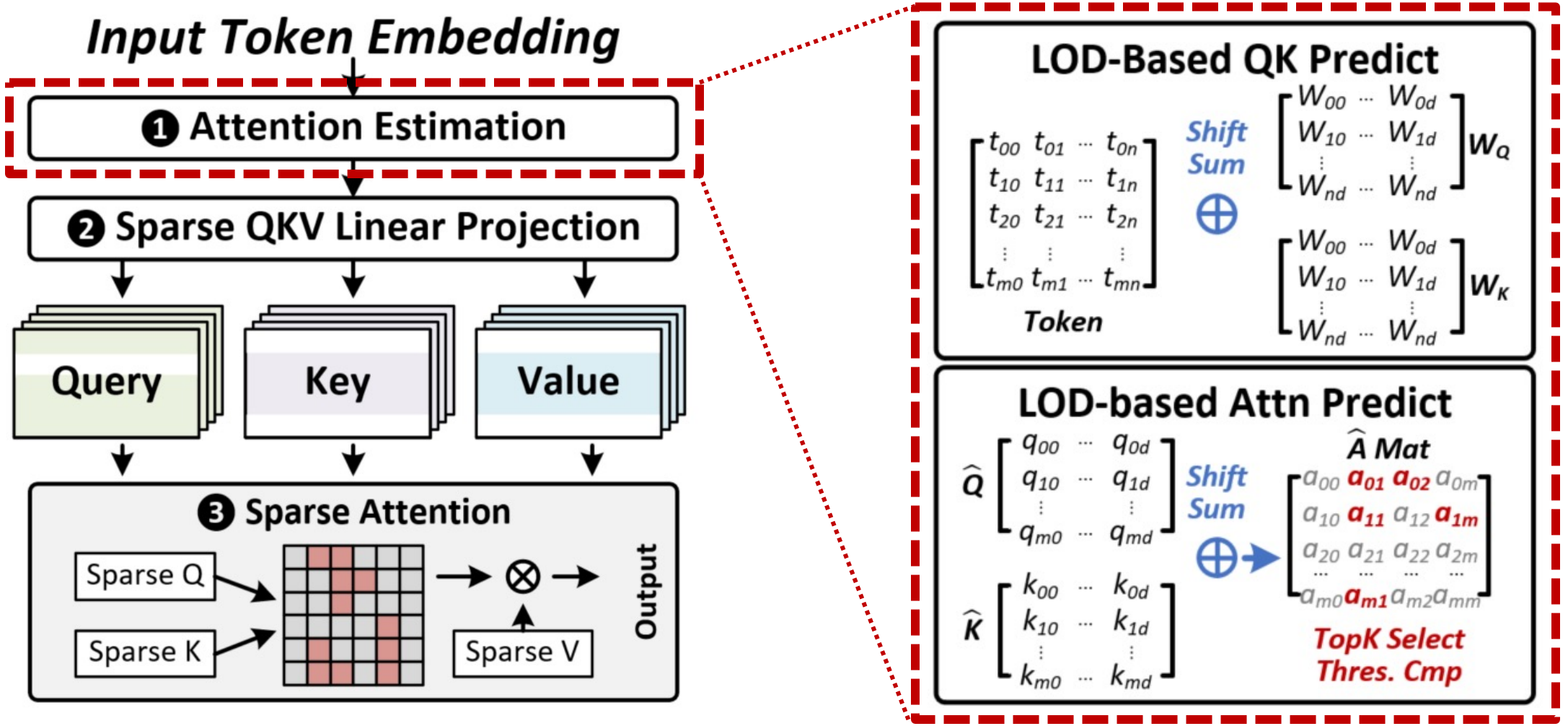
- Background and Motivation
- **Eager Prediction Mechanism**
- Hardware Innovation
- Evaluation

Eager Prediction – Procedure

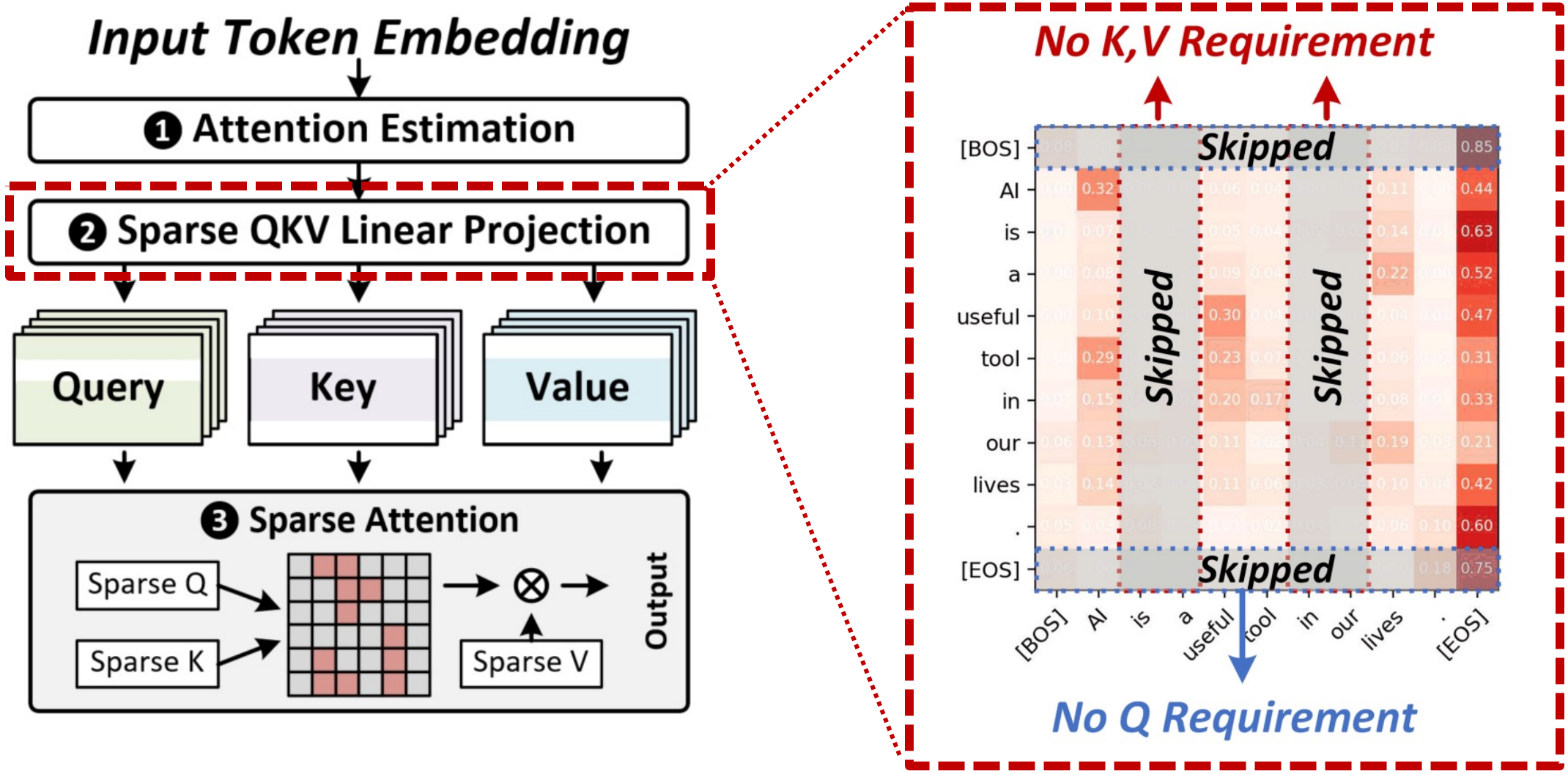


- **Predict the QKV result, and then the attention matrix**
- **Compute the sparsity mask of attention**
- **Deduce the sparsity of QKV**

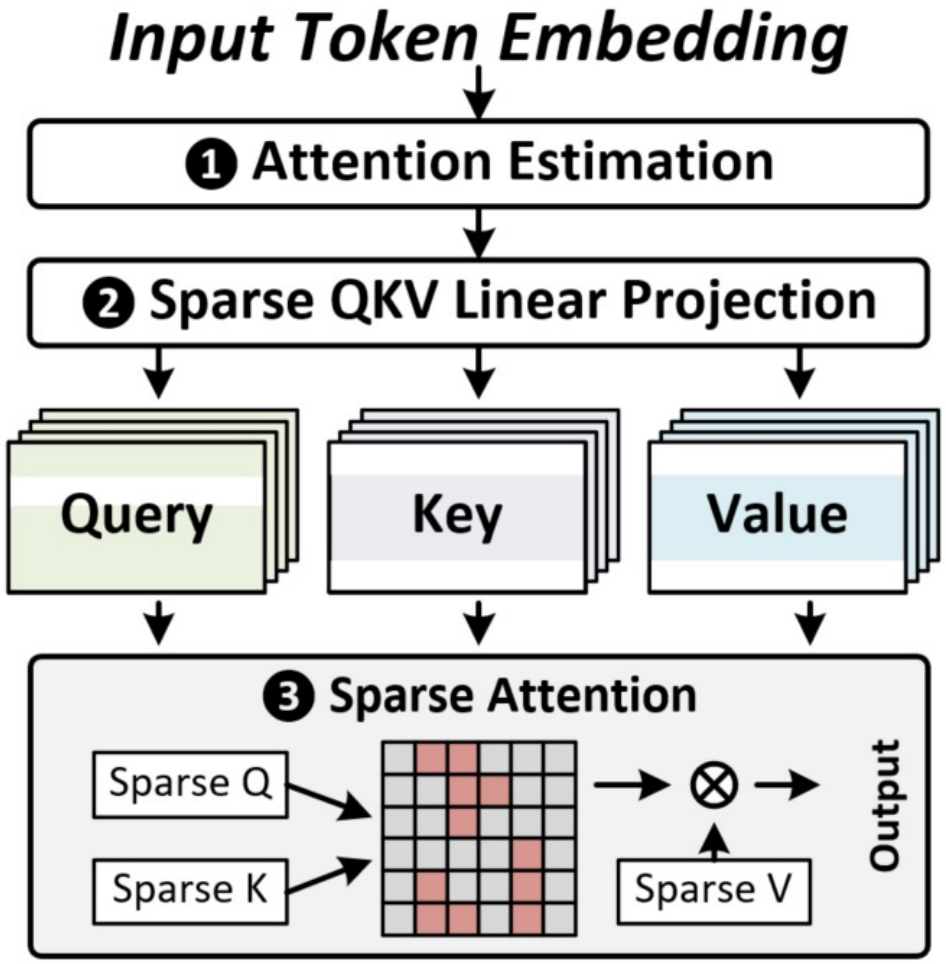
Eager Prediction – Procedure



Eager Prediction – Procedure



Eager Prediction – Algorithm



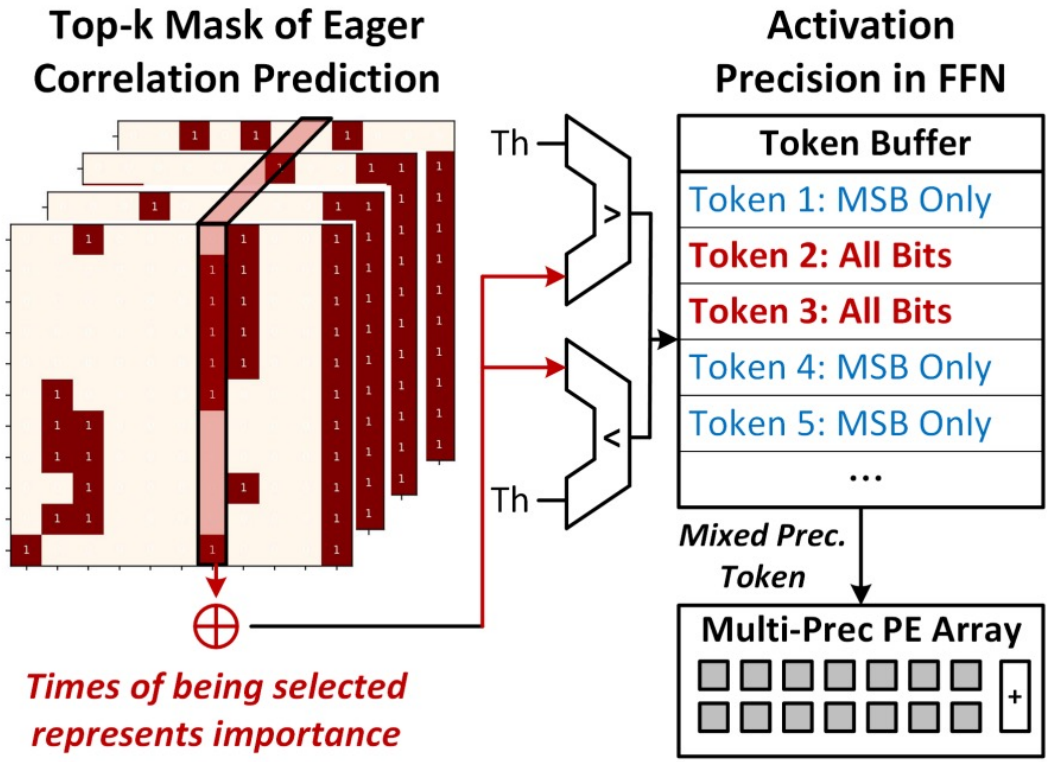
Algorithm Eager Prediction with Correlation Estimation

Input: $Token(T), Weight(W_Q, W_K, W_V)$
Input: k, r ▷ Top-K ratio and FFN threshold
Output: M^A, M^Q, M^{KV} ▷ Sparsity mask
Output: M^{FFN} ▷ Low precision mask for FFN

1: $E^T \leftarrow LOD(T)$
2: $E^{WQ} \leftarrow LOD(W_Q)$
3: $E^{WK} \leftarrow LOD(W_K)$
4: $\hat{Q_{i,j}} \leftarrow \sum_k Sgn(T_{i,k}W^Q_{k,j}) << (E^T_{i,k} + E^{WQ}_{k,j})$
5: $\hat{K_{i,j}} \leftarrow \sum_k Sgn(T_{i,k}W^K_{k,j}) << (E^T_{i,k} + E^{WK}_{k,j})$
6: $\hat{A_{i,j}} \leftarrow \sum_k Sgn(\hat{Q_{i,k}}\hat{K^T_{k,j}}) << (\hat{Q_{i,k}} + \hat{K^T_{k,j}})$
7: $M^A \leftarrow \hat{A_i} \geq top\_k(\hat{A_i}, k)$
8: $M^Q \leftarrow max(\hat{A_i}) - 2^{nd}\_max(\hat{A_i}) \leq Thr$
9: $M^{KV} \leftarrow \prod_k M^{EA}_{k,i}$
10: $M^{FFN} \leftarrow \sum M^{EA} \leq L \times C \times k \times r$

Eager Prediction – FFN Acceleration

- Adaptively reduces FFN precision according to the token's appearance in the top-k result

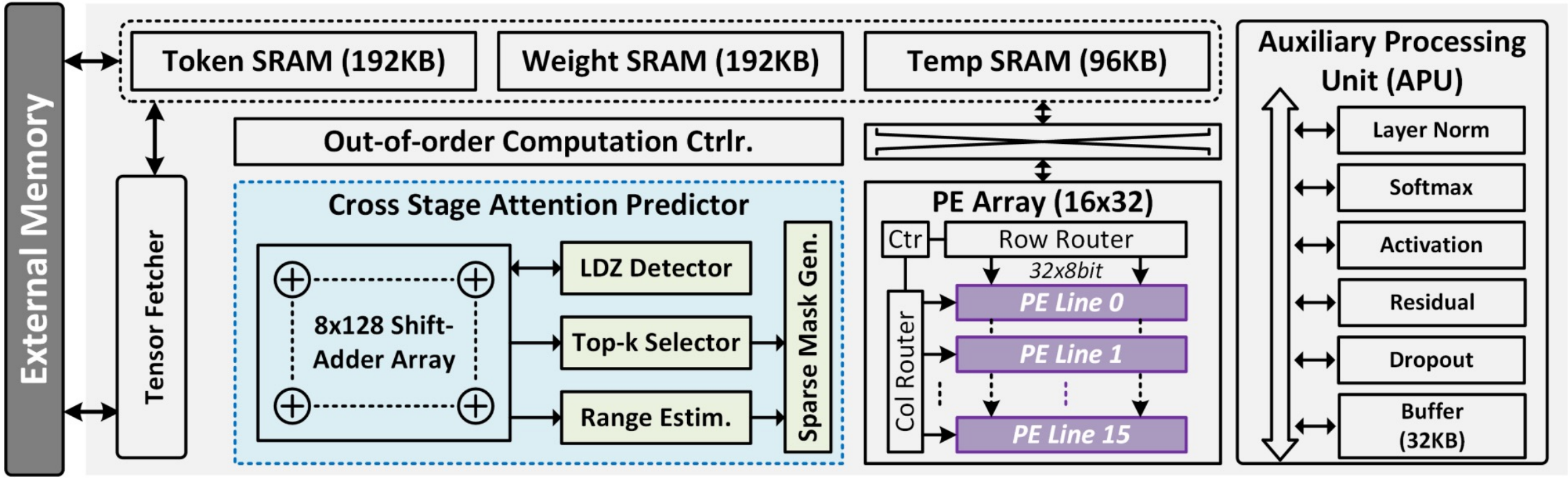


- A token's key's being selected by top-k means the token's content is crucial to the query.
- The more times being selected by top-k, the more important the token is.
- EP only assigns high FFN precision to important tokens.

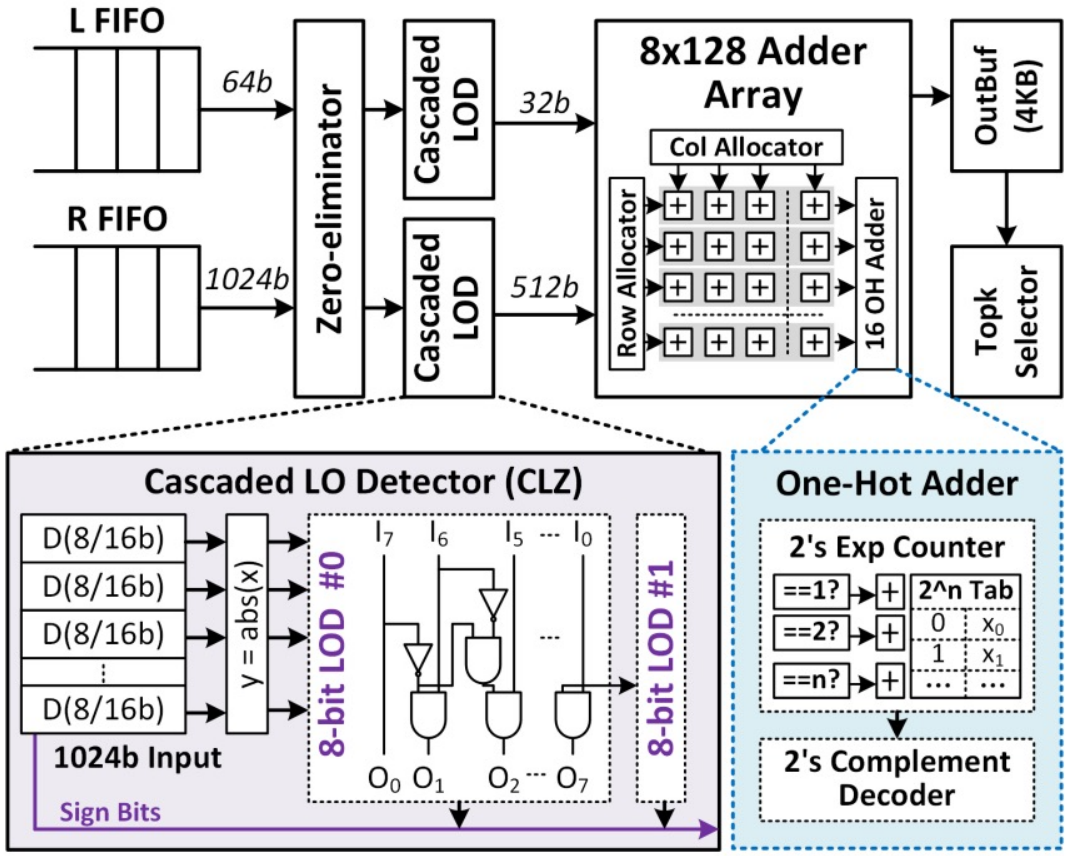
Outline

- Background and Motivation
- Eager Prediction Mechanism
- **Hardware Innovation**
- Evaluation

Overall Architecture

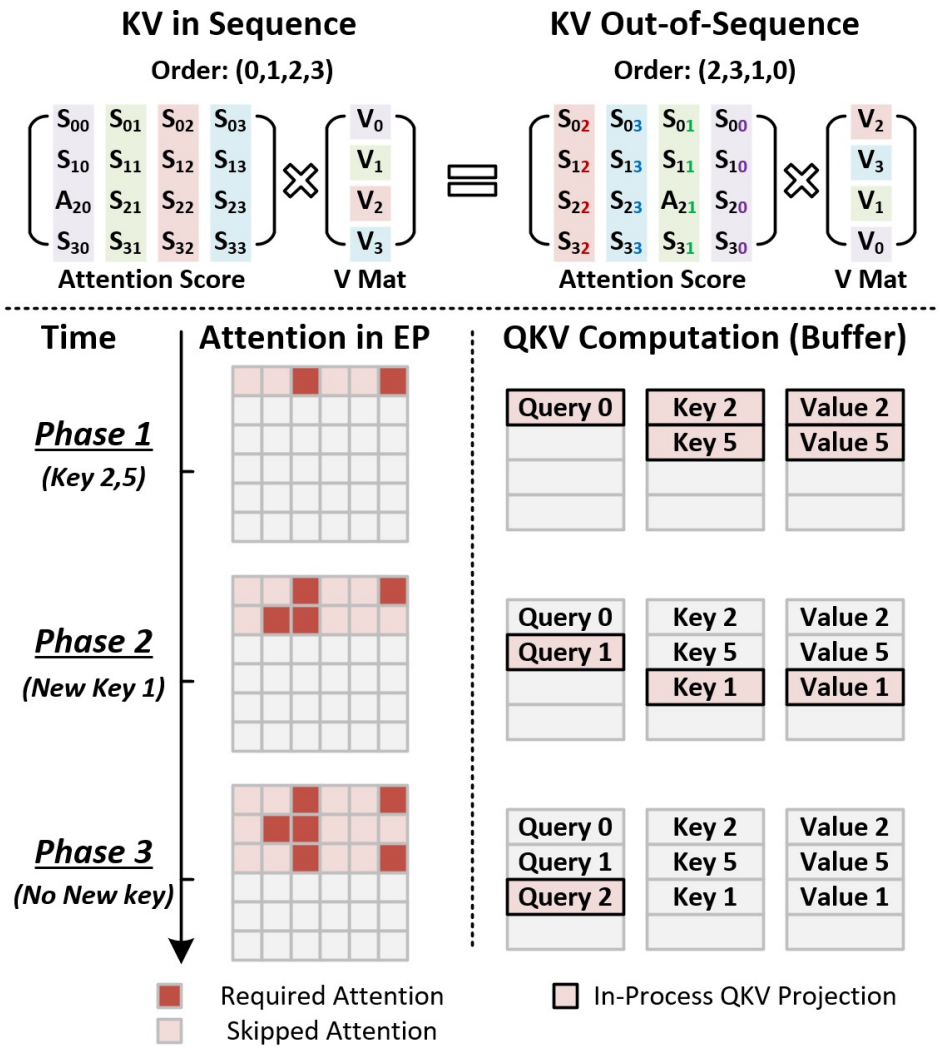


Multiplication-free EP Unit



- Using leading-one detector (LOD) for MM prediction
- FIFO stores MM input matrix, with sparse data eliminated
- Cascaded LOD can be applied to INT16/INT8 input datatype
- One-hot adder tree adds up the position of LOD result

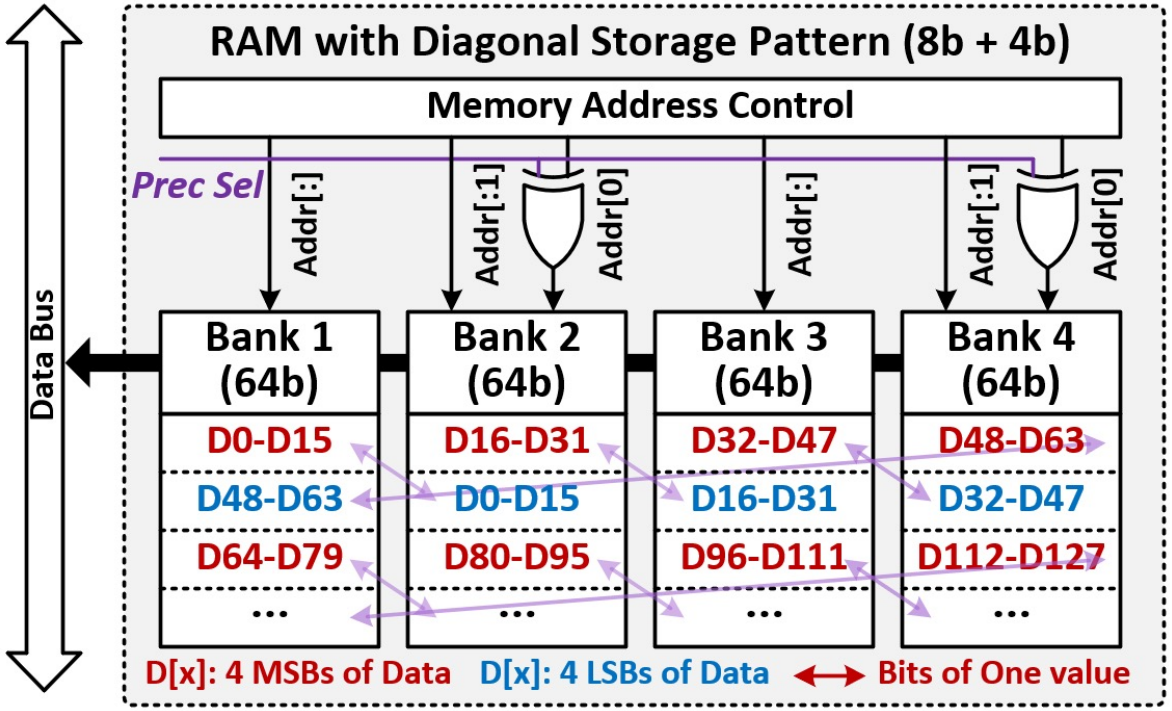
Out-of-order QKV Scheduler

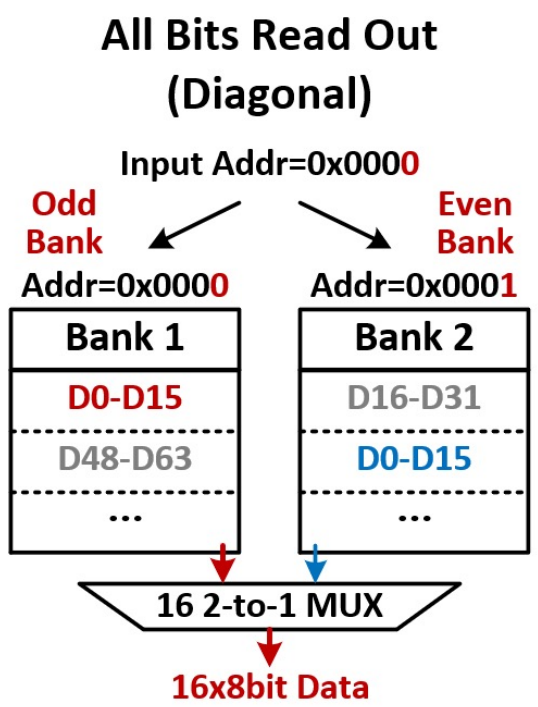


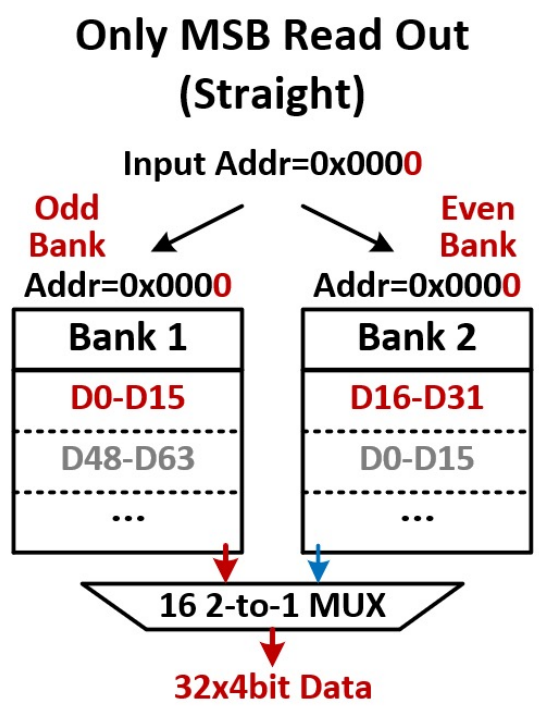
- Reorder the Key and Value matrix to minimize the memory footprint
- Only load additional required KV for new query, unused ones skipped
- Reorder the order of query to maximize data reuse of KV
- If a new query needs the same KV as the last one, it has high priority

Diagonal Storage Pattern

- **Store MSB parts and LSB parts with periodic shift**
  Ensure readout for MSB-only and full data with no waste







Outline

- Background and Motivation
- Eager Prediction Mechanism
- Hardware Innovation
- **Evaluation**

Experimental Setup

Software:

- Model build and finetune with PyTorch
  - BERT-base, BERT-large for GLUE, SQuAD and Wiki-2
  - ViT/B-32, ViT/B-16 for ImageNet-1k classification

Hardware:

- RTL design with Synopsys Design Compiler with 28nm 1P8M CMOS
- Placement and routing with DCG and ICC
- Post-layout simulation with VCS and PT

Processor Layout and Specification

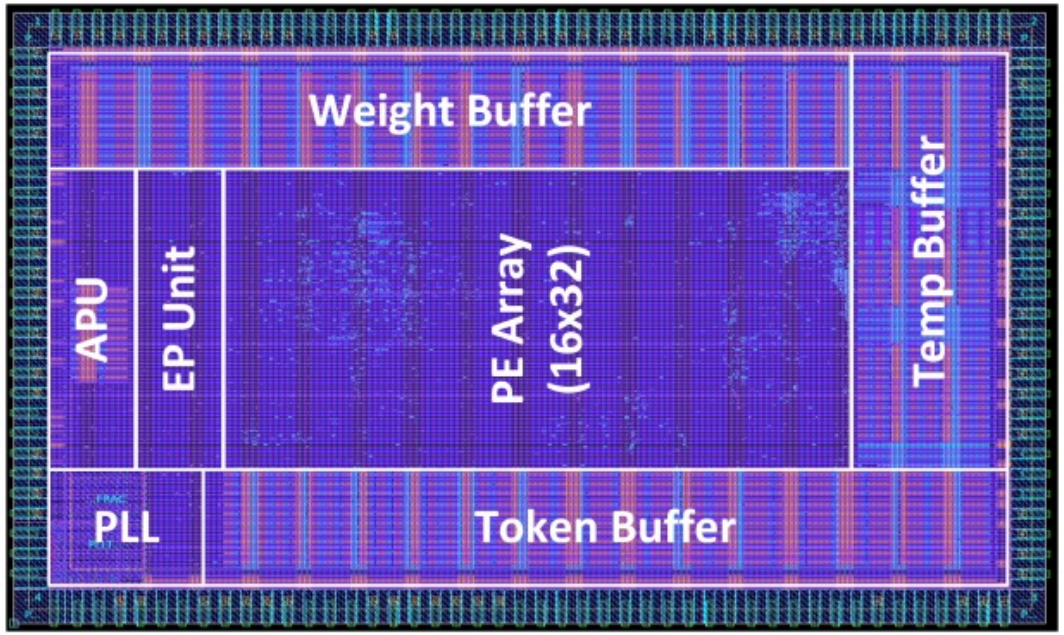
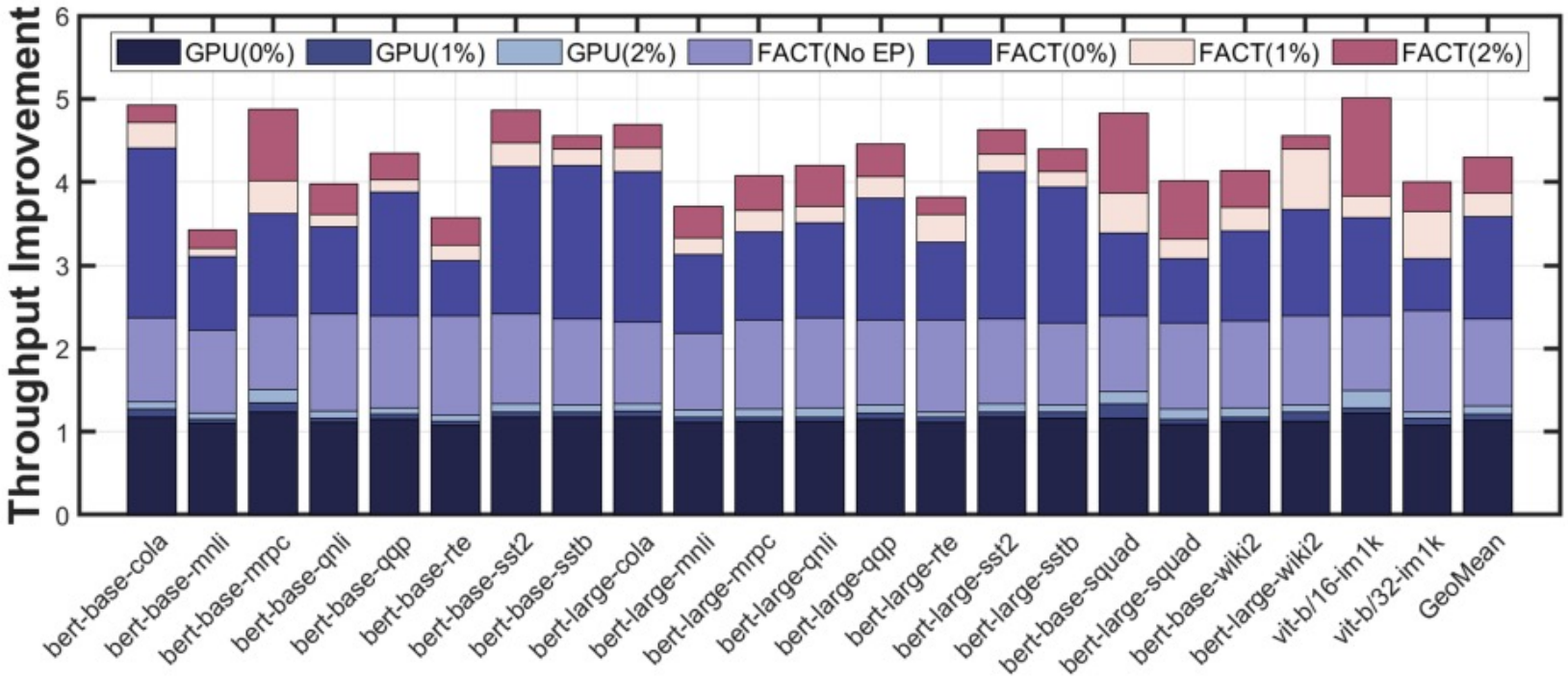


Figure: Chip layout after placement

Table: Area and Power breakdown @500MHz

Module	Area [mm^2]	Power [mW]
PE Array	2.76	162.93
EP Unit - Adder	0.23	16.04
EP Unit - Others	0.02	1.59
APU	0.56	23.95
SRAM	2.23	110.28
Others	0.22	22.28
Total	**6.03**	**337.07**

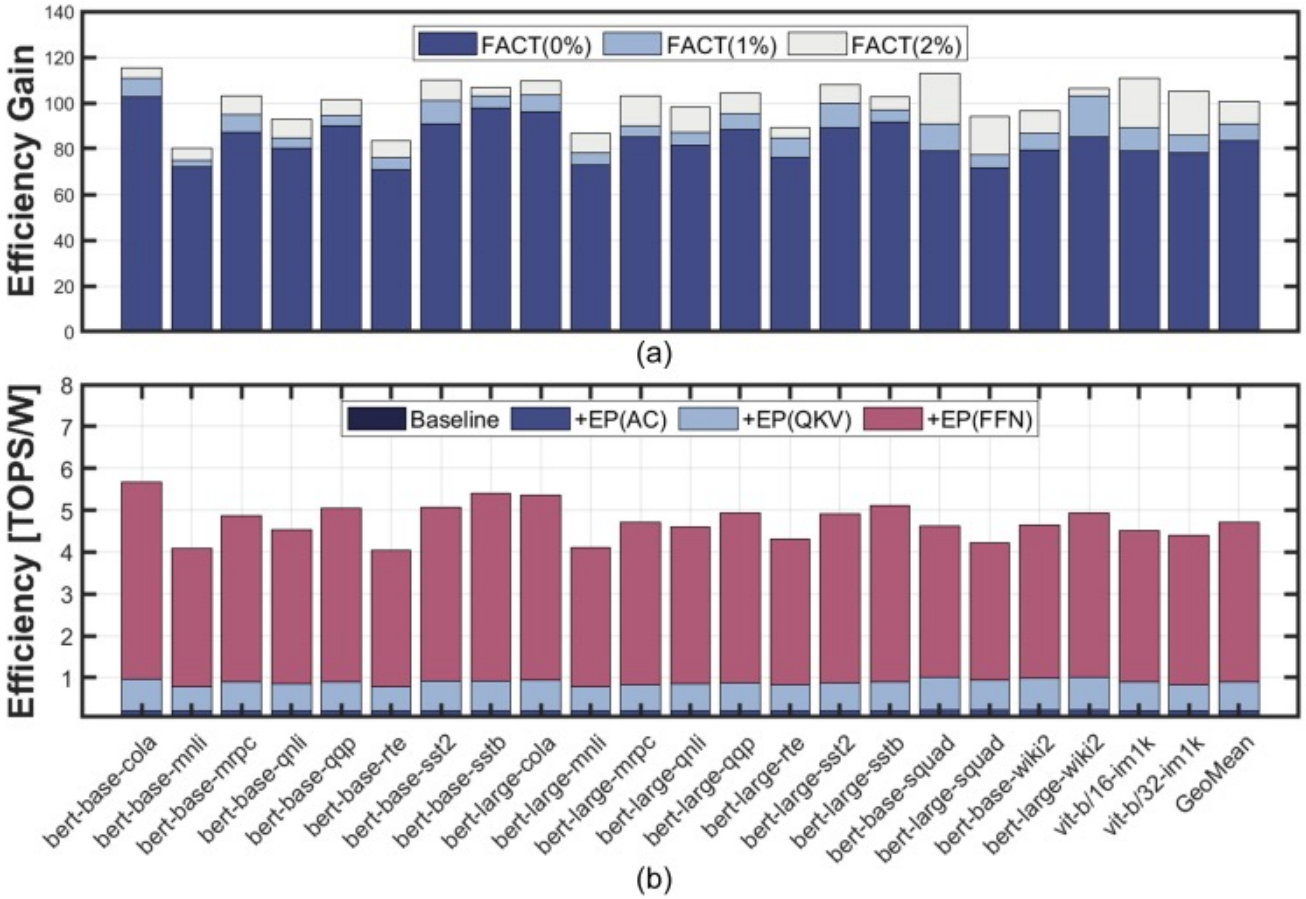
Evaluation – Throughput and Efficiency



FACT accelerates the whole model inference process by 3.59x/3.87x/4.30x on geomean average of 22 tasks, within 0%/1%/2% accuracy degradation

Specifications	
Acceleration for	QKV, Attention, FFN
Frequency [MHz]	500
Area [mm^2]	6.03
Throughput [GOPS]	928
Area Efficiency [GOPS/mm2]	153.9
Energy Efficiency [GOPS/W]	4388

Evaluation – Throughput and Efficiency



- **83.4x higher energy efficiency compared to V100 GPU**
- **Optimizing the QKV and FFN layers yields 22.9x efficiency gain**

Conclusion

- **An energy-efficient Transformer inference design with:**
  - Eager correlation prediction algorithm
    - √ Reduces computation of all the main blocks of Transformer
  - Multiplication-free eager prediction unit
    - √ Ensure neglectable sparsity prediction overhead
  - Out-of-order QKV Scheduler
    - √ Maximizes data reuse for sparse attention computation
  - Diagonal data storage pattern
    - √ Avoids unnecessary memory footprint for mix-precision FFN

Thank you

Any Questions?