EdgeBERT: Sentence-Level Energy Optimizations for Latency-Aware Multi-Task NLP Inference

Thierry Tambe[1], Coleman Hooper[1], Lillian Pentecost[1], Tianyu Jia[1], En-Yu Yang[1], Marco Donato[2], Victor Sanh[3], Paul Whatmough[4,1], Alexander M. Rush[5,3], David Brooks[1], Gu-Yeon Wei[1]

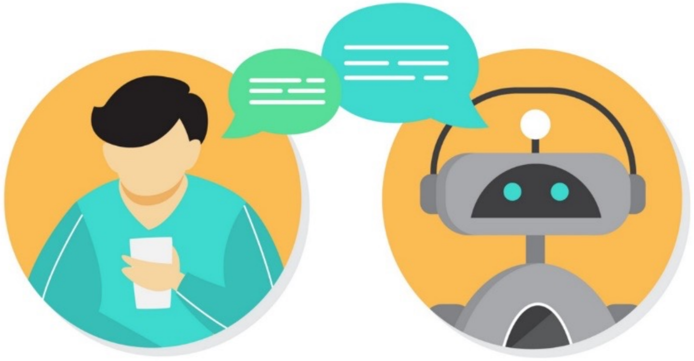
[1]Harvard University, [2]Tufts University, [3]Hugging Face, [4]Arm Research, [5]Cornell University

54th IEEE/ACM International Symposium on Microarchitecture (MICRO 2021)

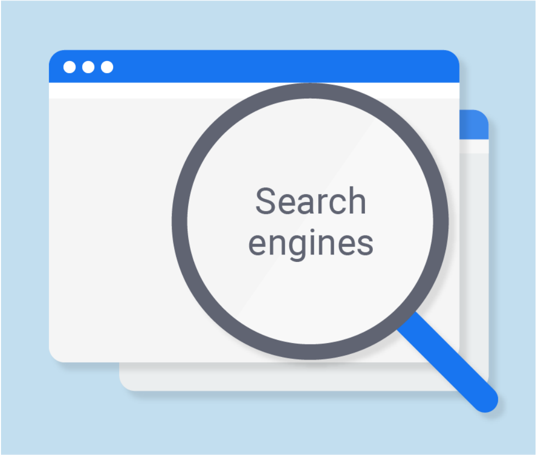
Harvard John A. Paulson
School of Engineering
and Applied Sciences

Deep learning based NLP is applied widely

Language Modeling & Understanding



Chat Bots

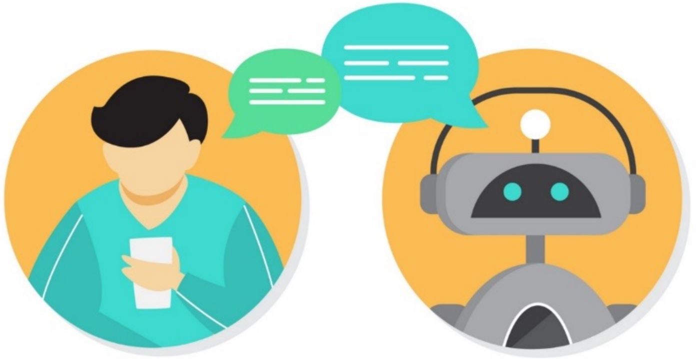


Search Engines

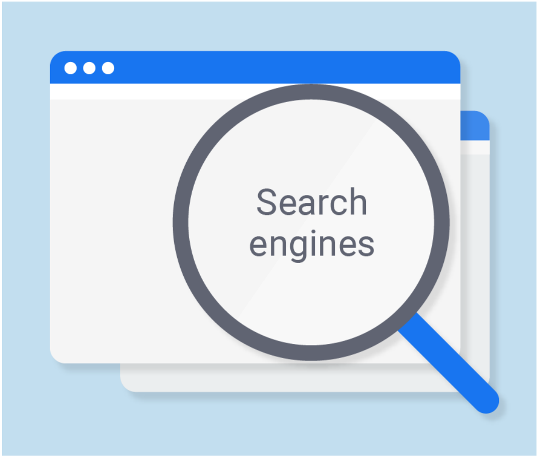
Image references
https://sitechecker.pro/search-engines/
https://towardsdatascience.com/automatic-question-answering-ac7593432842
https://usersnap.com/blog/design-language-switch/

Deep learning based NLP is applied widely

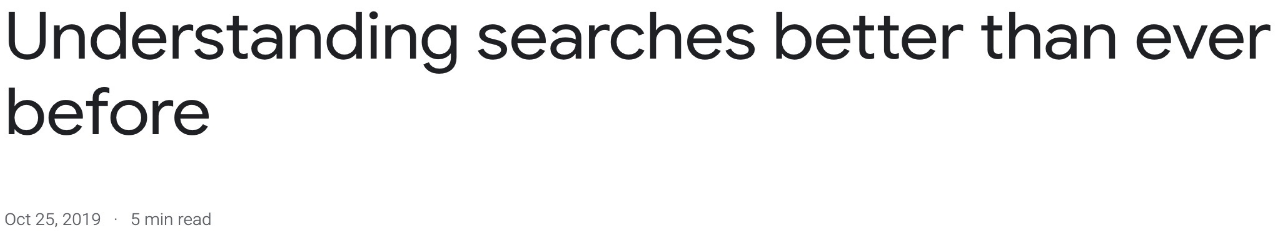
Language Modeling & Understanding



Chat Bots



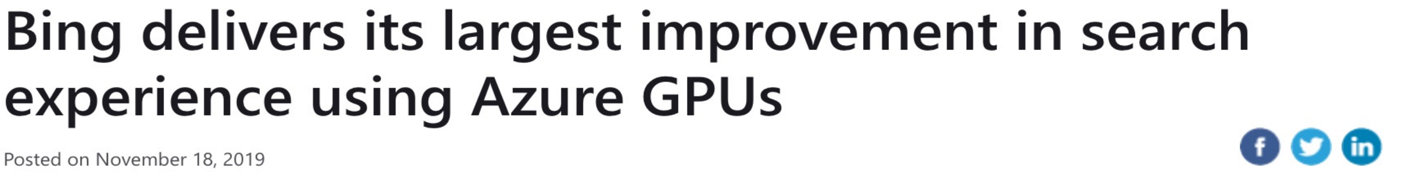
Search Engines



Understanding searches better than ever before

Oct 25, 2019 · 5 min read

https://blog.google/products/search/search-language-understanding-bert/



Bing delivers its largest improvement in search experience using Azure GPUs

Posted on November 18, 2019

https://azure.microsoft.com/en-us/blog/bing-delivers-its-largest-improvement-in-search-experience-using-azure-gpus/

Image references
https://sitechecker.pro/search-engines/
https://towardsdatascience.com/automatic-question-answering-ac7593432842
https://usersnap.com/blog/design-language-switch/

Transformers are behind NLP success

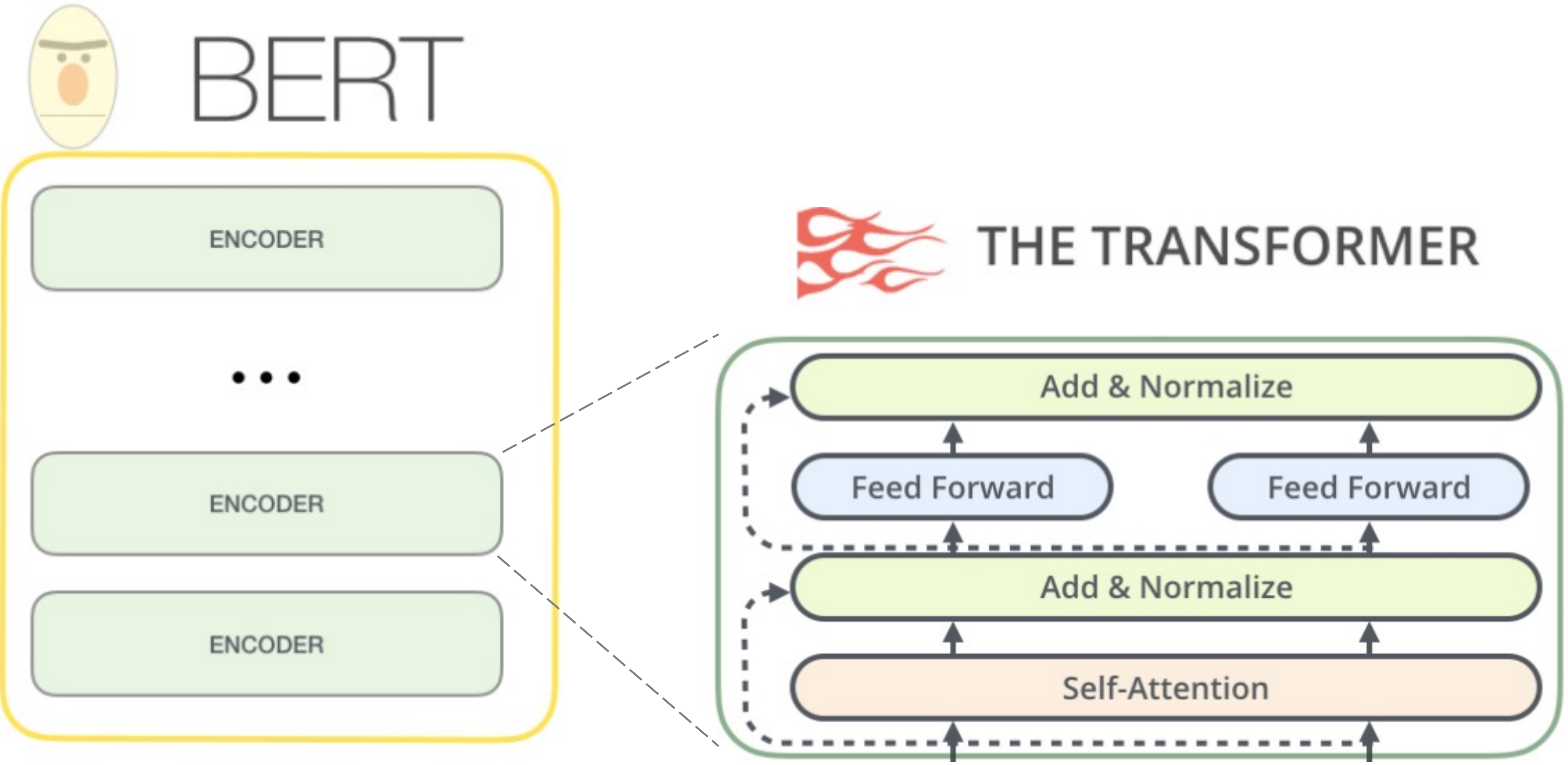
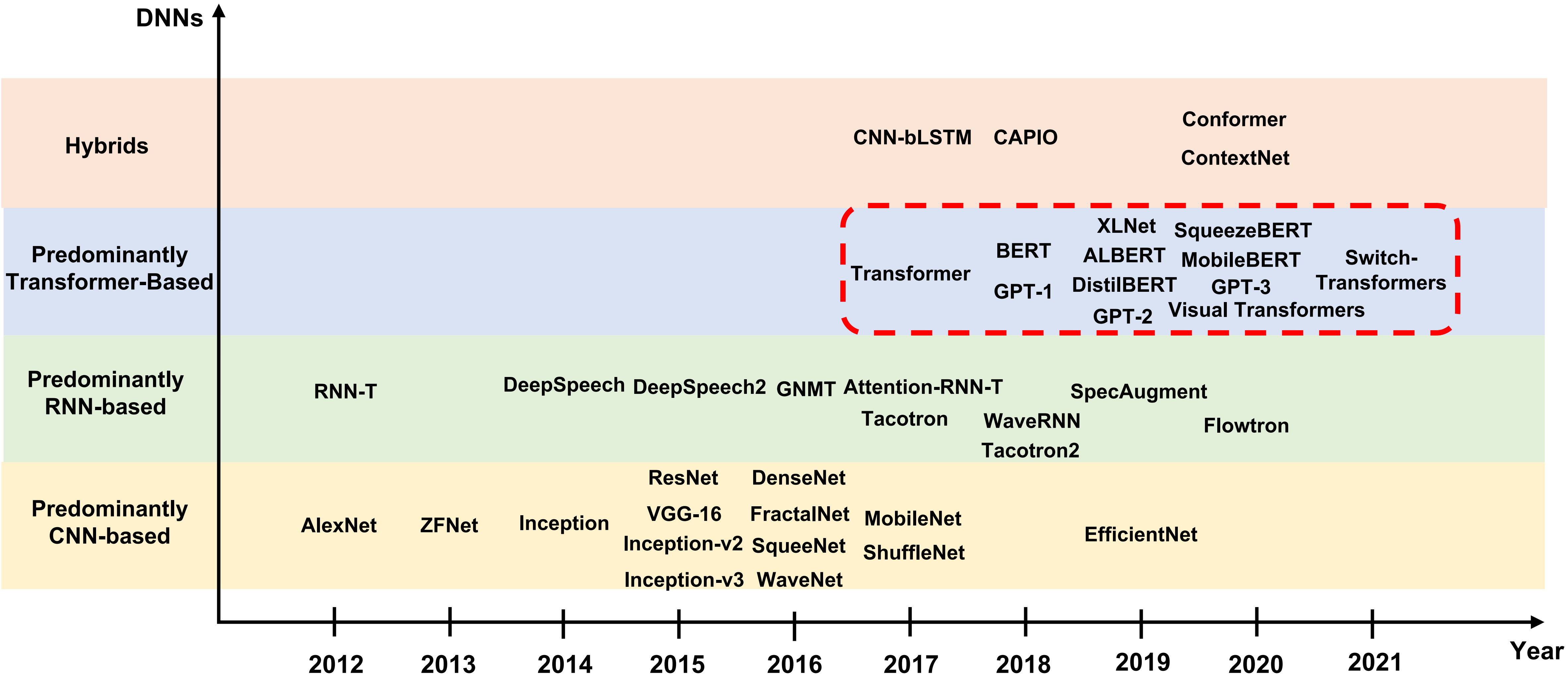
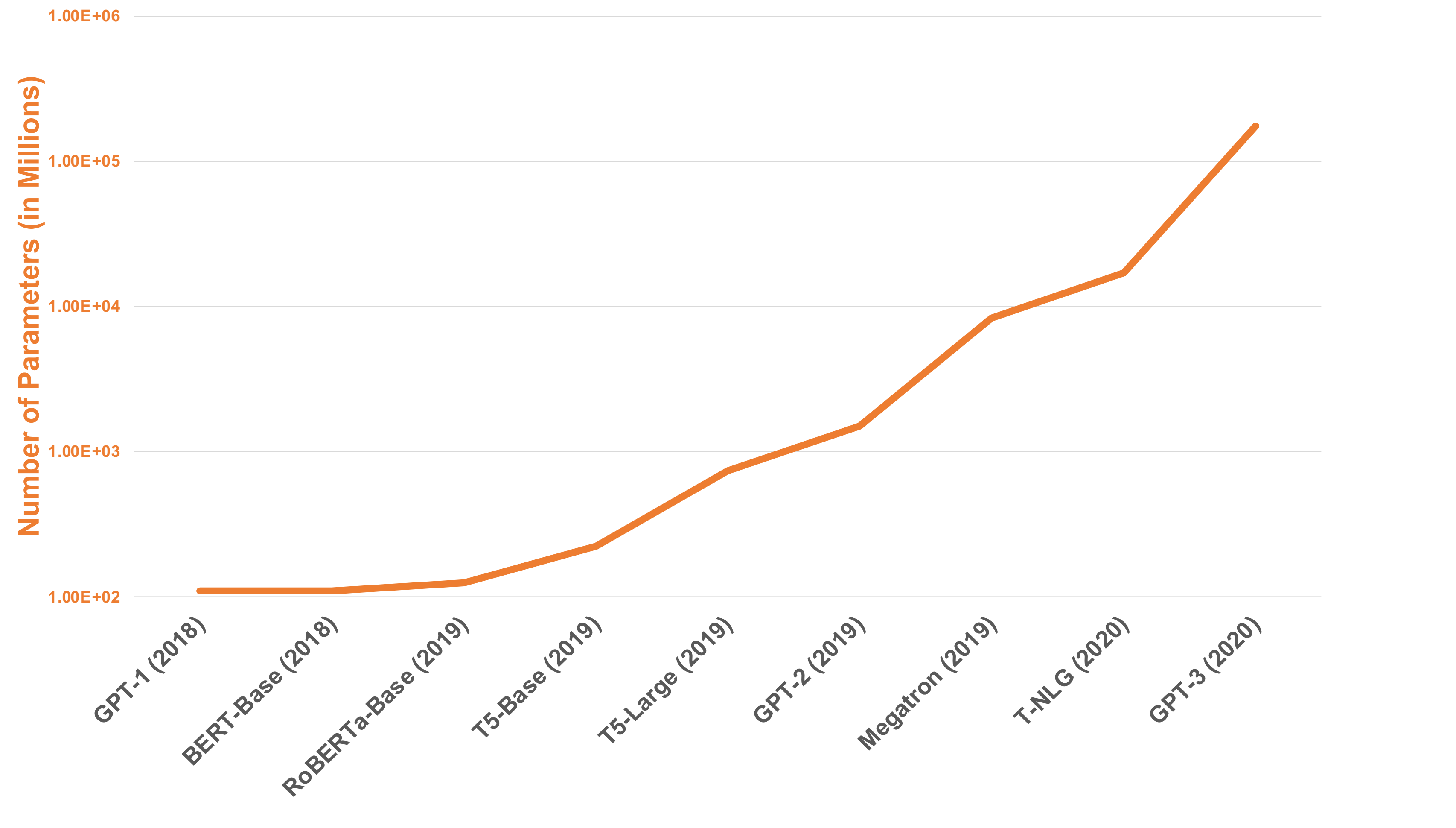


Image references
https://jalammar.github.io/illustrated-transformer/
https://jalammar.github.io/illustrated-gpt2/

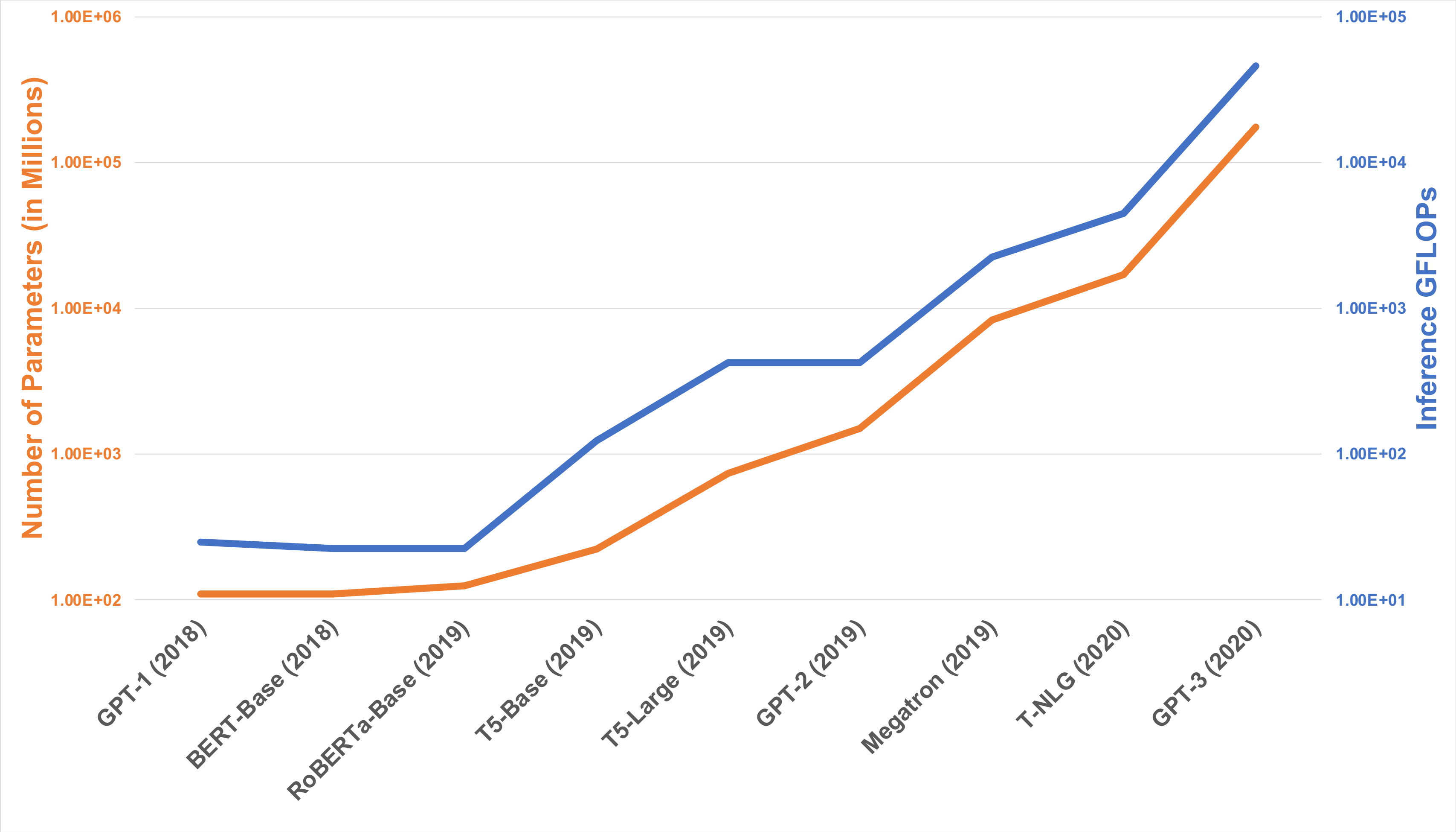
Newer AI models are increasingly Transformer-based



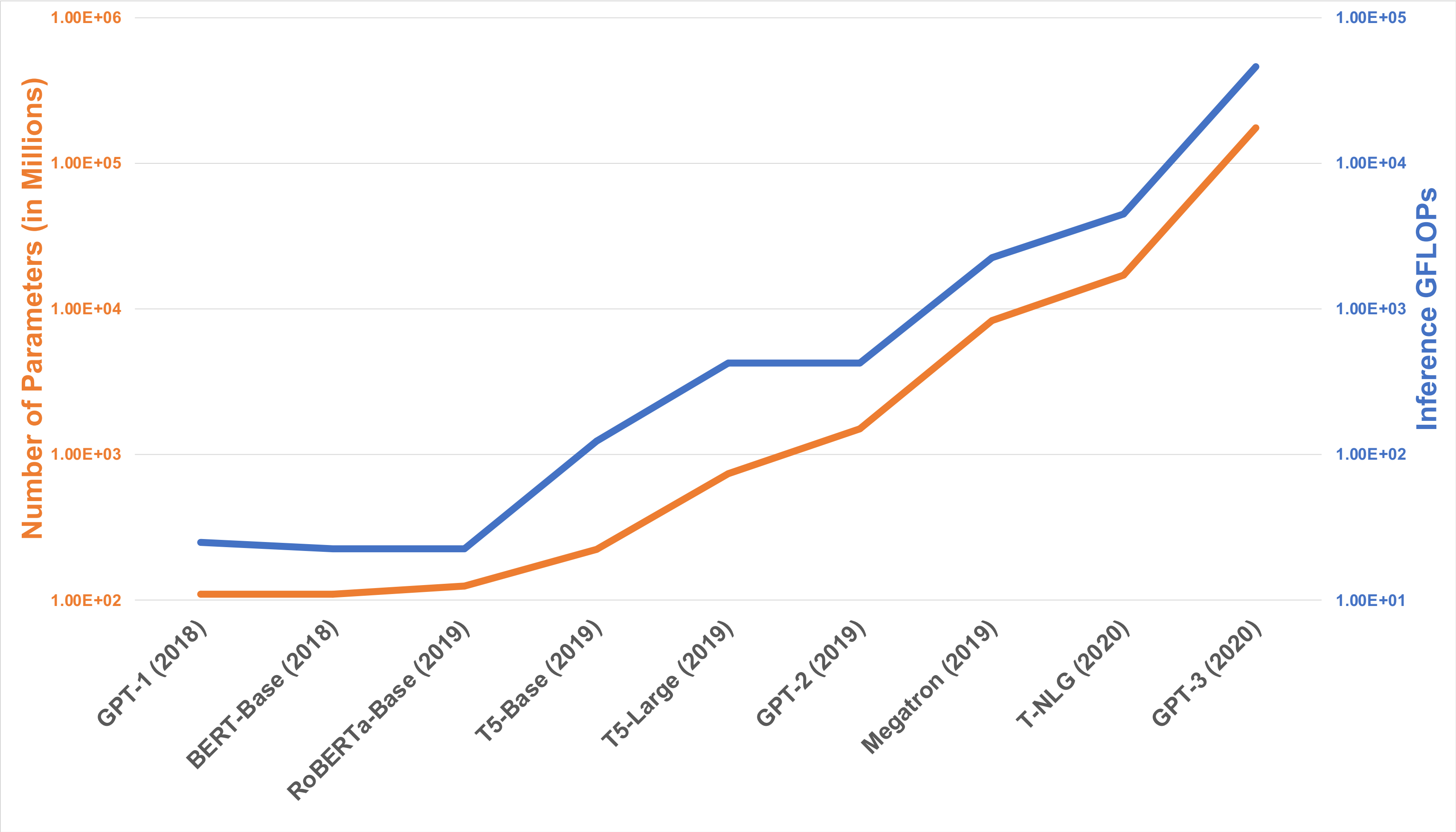
NLP growing overhead



NLP growing overhead



NLP growing overhead



- Need for a 2-dimensional compression along the memory and computations axes for energy-efficient inference
- End of Moore’s law requires software-hardware specialization

What is EdgeBERT?

EdgeBERT is a cross-stack (algorithm, architecture, solid-state) set of optimizations for minimizing the energy consumption of multi-task NLP inference at a sentence granularity under the constraint of an application end-to-end latency target.

Abstracting Energy Consumption

$$Energy \propto \alpha\, C\, V_{DD}^2\, N_{cycles}$$

- α – switching activity factor
- C – wire and device capacitance
- V_{DD}^2 – supply voltage
- N_{cycles} – # of inference clock cycles

All-Encompassing Energy Reduction

$$Energy \propto \alpha\ C\ \overset{\substack{\text{Latency-Aware}\\ \text{DVFS}\\ \uparrow}}{\boxed{V_{DD}^2}}\ N_{cycles}$$

- **Latency-aware dynamic voltage frequency scaling enforces a quadratic reduction in the accelerator energy consumption**

All-Encompassing Energy Reduction

Latency-Aware
DVFS

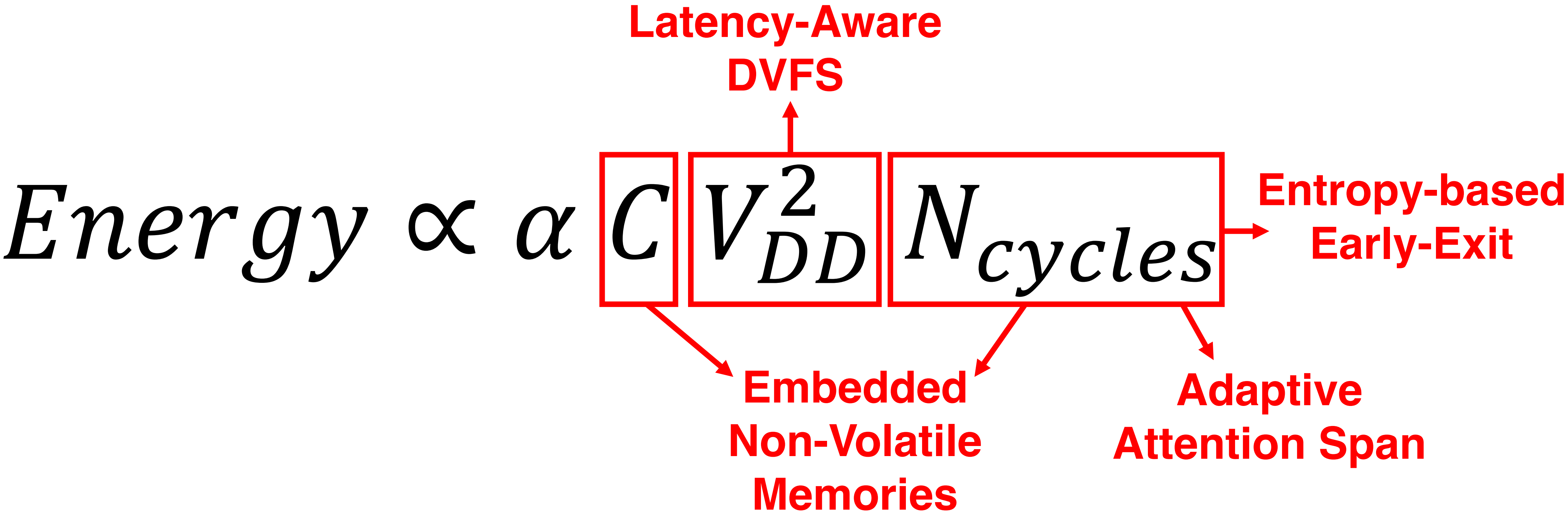
$$Energy \propto \alpha\ C\ \boxed{V_{DD}^{2}}\ \boxed{N_{cycles}}$$

Entropy-based
Early-Exit

Adaptive
Attention Span

➢ **Entropy-based early exit and adaptive attention span reduce the required number of FLOPs**

All-Encompassing Energy Reduction



$$Energy \propto \alpha \boxed{C} \boxed{V_{DD}^2} \boxed{N_{cycles}}$$

Latency-Aware DVFS

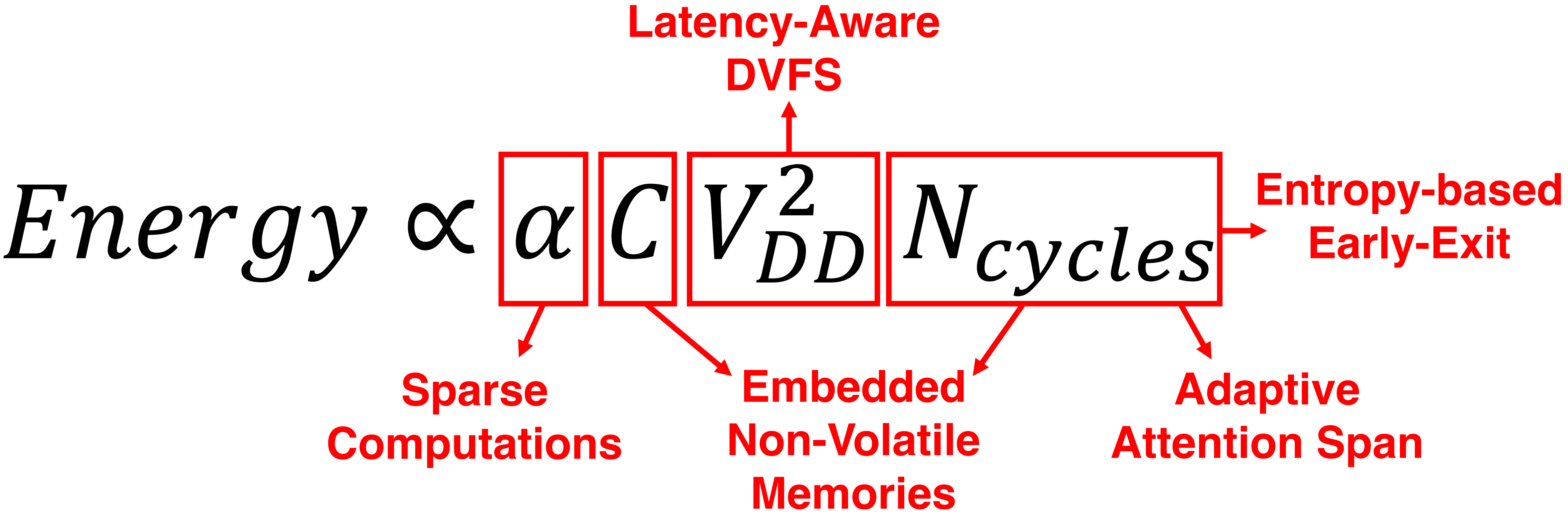
Entropy-based Early-Exit

Embedded Non-Volatile Memories

Adaptive Attention Span

- ➢ **eNVMs for NLP word embedding storage ultimately reduce on-chip capacitance and memory read cycles**

All-Encompassing Energy Reduction



$$Energy \propto \boxed{\alpha}\,\boxed{C}\,\boxed{V_{DD}^2}\,\boxed{N_{cycles}}$$

Latency-Aware DVFS

Entropy-based Early-Exit

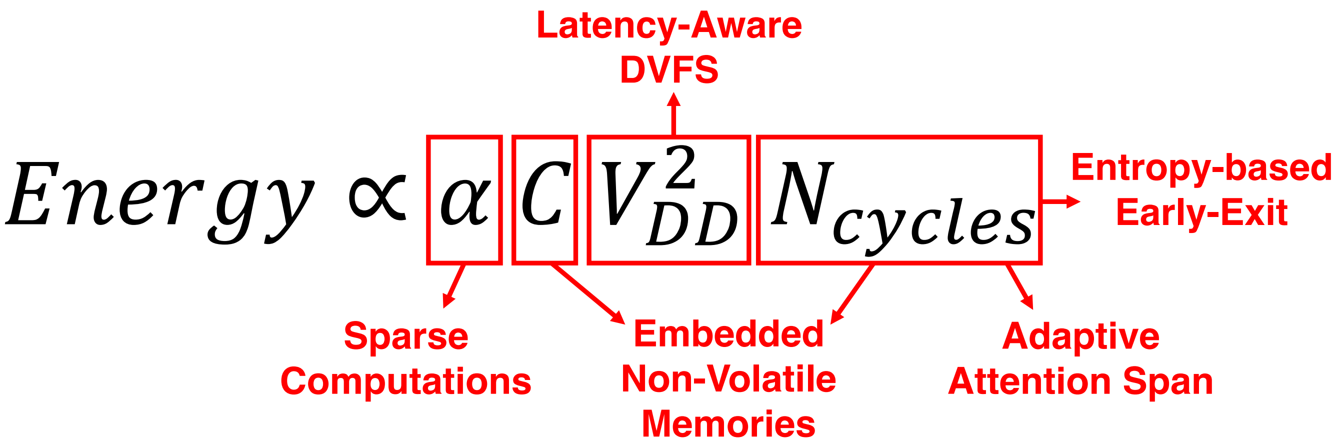
Sparse Computations

Embedded Non-Volatile Memories

Adaptive Attention Span

- ➢ **Sparse computations in the EdgeBERT HW accelerator considerably lowers energy consumption via MAC gating and logic skipping**

Energy Savings Contributions in 12nm Accelerator Adaptation



$$Energy \propto \boxed{\alpha}\,\boxed{C}\,\boxed{V_{DD}^{2}}\,\boxed{N_{cycles}}$$

Latency-Aware DVFS

Entropy-based Early-Exit

Sparse Computations

Embedded Non-Volatile Memories

Adaptive Attention Span

Energy Savings Contributions in 12nm Accelerator Adaptation

Latency-Aware DVFS

$$Energy \propto \boxed{\alpha}\,\boxed{C}\,\boxed{V_{DD}^2}\,\boxed{N_{cycles}}$$

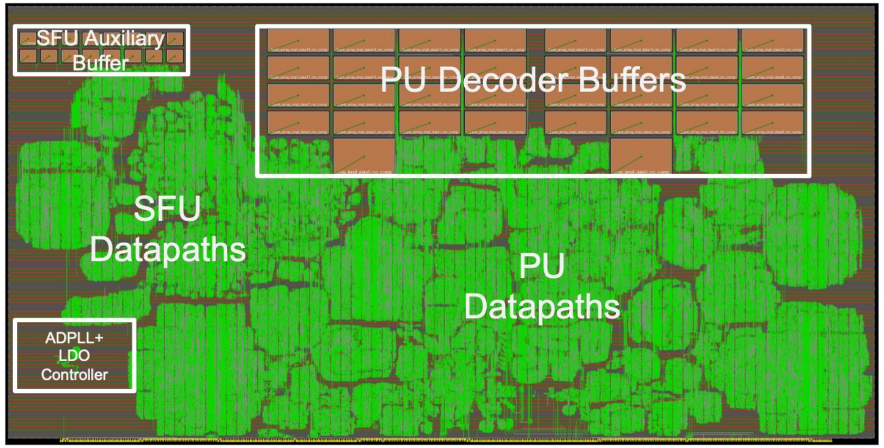
Entropy-based Early-Exit

Sparse Computations

Embedded Non-Volatile Memories

Adaptive Attention Span

12nm EdgeBERT accelerator



Energy Savings Contributions in 12nm Accelerator Adaptation

Latency-Aware DVFS

$$Energy \propto \boxed{\alpha}\,\boxed{C}\,\boxed{V_{DD}^{2}}\,\boxed{N_{cycles}}$$

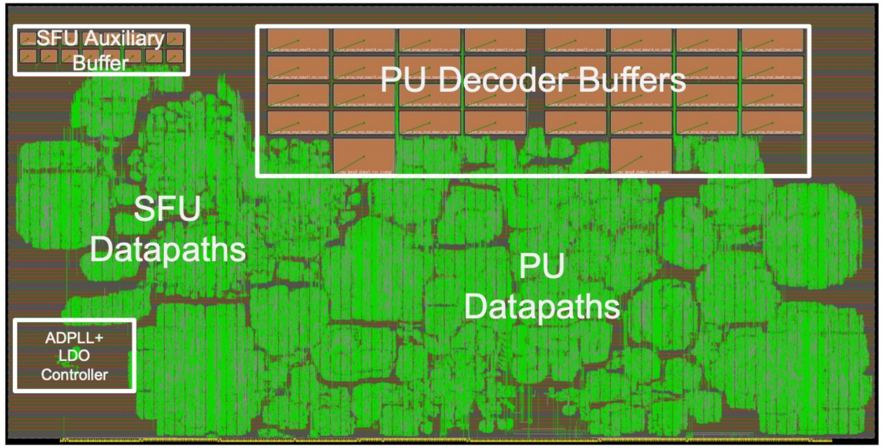
Entropy-based Early-Exit

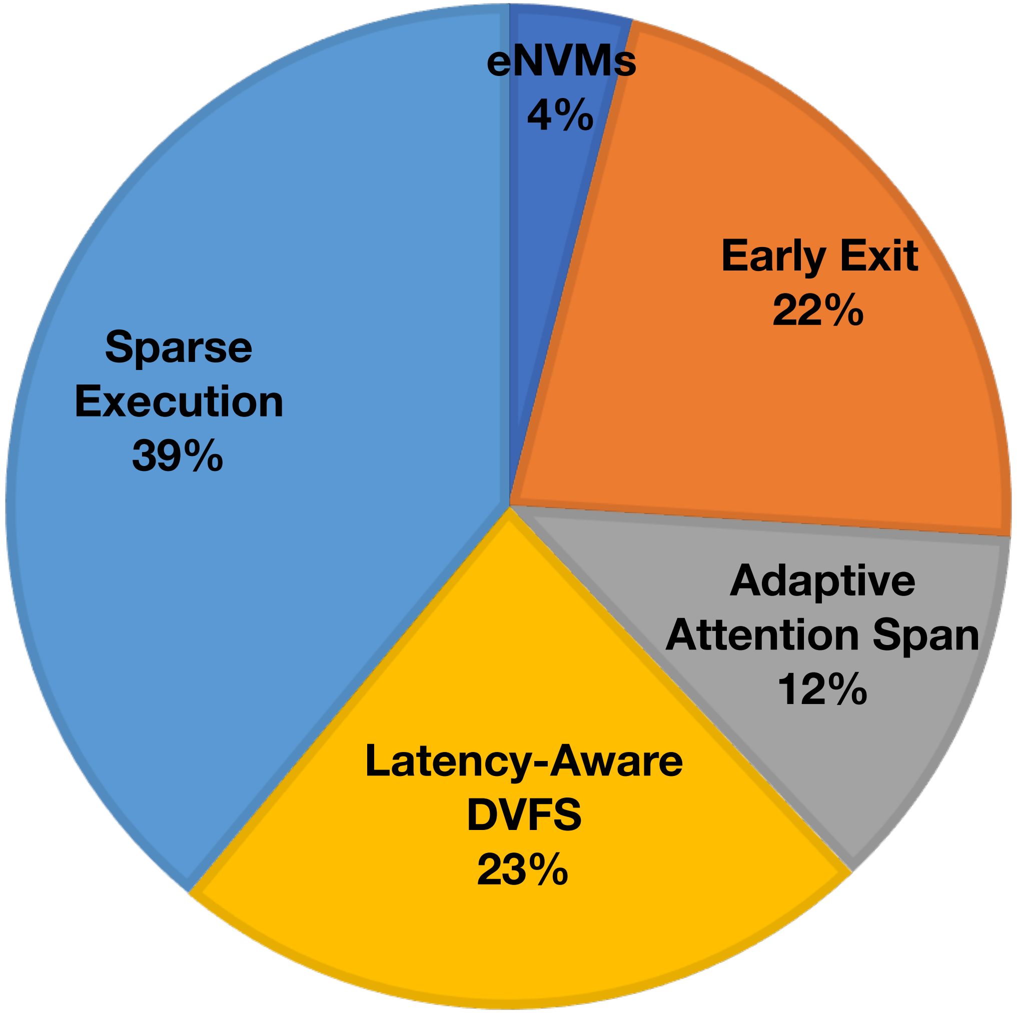
Sparse Computations

Embedded Non-Volatile Memories

Adaptive Attention Span

12nm EdgeBERT accelerator





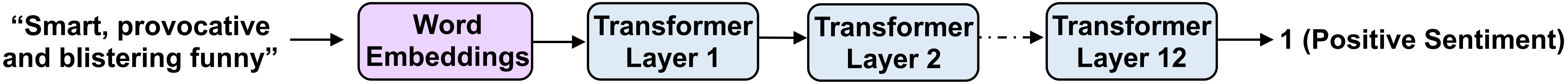
Outline

- Motivation
- EdgeBERT Optimizations
- Synergistic Evaluation
- Hardware Architecture
- Hardware Evaluation
- Conclusion

Outline

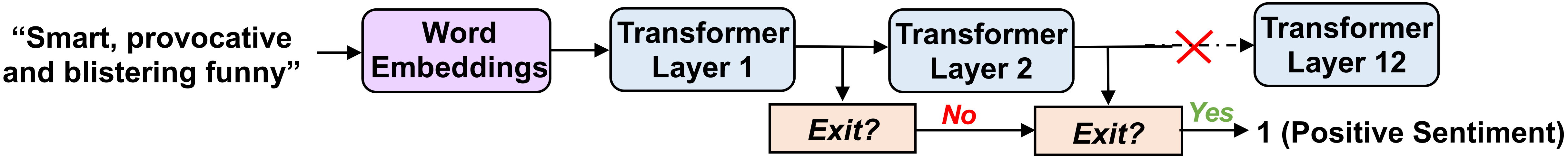
- Motivation
- EdgeBERT Optimizations
    - Entropy-based DVFS for Latency-Bounded NLP Inference
    - Adaptive Attention Span
    - Embedding Storage in eNVMs
- Synergistic Evaluation
- Hardware Architecture
- Hardware Evaluation
- Conclusion

Conventional BERT inference



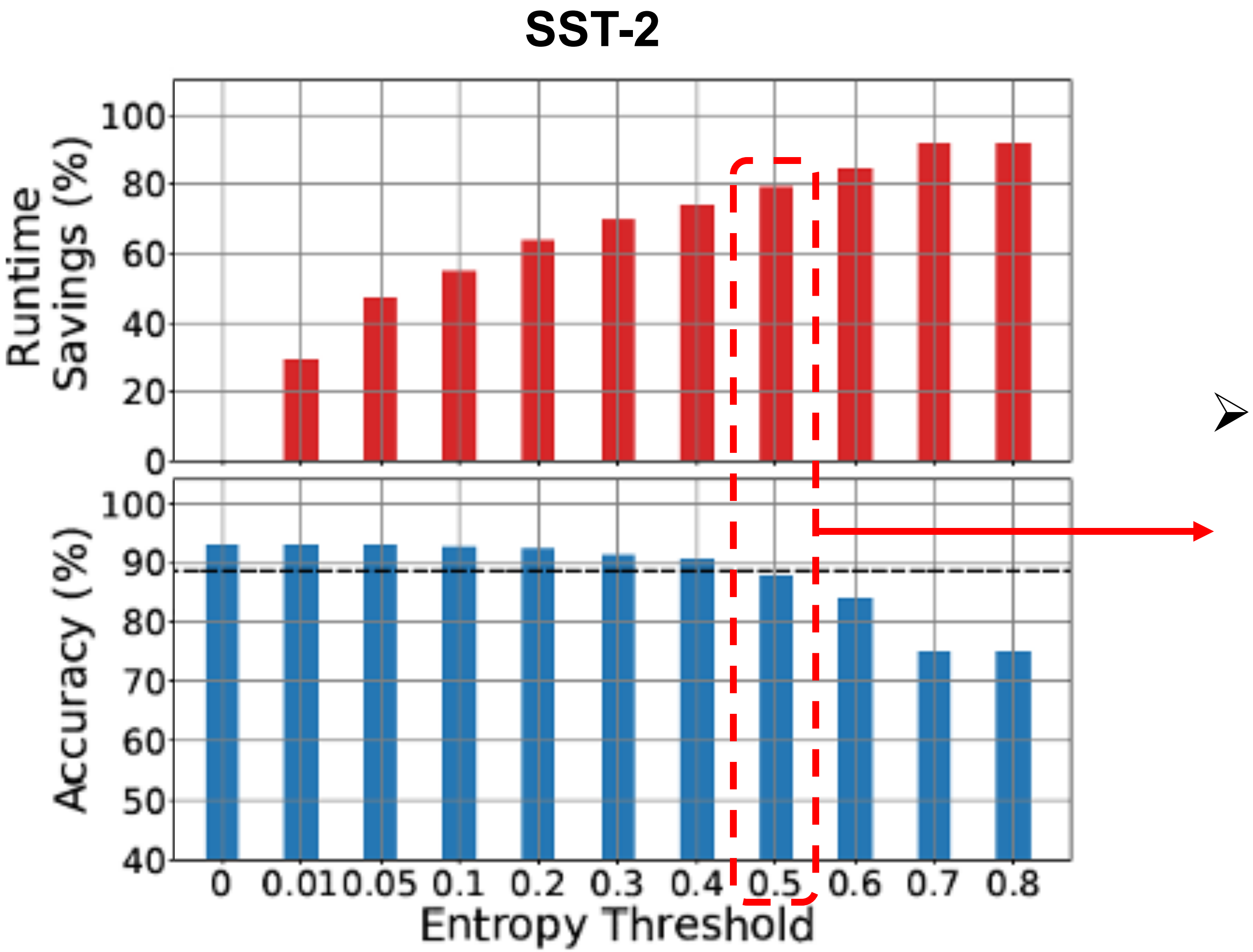
➢ **Computation goes through all 12 Transformer layers**

BERT inference with early exit



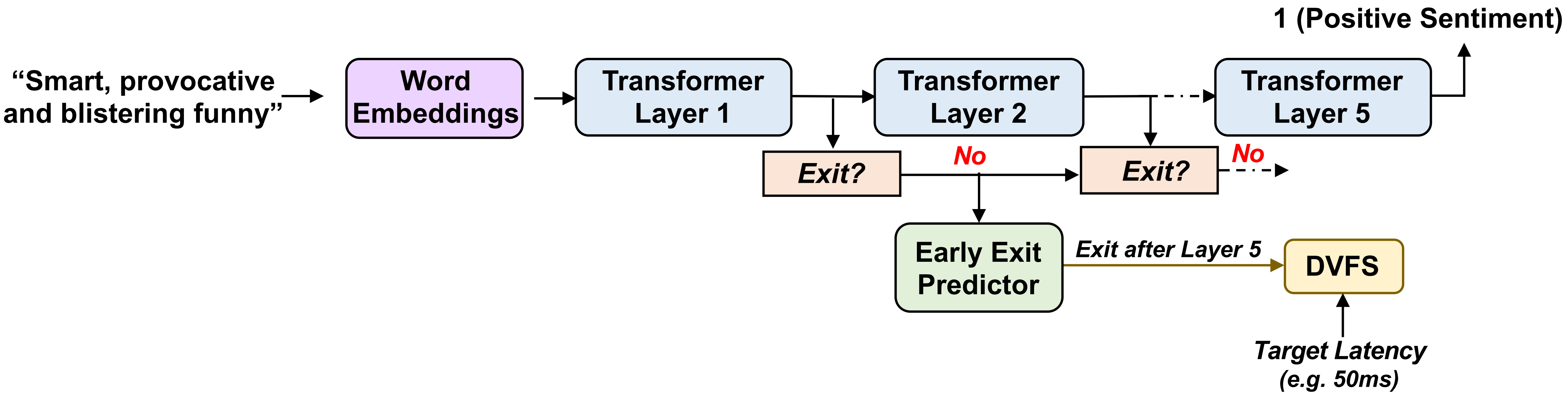
- ➢ Inference exits early if the entropy is smaller than a user-given threshold

Early exit achieves significant latency savings



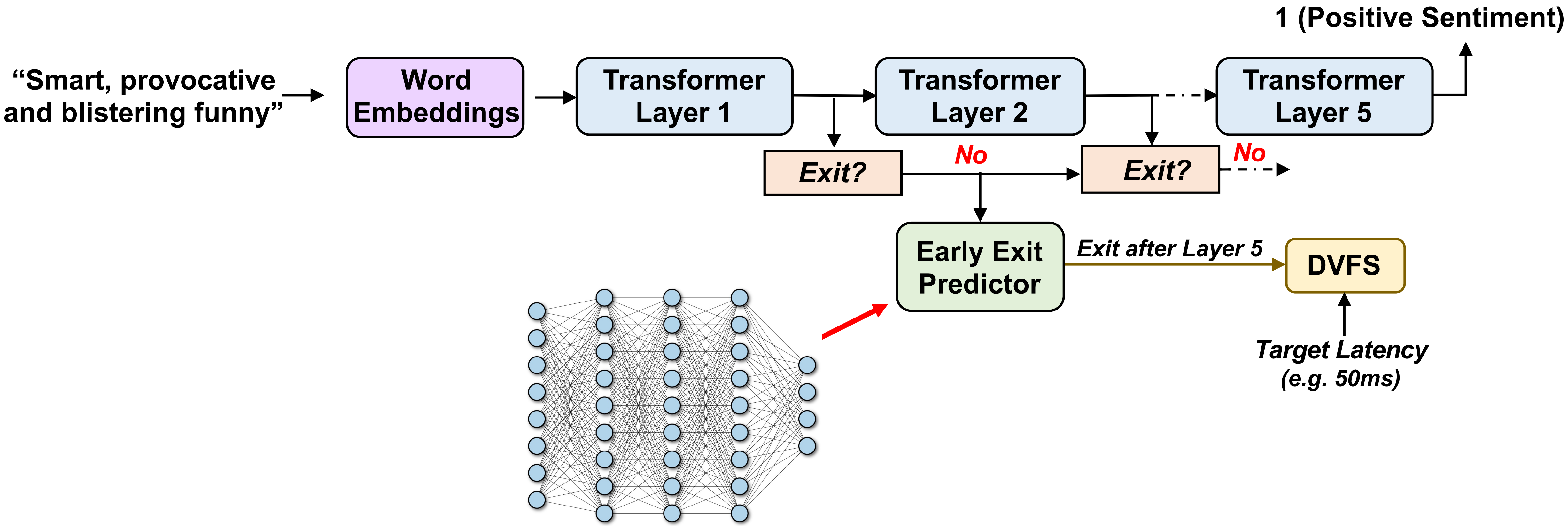
- ➢ On the SST-2 task, close to 80% of BERT computations can be saved while maintaining 95% of the original accuracy.

Proposed latency-aware inference



- **DVFS uses the predicted early exit layer to lower the energy consumption during a sentence inference**

Proposed latency-aware inference



1 (Positive Sentiment)

"Smart, provocative and blistering funny"

Word Embeddings

Transformer Layer 1

Transformer Layer 2

Transformer Layer 5

Exit?

No

Exit?

No

Early Exit Predictor

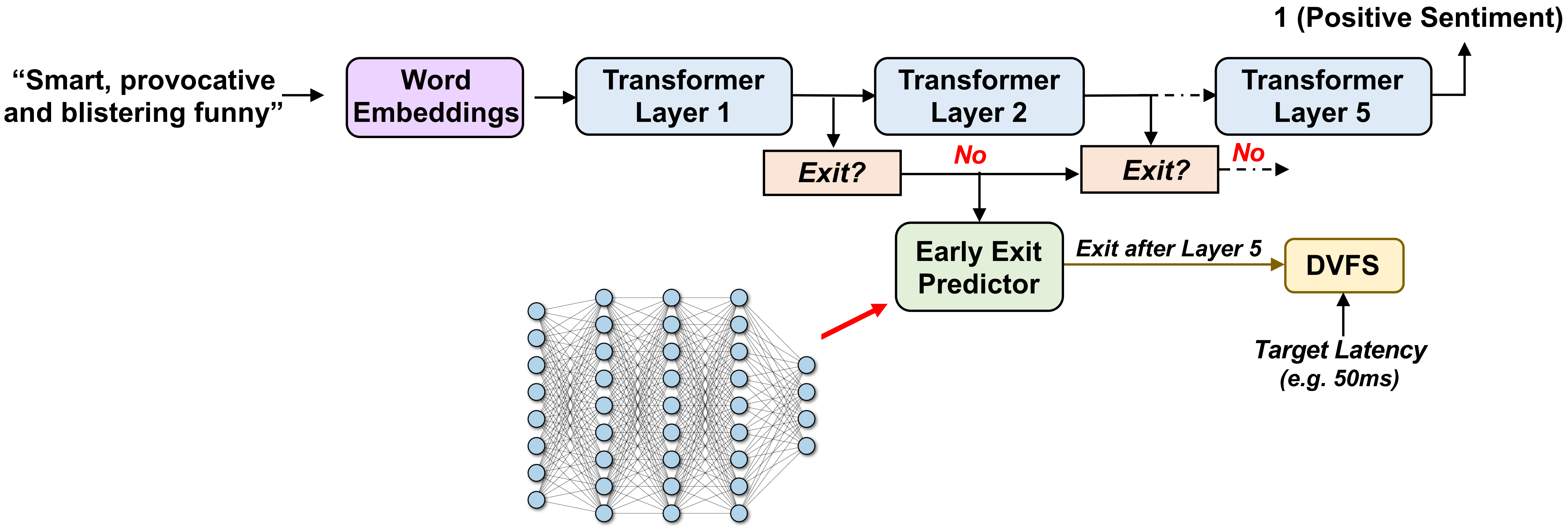
Exit after Layer 5

DVFS

Target Latency (e.g. 50ms)

- ➢ **Early exit predictor is a 5-layer neural network perceptron**

Proposed latency-aware inference

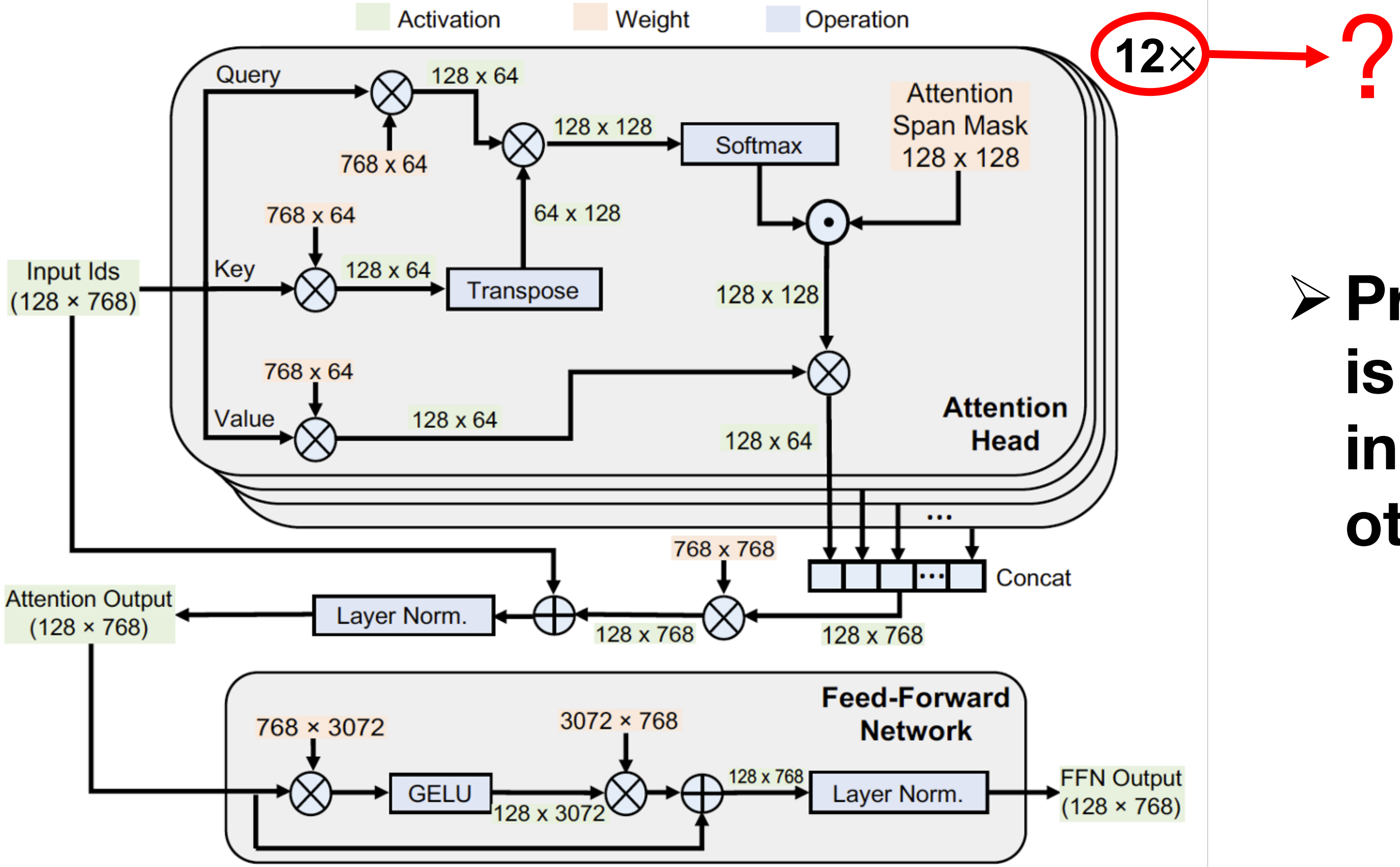


- **Accounts for up to 45% of the total accelerator energy reduction**

Outline

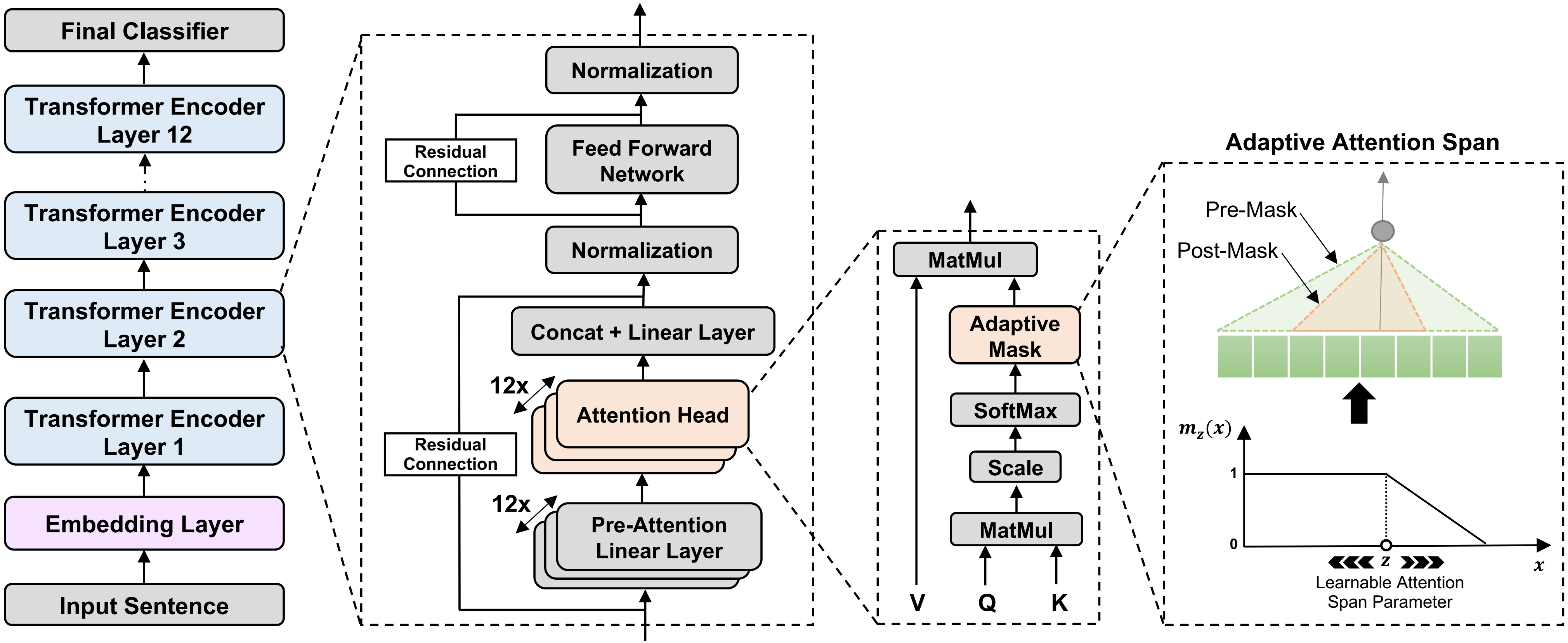
- Motivation
- EdgeBERT Optimizations
    - Entropy-based DVFS for Latency-Bounded NLP Inference
    - Adaptive Attention Span
    - Embedding Storage in eNVMs
- Synergistic Evaluation
- Hardware Architecture
- Hardware Evaluation
- Conclusion

Does BERT really need 12 attention heads?

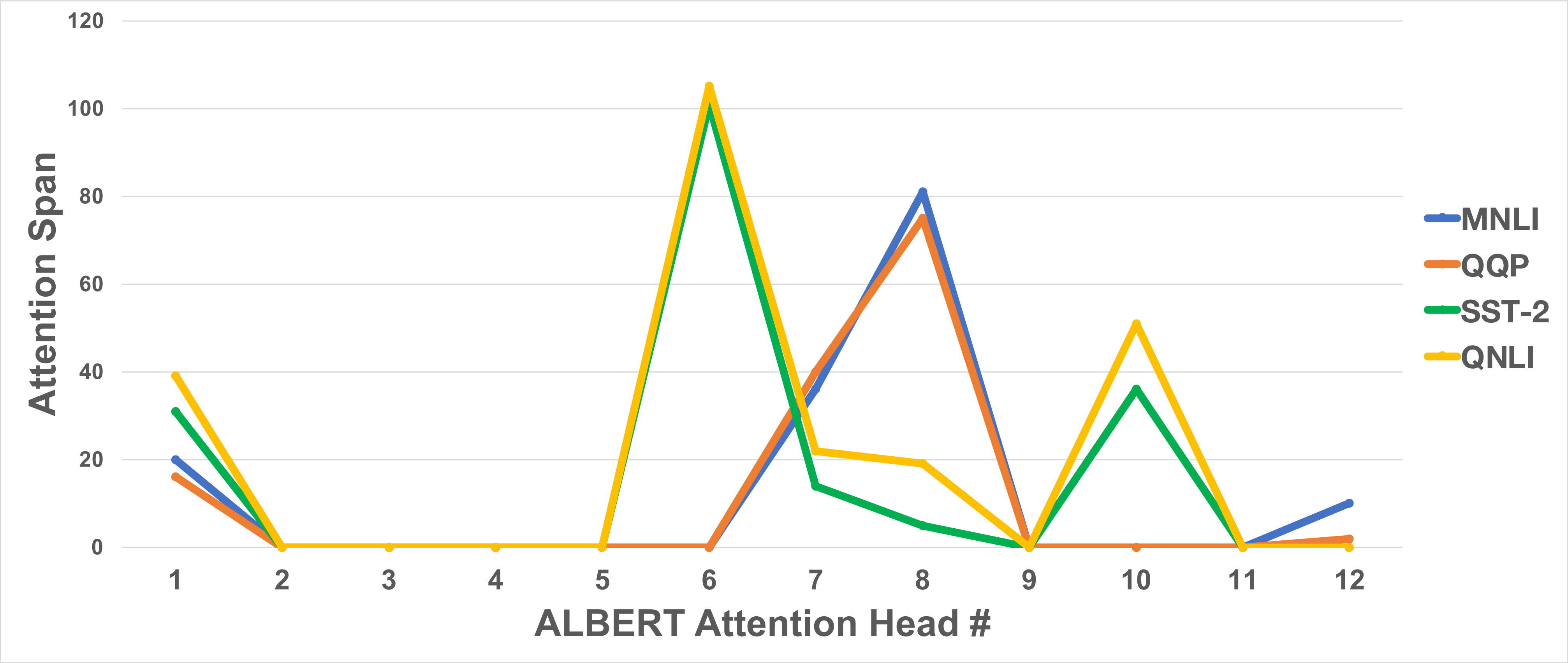


➢ **Prior work have shown that there is a large amount of redundancy in attention heads in BERT and other Transformer-based models**

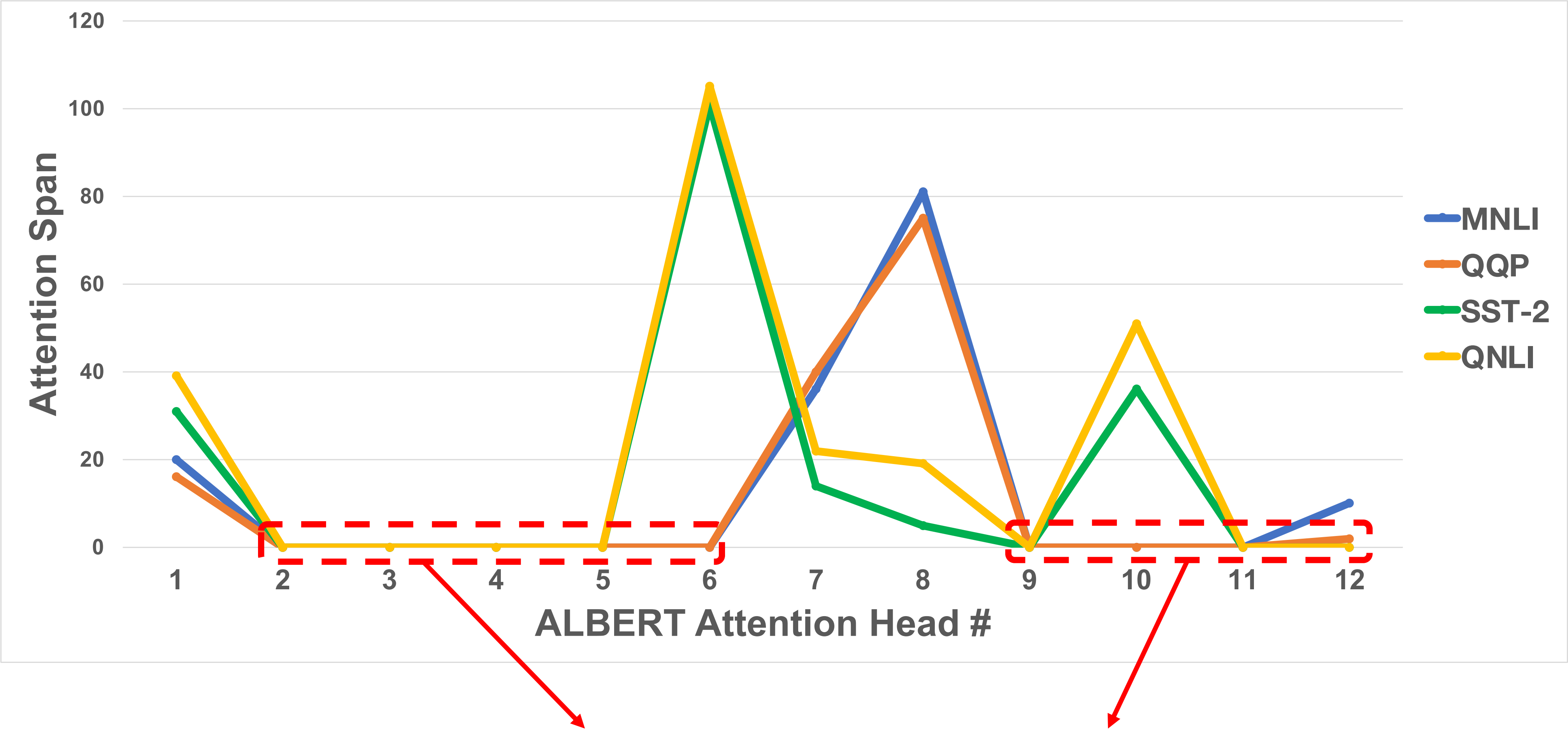
EdgeBERT optimizes the attention span of each head during finetuning



Many attention heads can be turned off

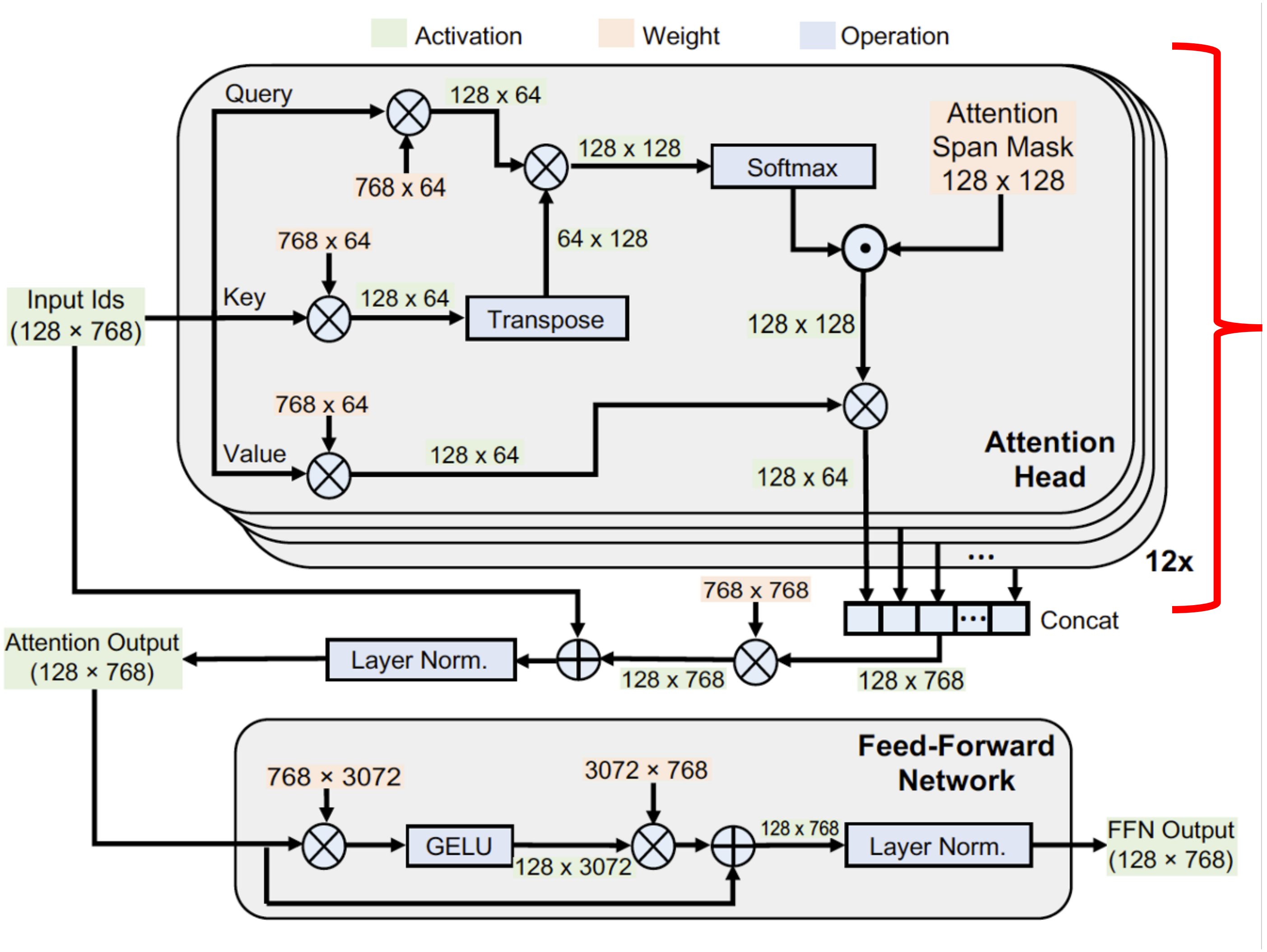


Many attention heads can be turned off



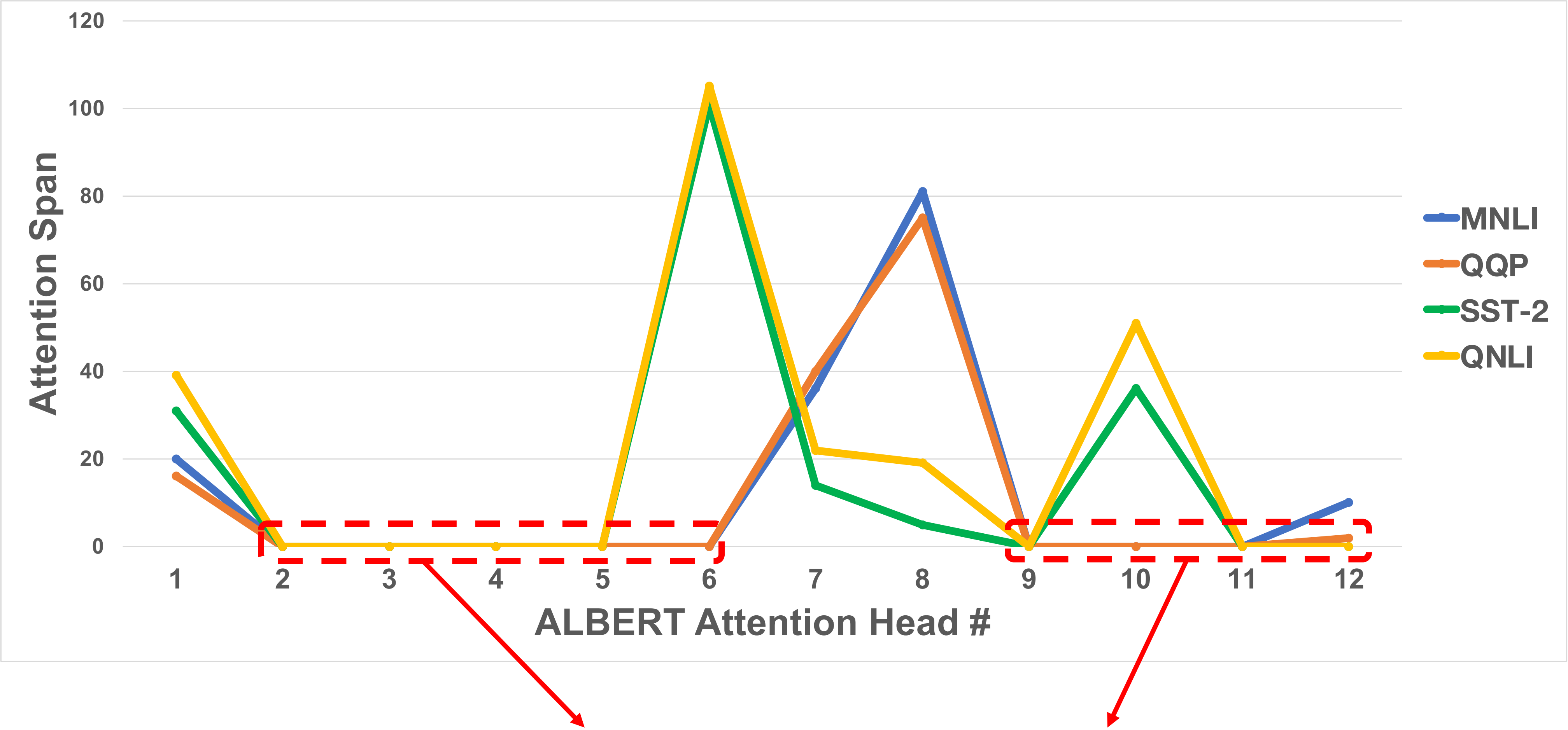
- ➢ **Finetuning results show that up to half of ALBERT attention heads can be completed turned off prior to inference!**

HW Implications



All the computations inside an attention head can be effectively skipped in case its associated attention span mask is null!

Many attention heads can be turned off

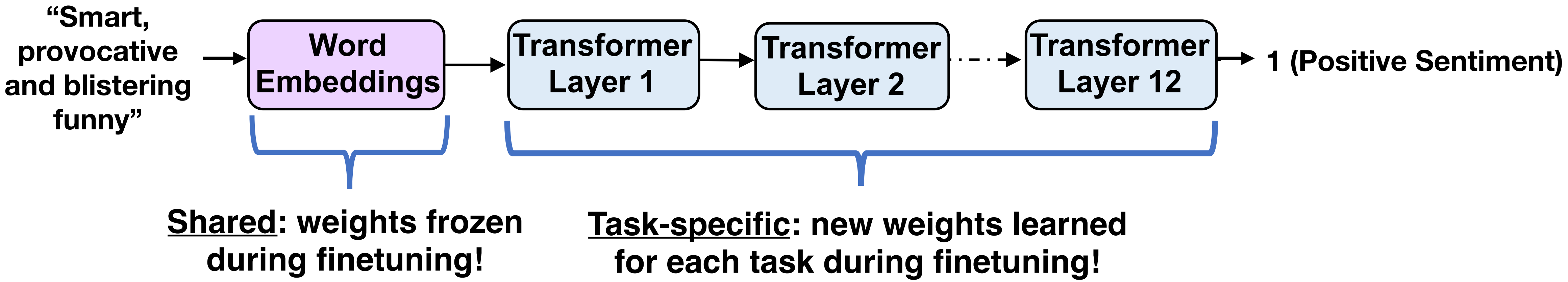


- ➢ **Adaptive attention span accounts for up to 12% of the total accelerator energy reduction**

Outline

- Motivation
- EdgeBERT Optimizations
    - Entropy-based DVFS for Latency-Bounded NLP Inference
    - Adaptive Attention Span
    - Embedding Storage in eNVMs
- Synergistic Evaluation
- Hardware Architecture
- Hardware Evaluation
- Conclusion

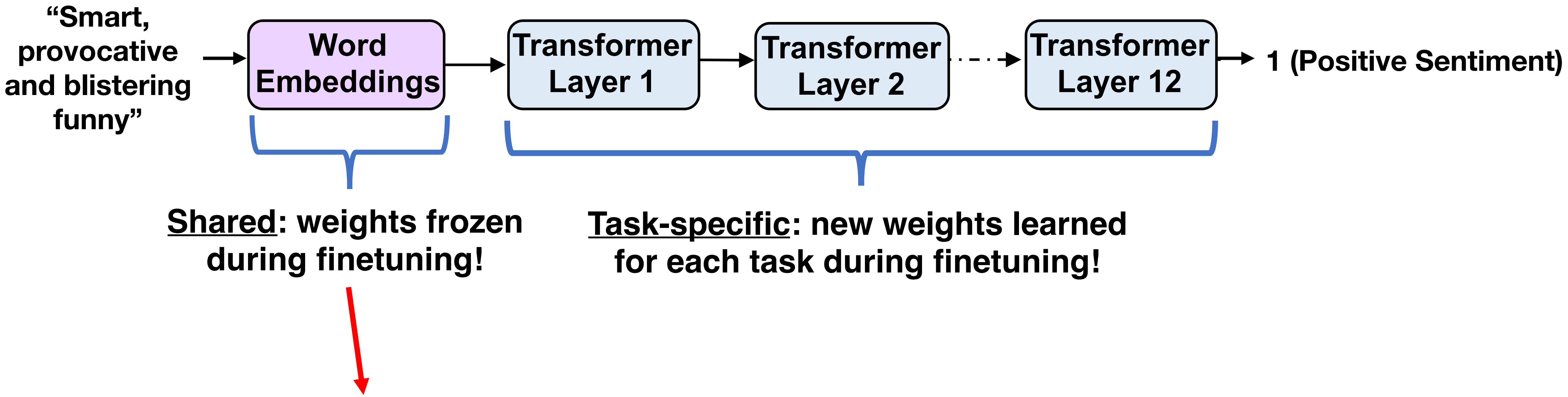
BERT word embeddings are appealing for storage in non-volatile memories



Shared: weights frozen during finetuning!

Task-specific: new weights learned for each task during finetuning!

BERT word embeddings are appealing for storage in non-volatile memories



“Smart, provocative and blistering funny”

Word Embeddings

Transformer Layer 1

Transformer Layer 2

Transformer Layer 12

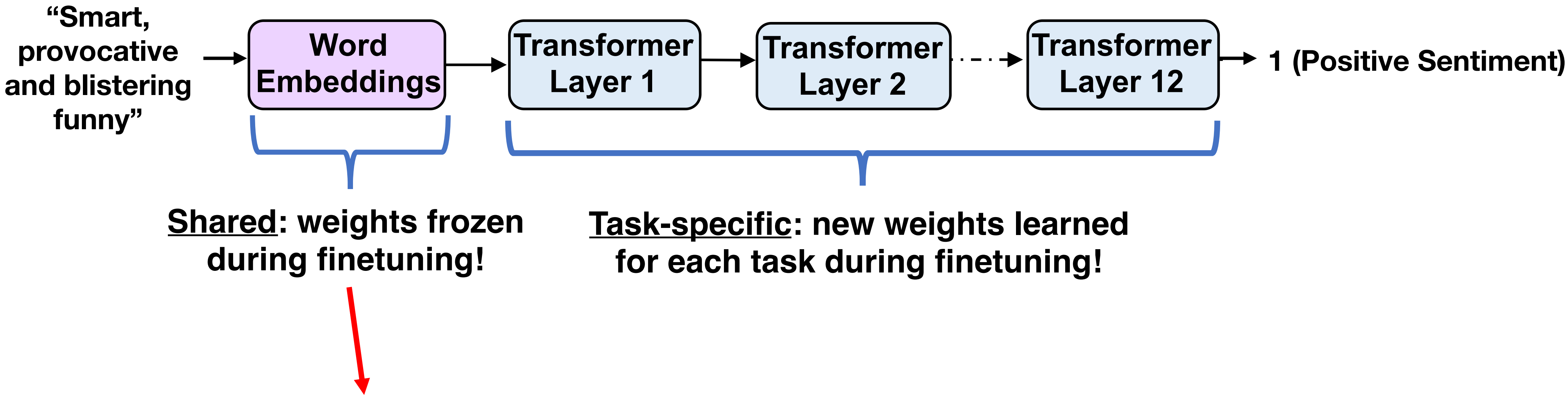
1 (Positive Sentiment)

Shared: weights frozen during finetuning!

Task-specific: new weights learned for each task during finetuning!

- **BERT word embeddings become read-only, therefore are a good match for NVM storage**

BERT word embeddings are appealing for storage in non-volatile memories



“Smart, provocative and blistering funny”

Word Embeddings

Transformer Layer 1

Transformer Layer 2

Transformer Layer 12

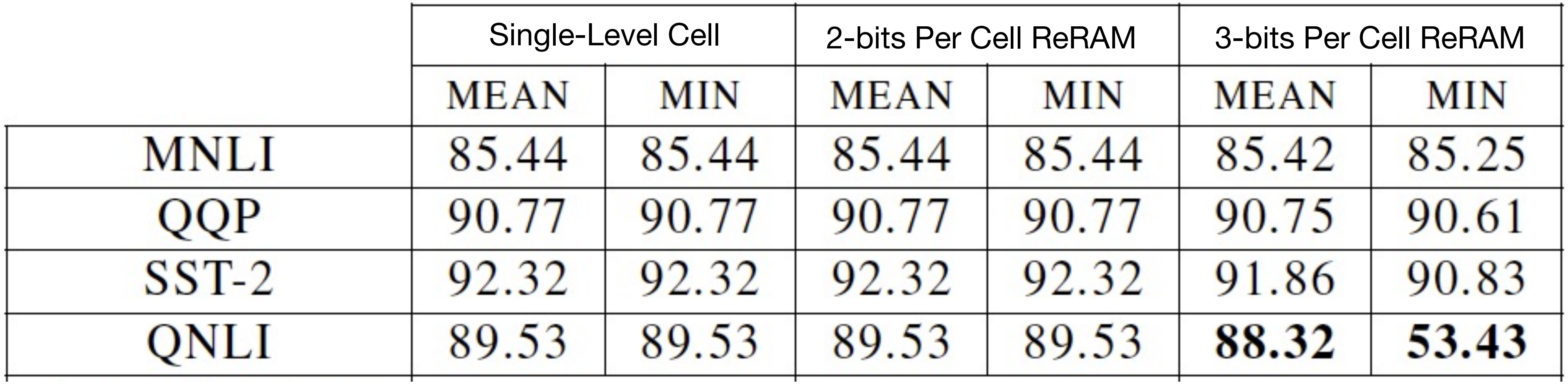
1 (Positive Sentiment)

Shared: weights frozen during finetuning!

Task-specific: new weights learned for each task during finetuning!

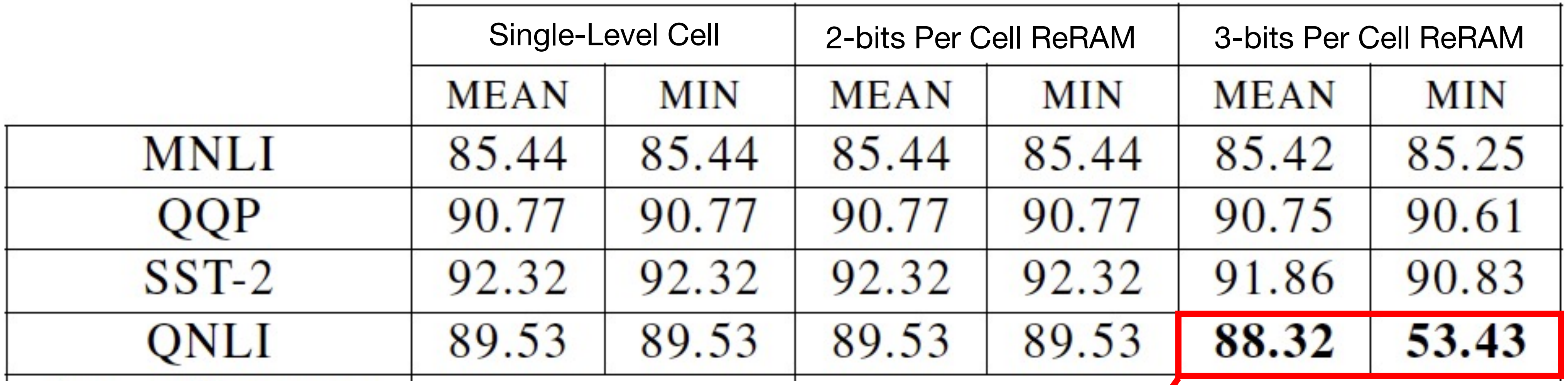
- ➢ **NVM provides benefit during intermittent operation**
  - ➢ **Obviates need to reload word embeddings from off-chip DRAM**

Viability of Multi-Level Cell ReRAM for Word Embedding Storage



	Single-Level Cell		2-bits Per Cell ReRAM		3-bits Per Cell ReRAM	
	MEAN	MIN	MEAN	MIN	MEAN	MIN
MNLI	85.44	85.44	85.44	85.44	85.42	85.25
QQP	90.77	90.77	90.77	90.77	90.75	90.61
SST-2	92.32	92.32	92.32	92.32	91.86	90.83
QNLI	89.53	89.53	89.53	89.53	**88.32**	**53.43**

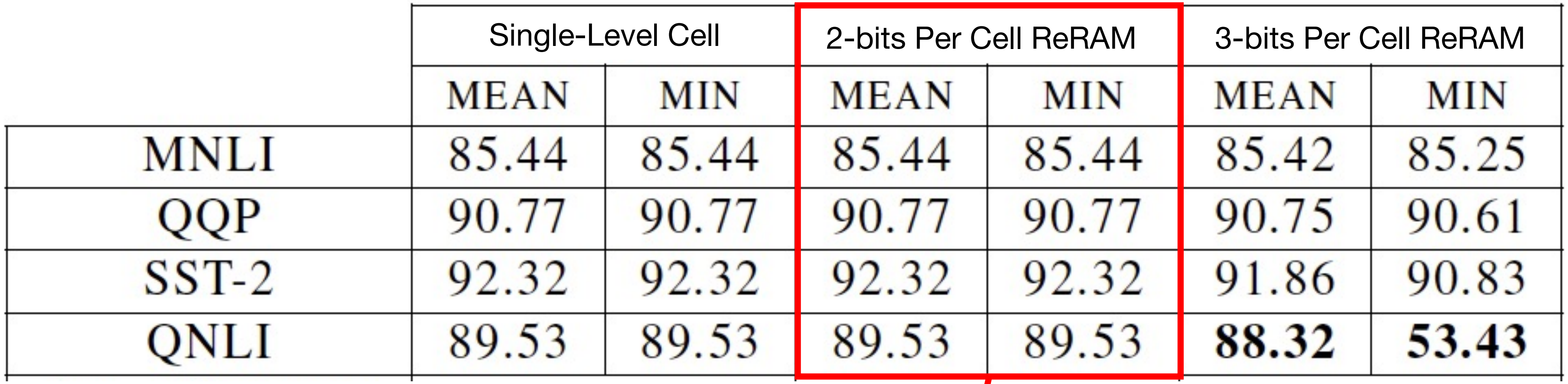
Viability of Multi-Level Cell ReRAM for Word Embedding Storage



	Single-Level Cell		2-bits Per Cell ReRAM		3-bits Per Cell ReRAM	
	MEAN	MIN	MEAN	MIN	MEAN	MIN
MNLI	85.44	85.44	85.44	85.44	85.42	85.25
QQP	90.77	90.77	90.77	90.77	90.75	90.61
SST-2	92.32	92.32	92.32	92.32	91.86	90.83
QNLI	89.53	89.53	89.53	89.53	**88.32**	**53.43**

- **3-bits per Cell ReRAM shows vulnerability**

Viability of Multi-Level Cell ReRAM for Word Embedding Storage



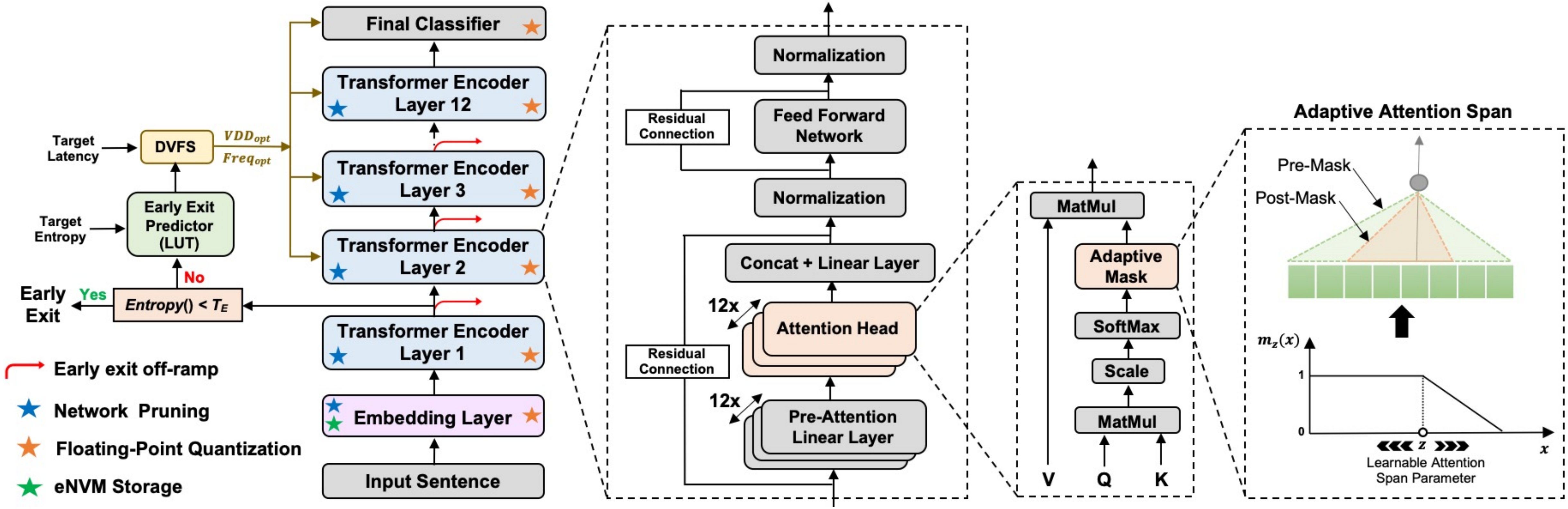
	Single-Level Cell		2-bits Per Cell ReRAM		3-bits Per Cell ReRAM	
	MEAN	MIN	MEAN	MIN	MEAN	MIN
MNLI	85.44	85.44	85.44	85.44	85.42	85.25
QQP	90.77	90.77	90.77	90.77	90.75	90.61
SST-2	92.32	92.32	92.32	92.32	91.86	90.83
QNLI	89.53	89.53	89.53	89.53	**88.32**	**53.43**

- **The EdgeBERT accelerator system leverages MLC2 ReRAMs for word embedding storage**

Outline

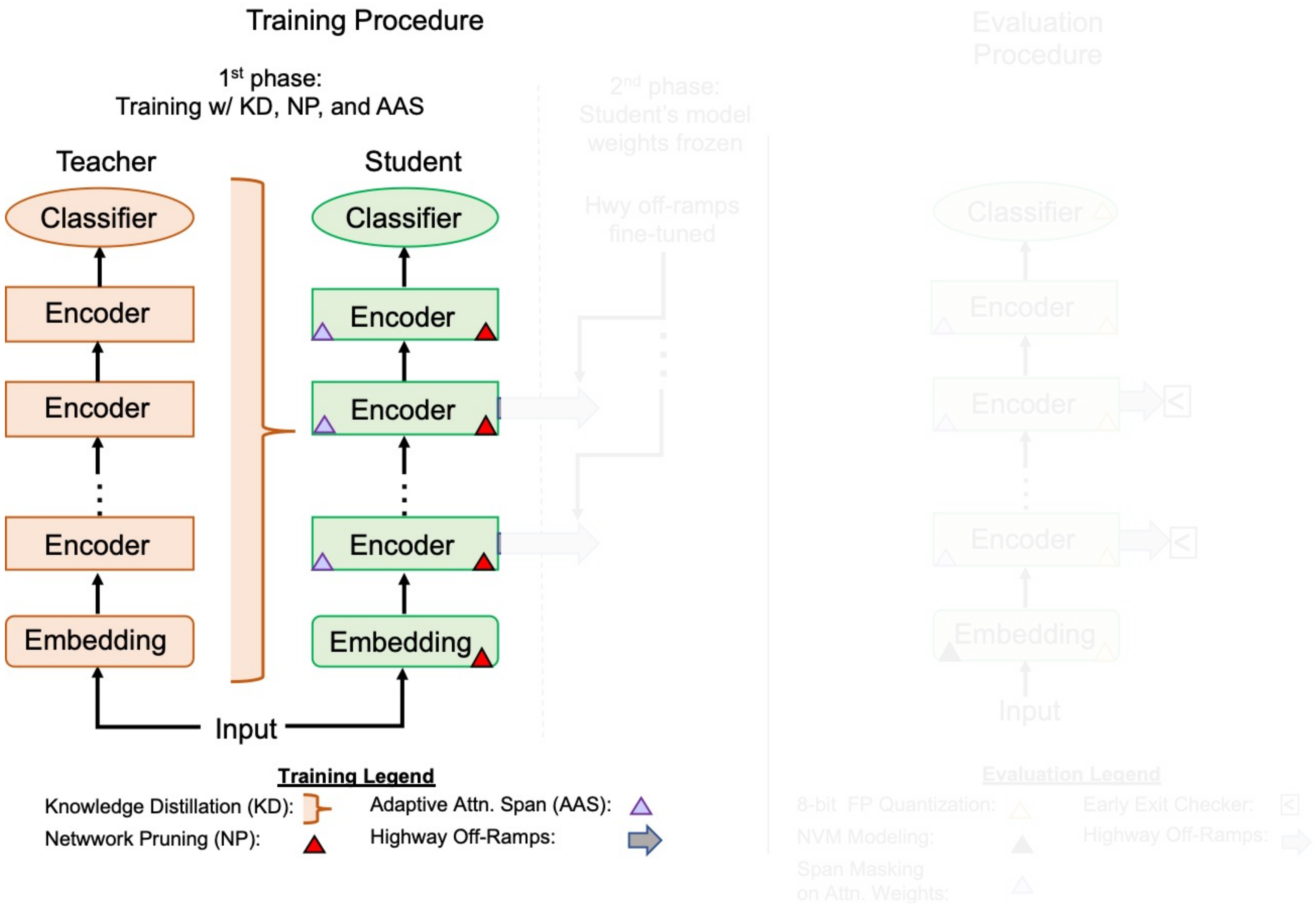
- Motivation
- EdgeBERT Optimizations
- Synergistic Evaluation
- Hardware Architecture
- Hardware Evaluation
- Conclusion

Summary of Optimizations

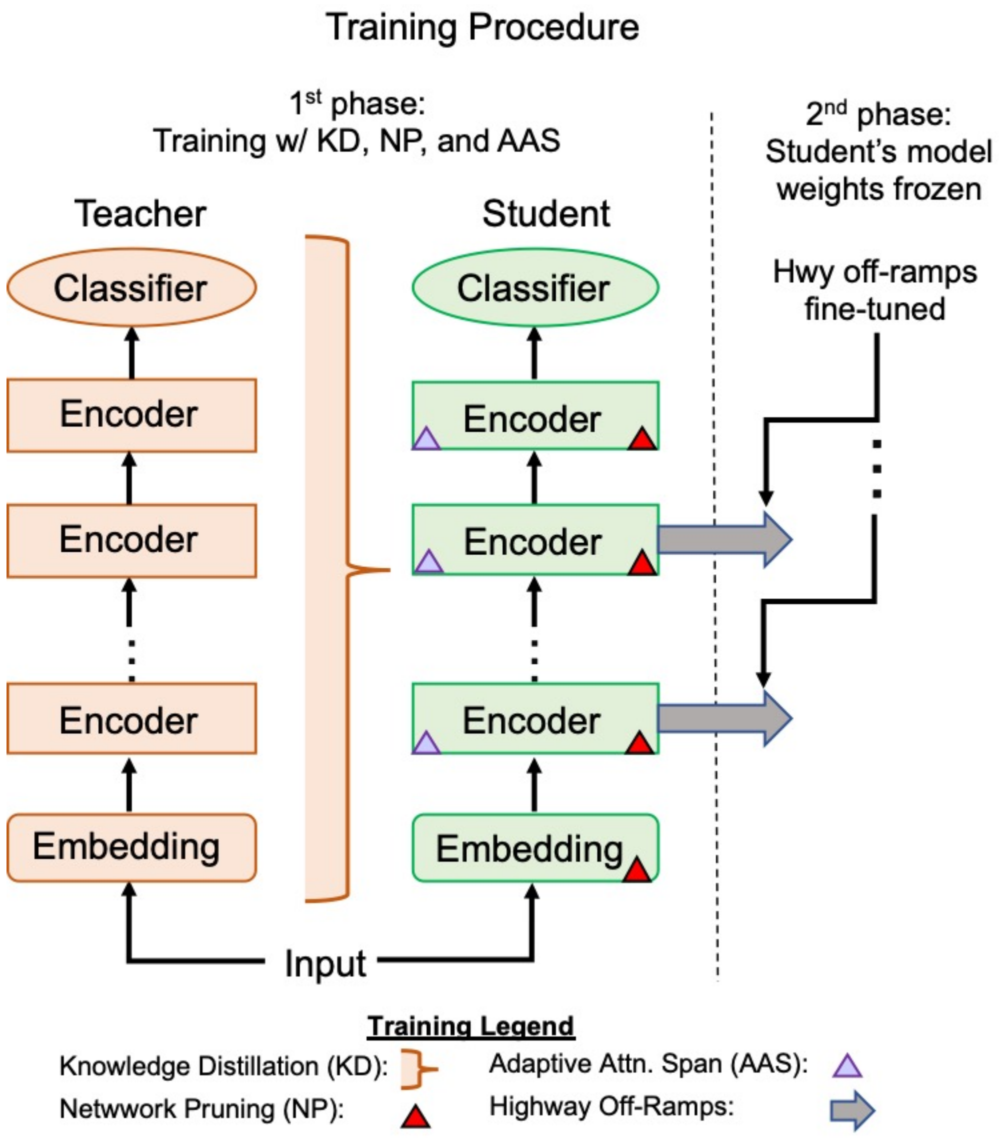


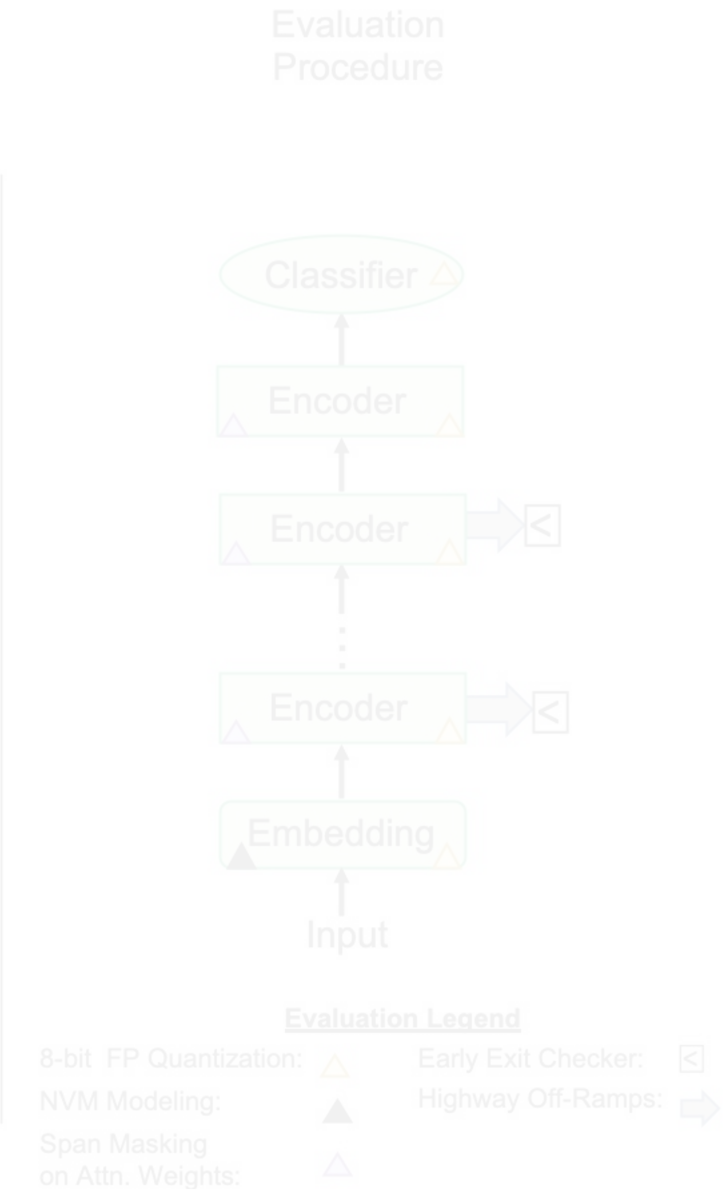
Cross-stack (algorithm, architecture, solid-state) optimizations for multi-task NLP inference

Training and Evaluation Procedure

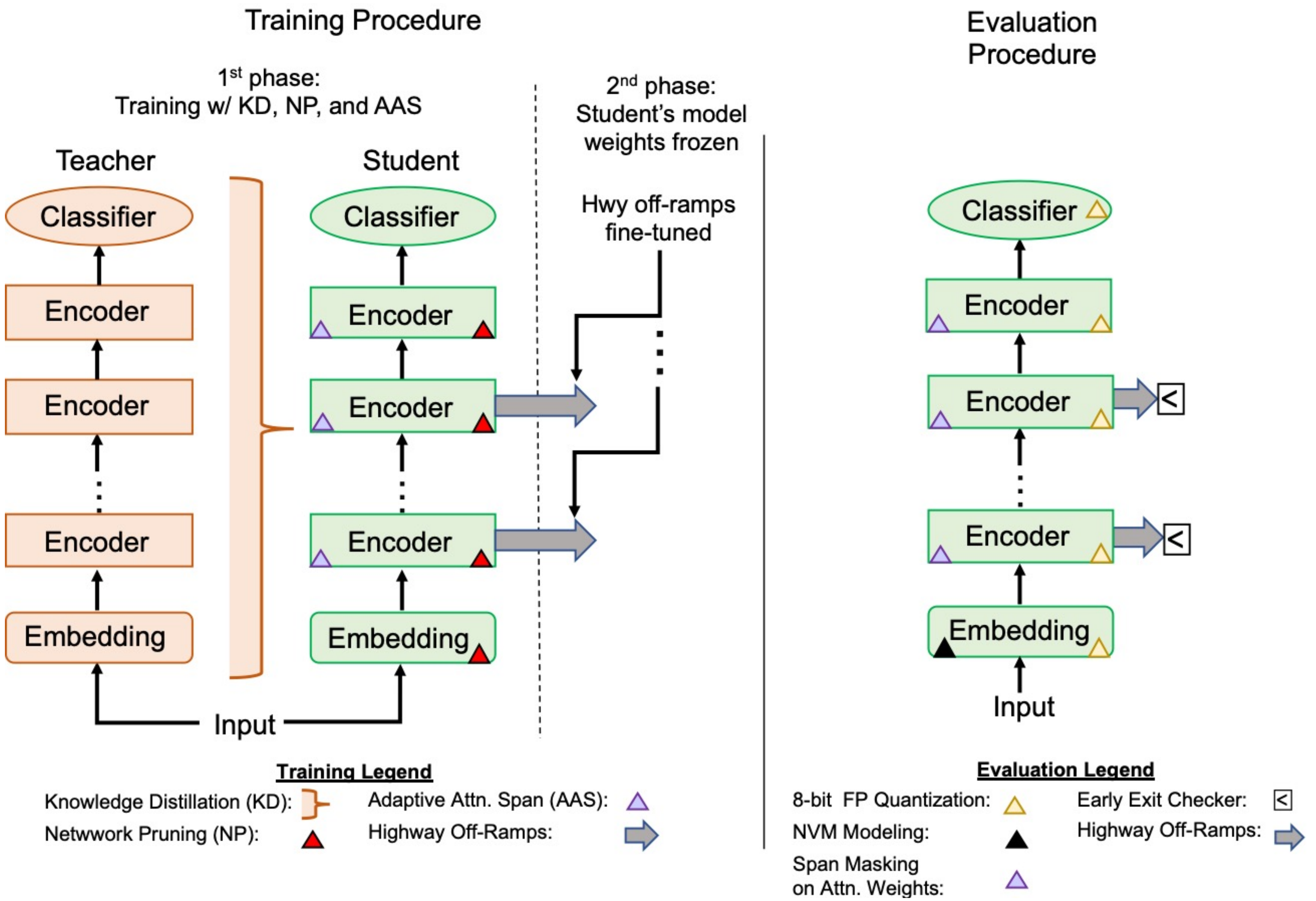


Training and Evaluation Procedure

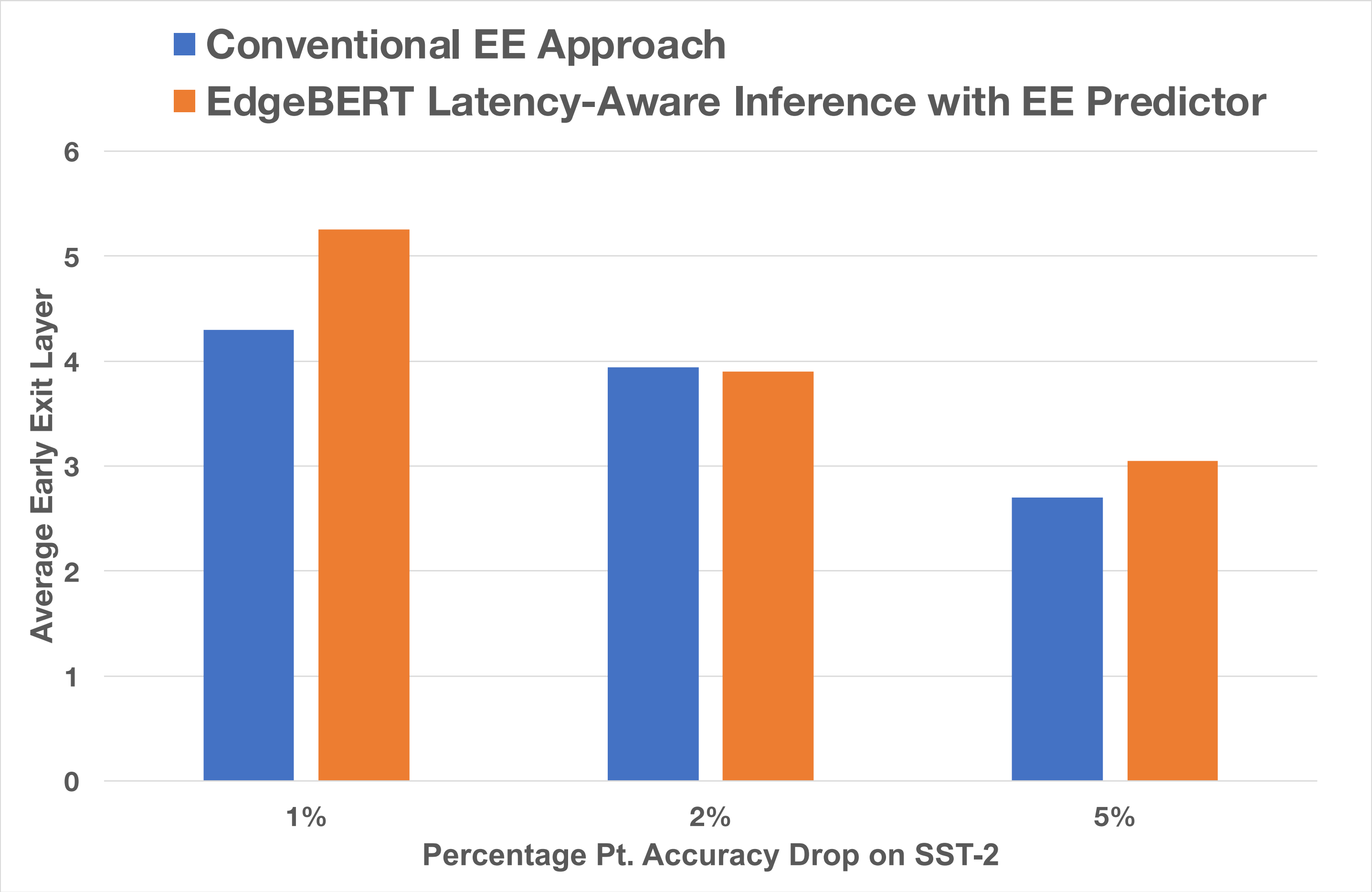




Training and Evaluation Procedure



Performance and Accuracy Implications



EdgeBERT latency-aware inference provides slightly higher or comparable average EE layer for the same accuracy threshold as the conventional EE approach

Reasonably Compact NVM Capacity

	Embedding Sparsity (%)	Embedding Sparsity (%)	Avg. Attn. Span
MNLI	60	50	12.7
QQP	60	80	11.3
SST-2	60	50	18.4
QNLI	60	60	21.5

➢ **40% density in the embedding layer across all tasks, i.e. ~2MB can be provisioned for on-chip ReRAM storage**

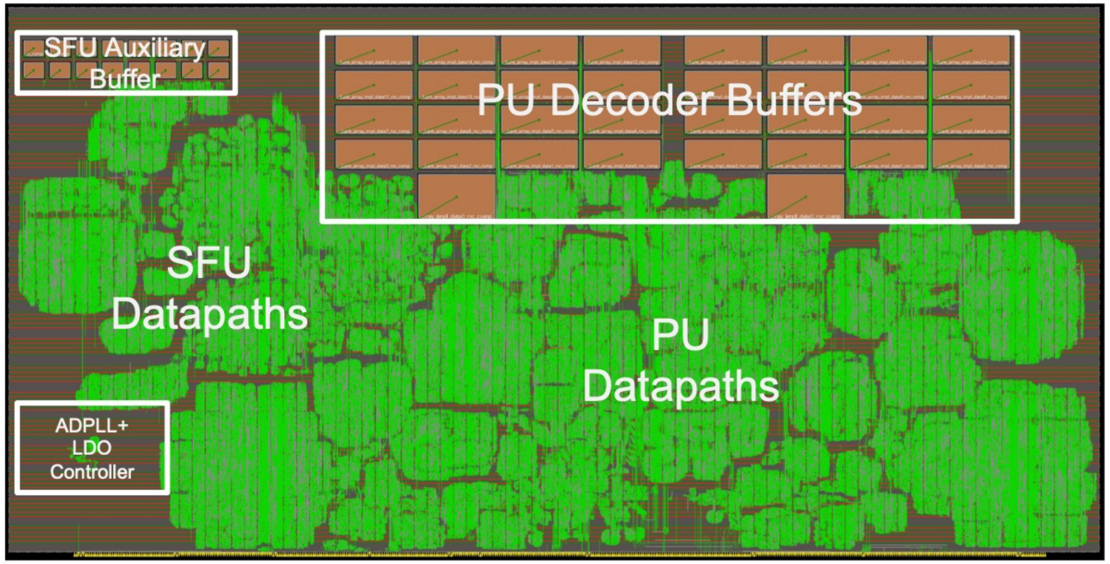
Ultra Low Attention Span

	Embedding Sparsity (%)	Embedding Sparsity (%)	Avg. Attn. Span
MNLI	60	50	12.7
QQP	60	80	11.3
SST-2	60	50	18.4
QNLI	60	60	21.5

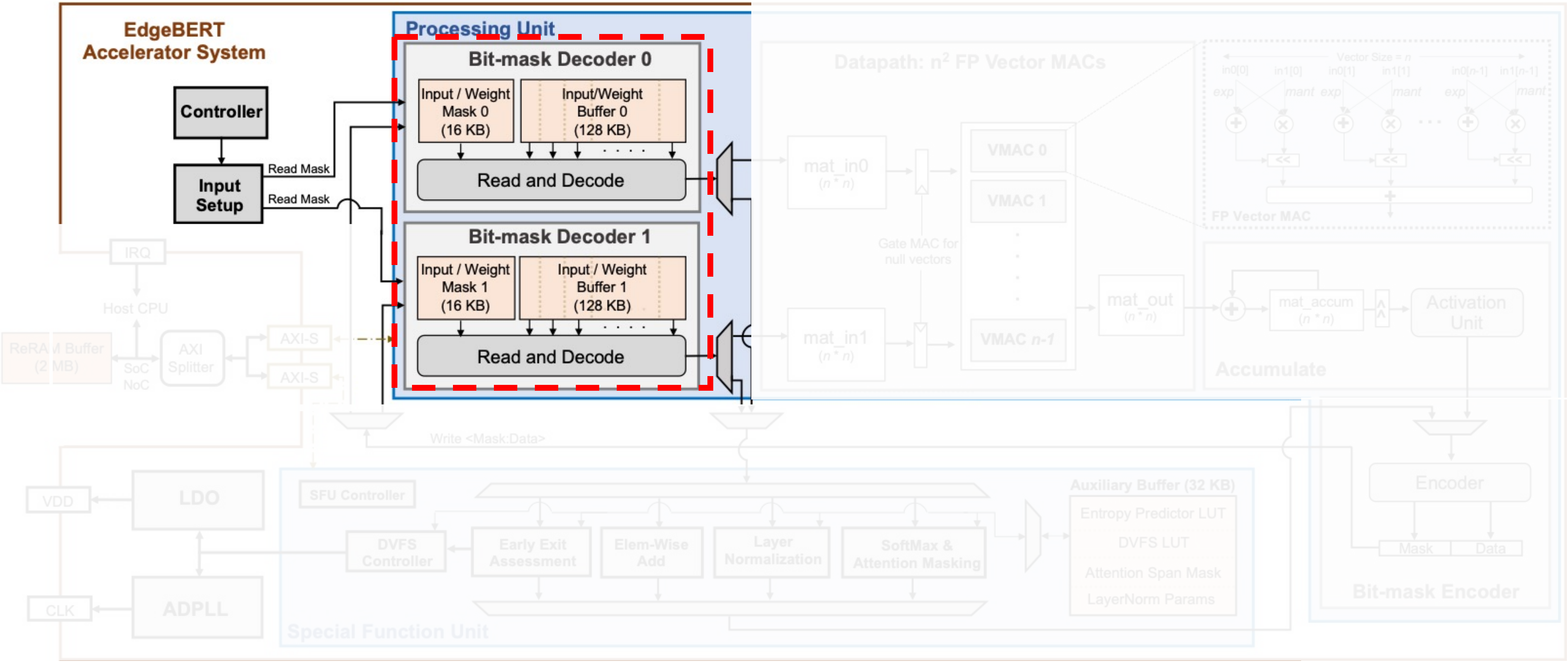
- **An average attention span less than 22**

Outline

- Motivation
- EdgeBERT Optimizations
- Synergistic Evaluation
- Hardware Architecture
- Hardware Evaluation
- Conclusion

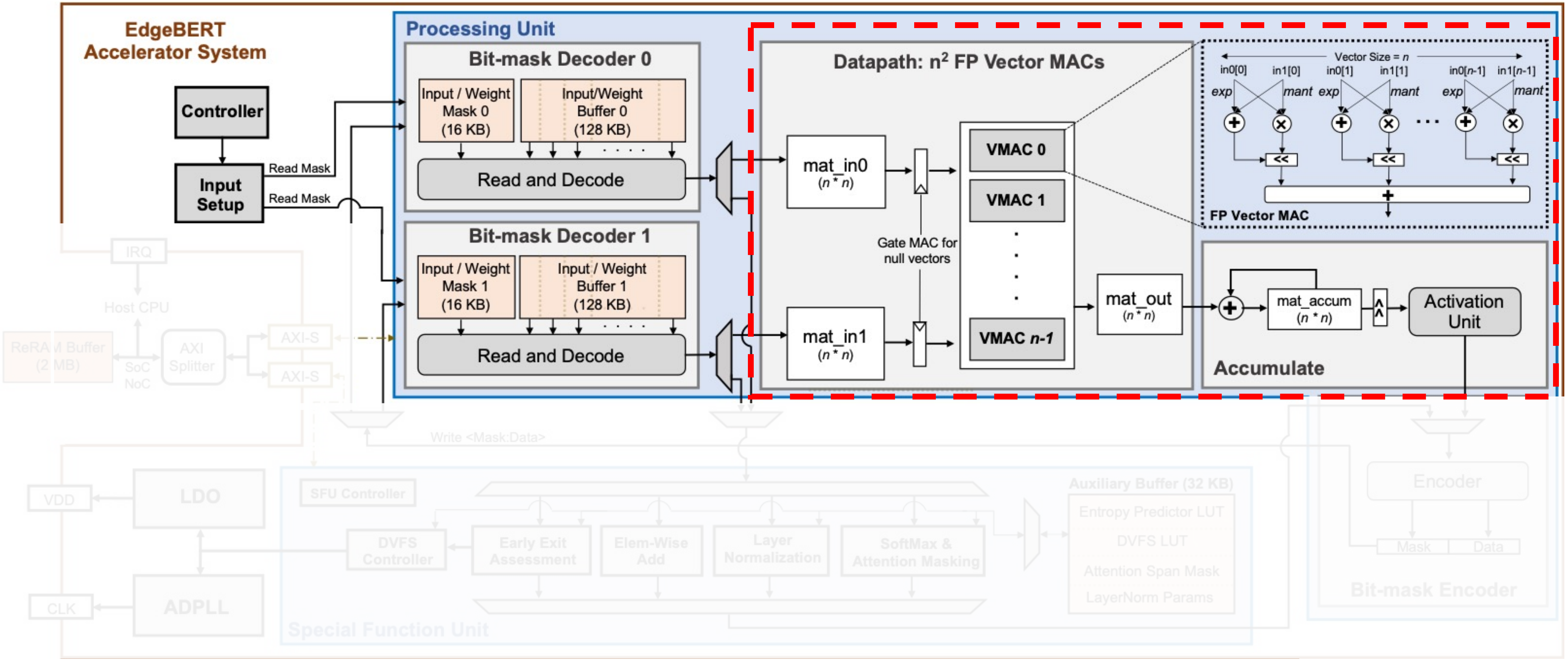


EdgeBERT Processing Unit (PU)



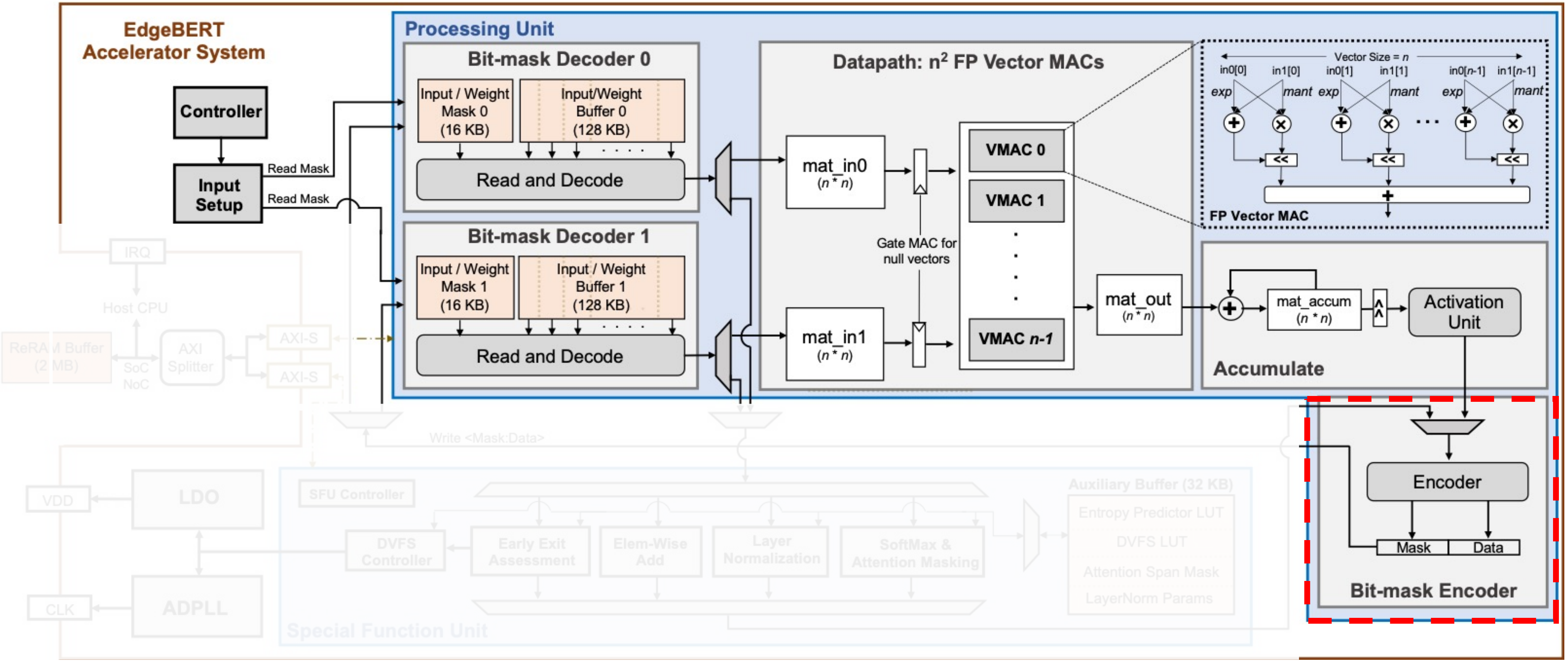
- **Bit-mask decoder for decompressing non-zero matrices**
  - **16 KB scratchpad containing binary masks for activations and weights**
  - **128 KB scratchpad containing non-zero activations and weights**

EdgeBERT Processing Unit (PU)



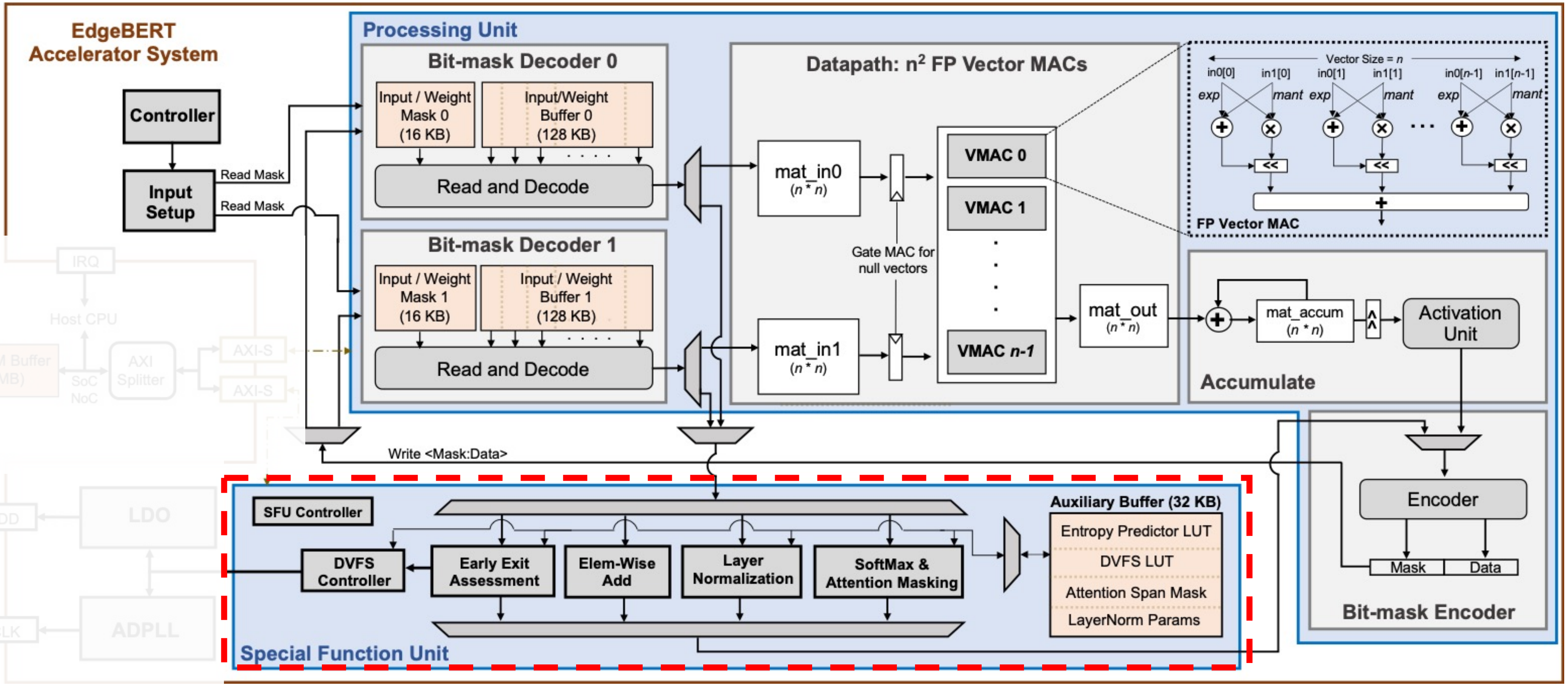
- **Datapath takes two n*n matrices and computes n*n*n MAC operations in n clock cycles**
  - **8-bit floating-point MAC**
  - **skips MAC computations on zero operands**

EdgeBERT Processing Unit (PU)



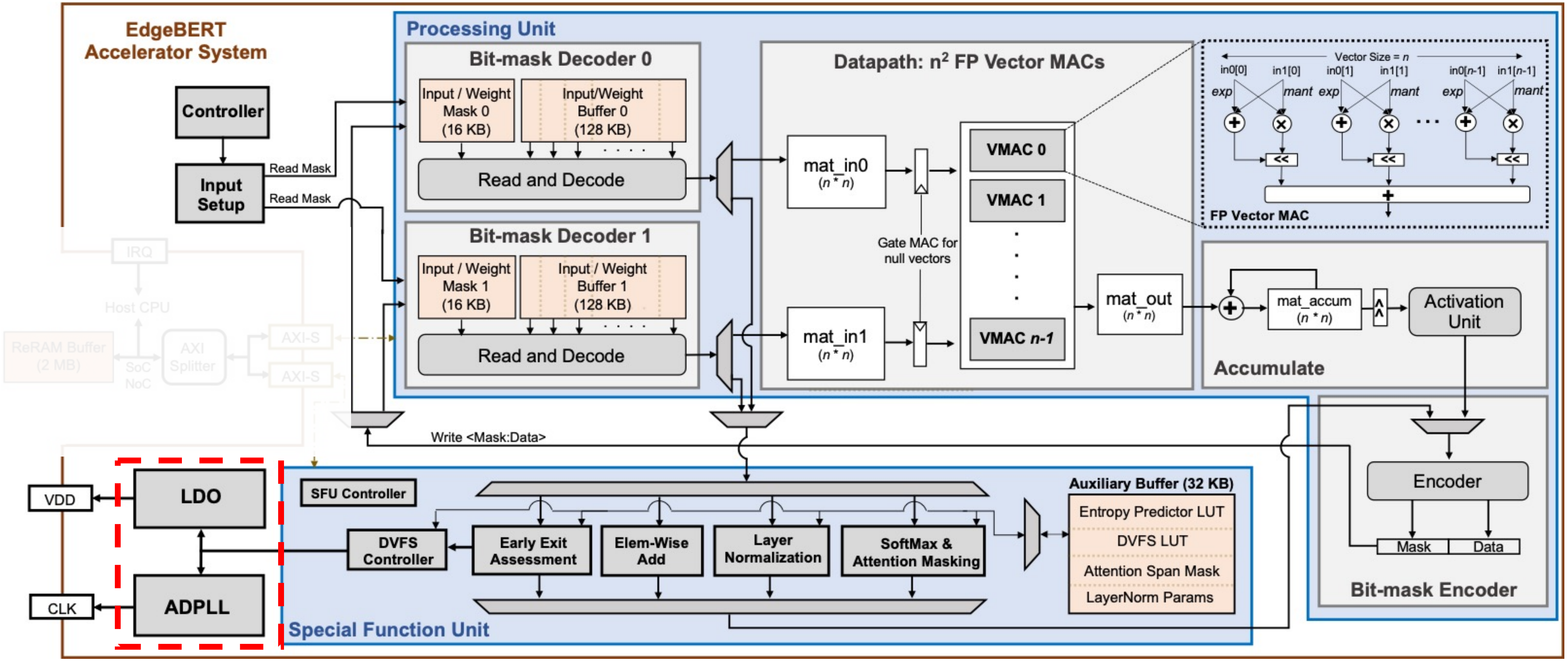
➢ **Bit-mask encoder for compressing back sparse matrices**

EdgeBERT Special Function Unit (SFU)



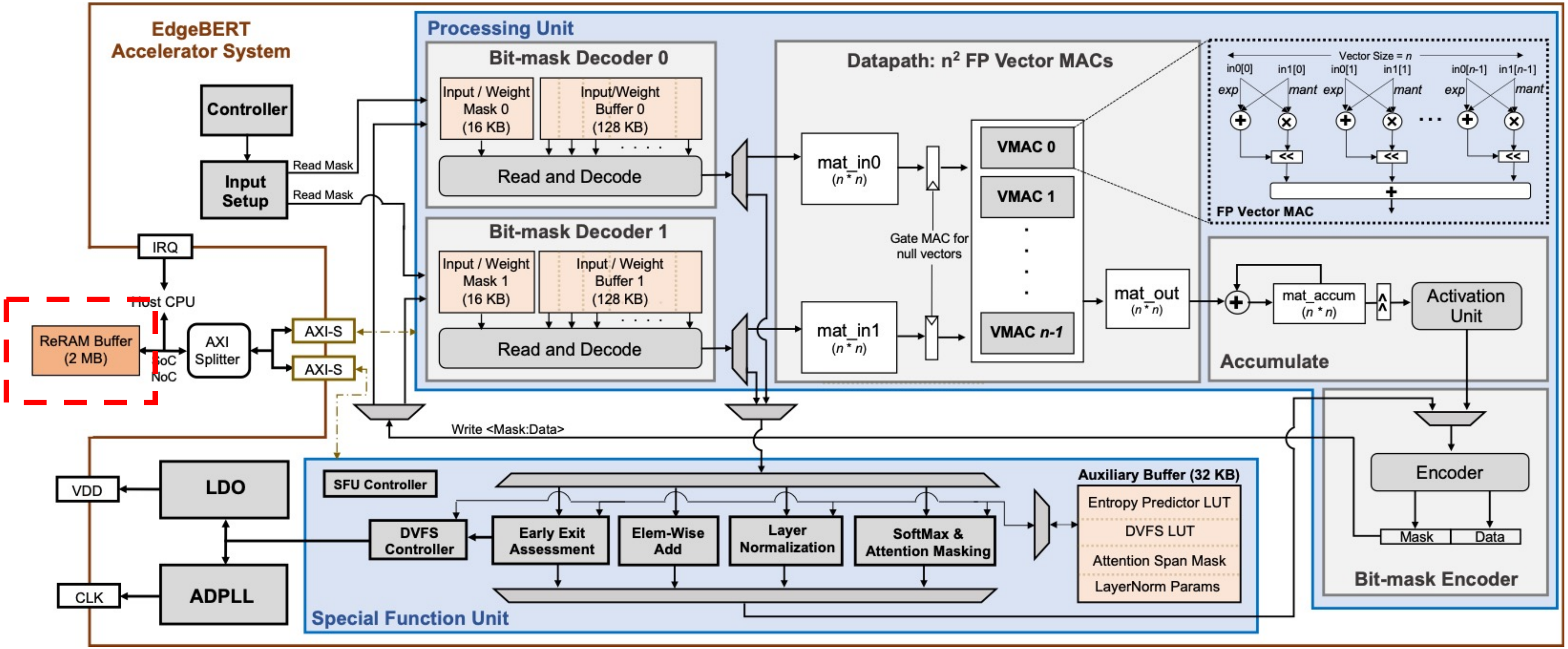
- ➢ **The special function unit (SFU) contains specialized datapaths for:**
  - ▪ **Early exit assessment, Layer Normalization, Element-wise Addition, DVFS control**
  - ▪ **SoftMax and attention masking -- only activated if attention span is not null**
- ➢ **32KB auxiliary buffer stores metadata**

Integrated LDO and ADPLL for DVFS



- ➢ **DVFS controller writes to LDO and ADPLL registers to generate energy-optimal VDD and CLK**

On-Chip 2MB ReRAM Buffer



- ➢ **2MB of on-chip MLC2 ReRAM to store the shared multi-task word embeddings**

Computing the Attention SoftMax

$$sofmax(x_i) = \frac{\exp\{x_i\}}{\sum_{j=1}^{N} \exp\{x_j\}}$$

⬇

- **Avoids numerical instability**

$$= \frac{\exp\{x_i - MAX_i(x)\}}{\sum_{j=1}^{N} \exp\{x_j - MAX_j(x)\}}$$

Computing the Attention SoftMax

$$sofmax(x_i) = \frac{\exp\{x_i\}}{\sum_{j=1}^{N} \exp\{x_j\}}$$

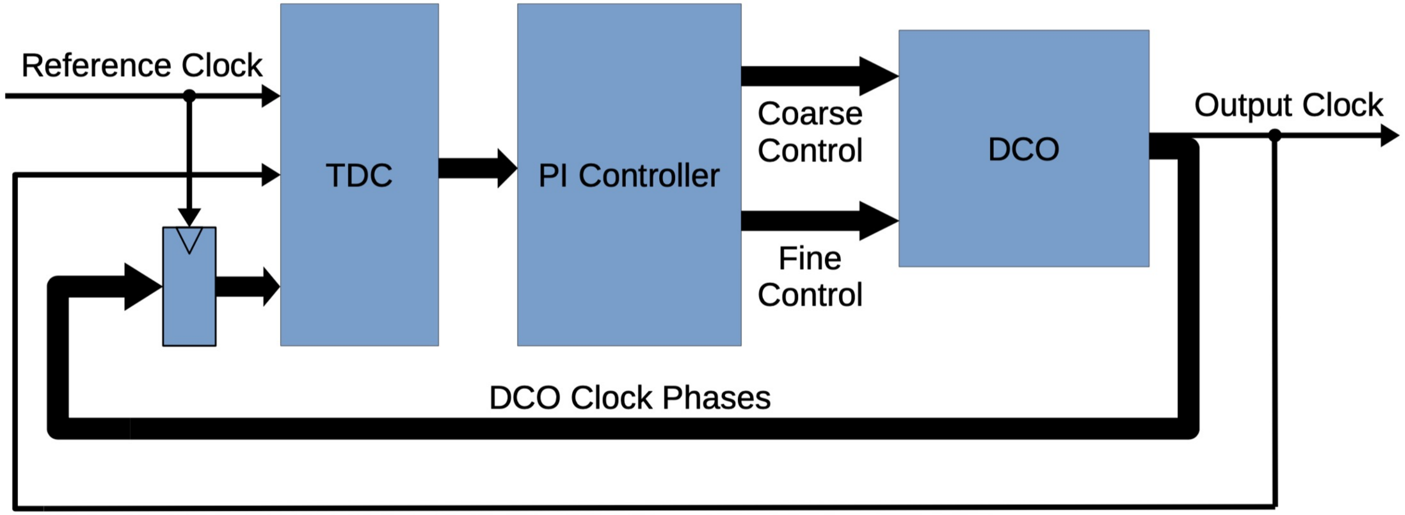
- ➢ **Suppress numerical instability**

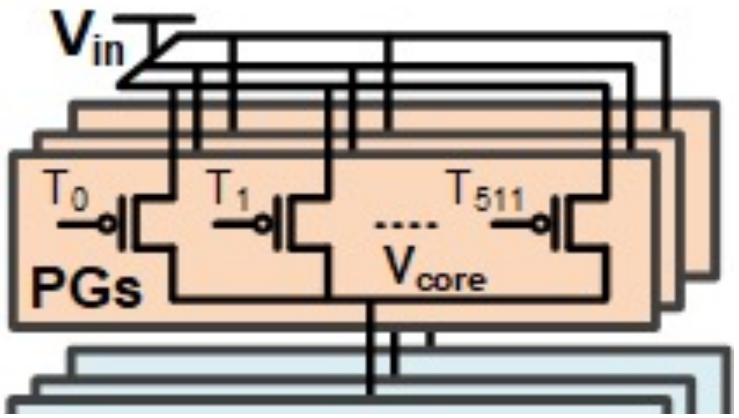
$$= \frac{\exp\{x_i - MAX_i(x)\}}{\sum_{j=1}^{N} \exp\{x_j - MAX_j(x)\}}$$

- ➢ **Eliminates computational cost of division**

$$= \exp\{x_i - MAX_i(x) - \ln(\sum_{i=1}^{N} \exp(x_i - MAX_i(x)))\}$$

ADPLL and LDO

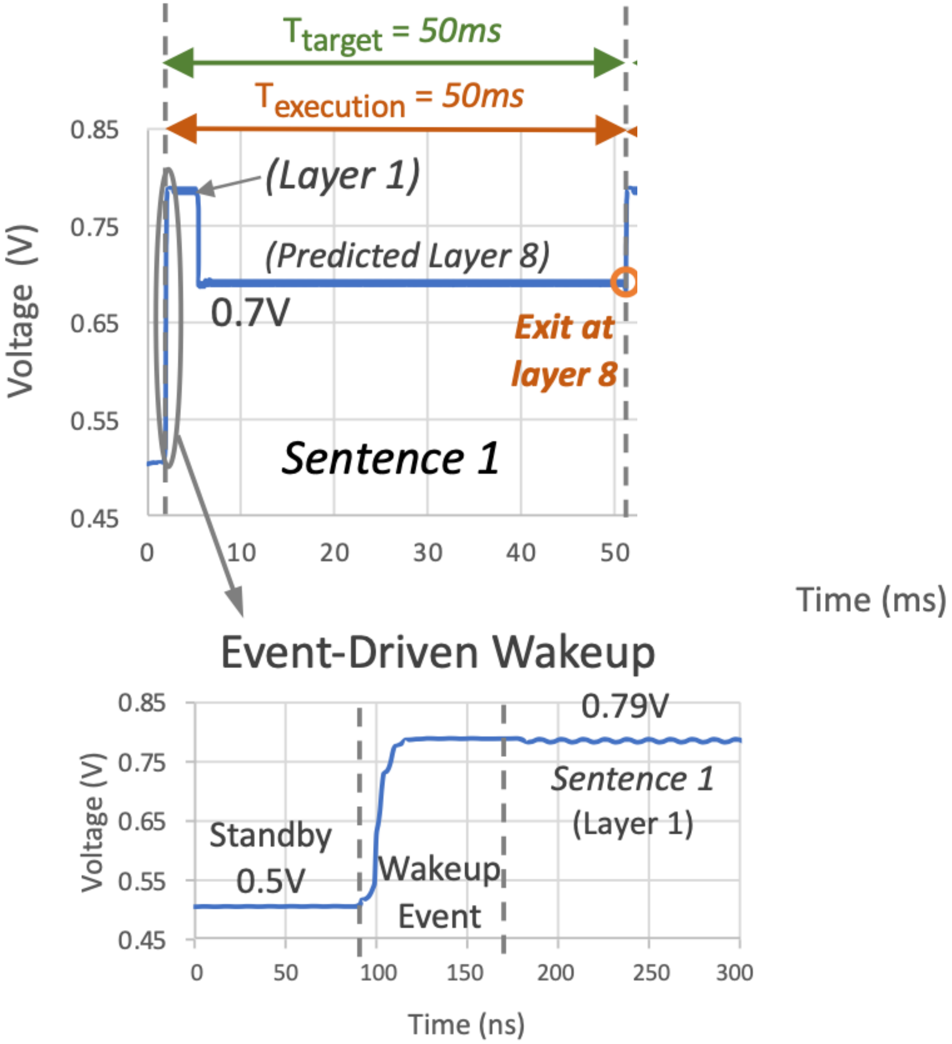




Source: https://fasoc.engin.umich.edu/ad-pll/

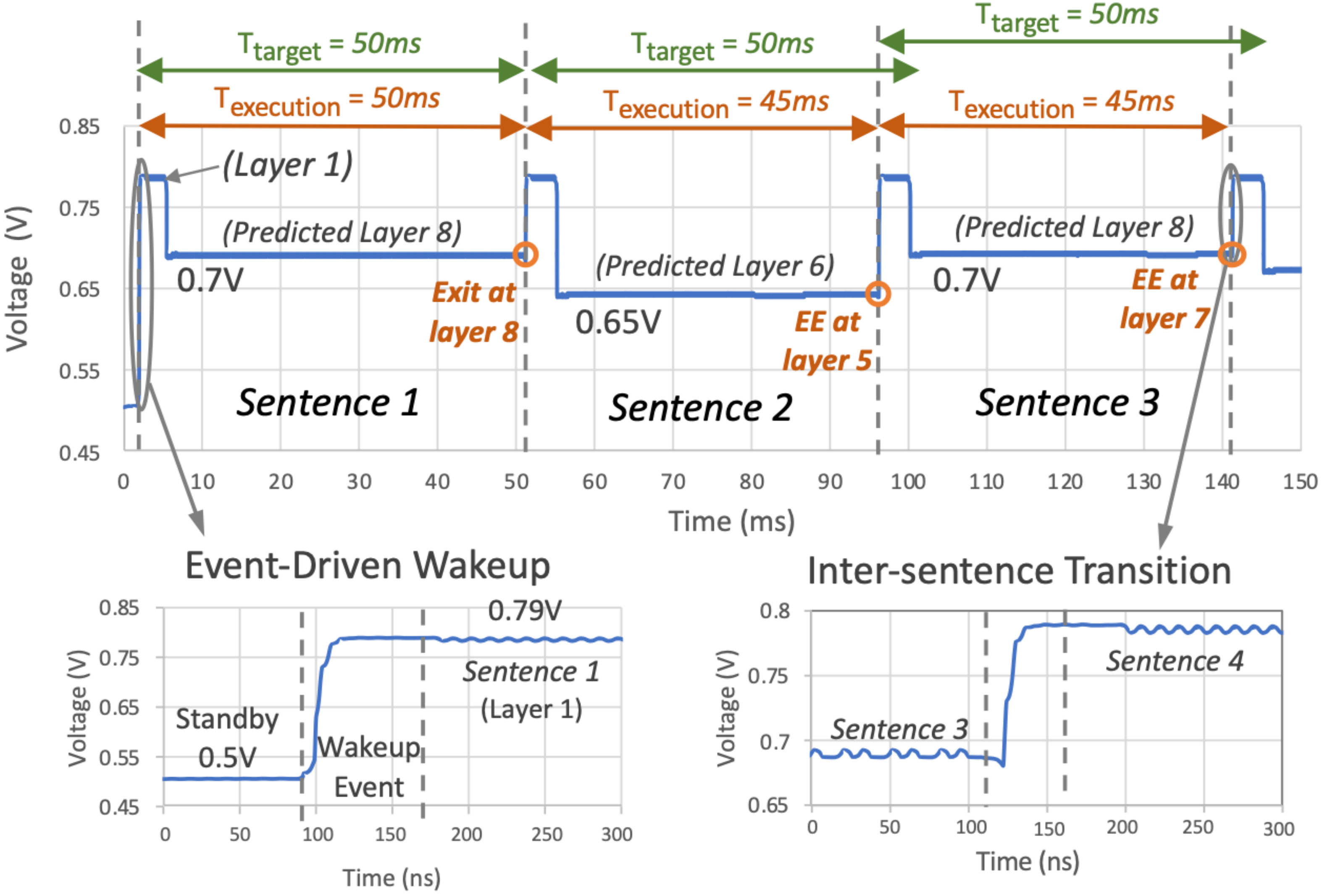
LDO RESPONSE TIME	$3.8ns/50mV$
LDO PEAK CURRENT EFFICIENCY	99.2% @ $I_{load,max}$
LDO $I_{load,max}$	$200mA$
ADPLL POWER	$2.46mW@1GHz$

Spice Simulations



With integrated LDO and ADPLL, the transition and settling time are optimized to be within 100ns

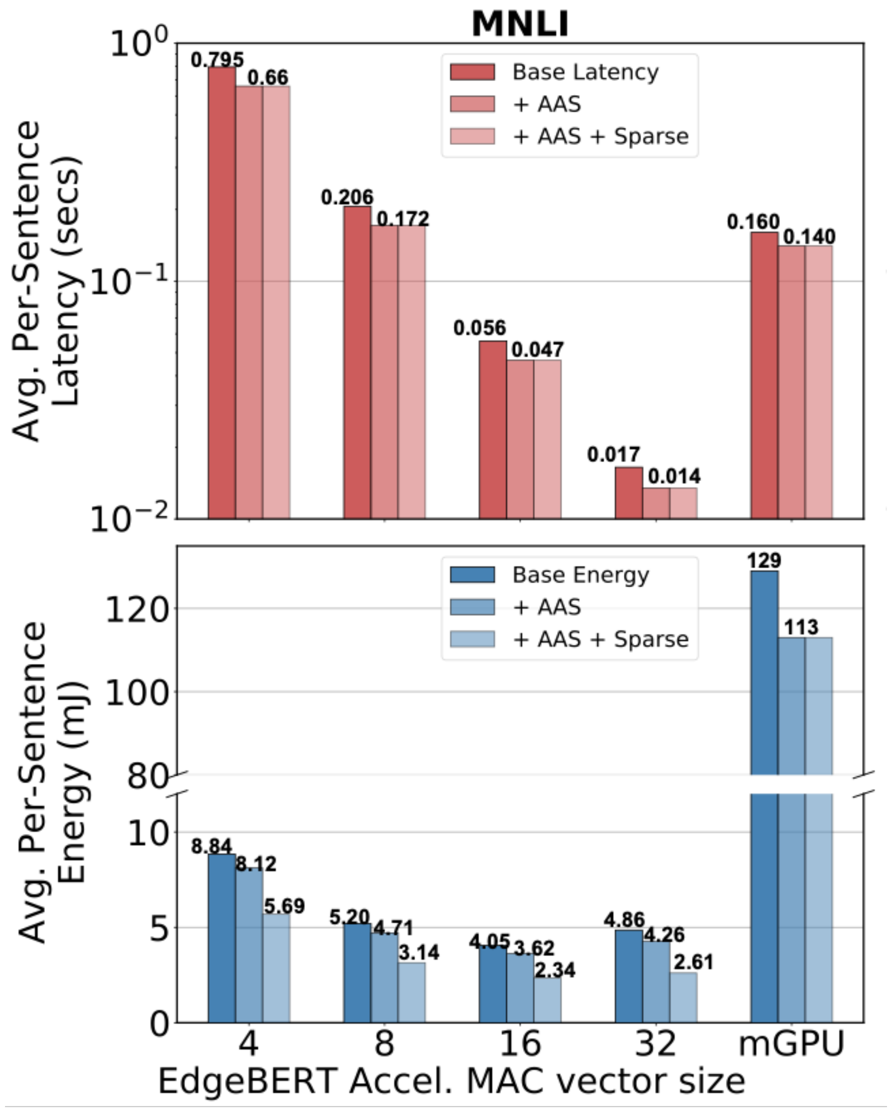
Spice Simulations



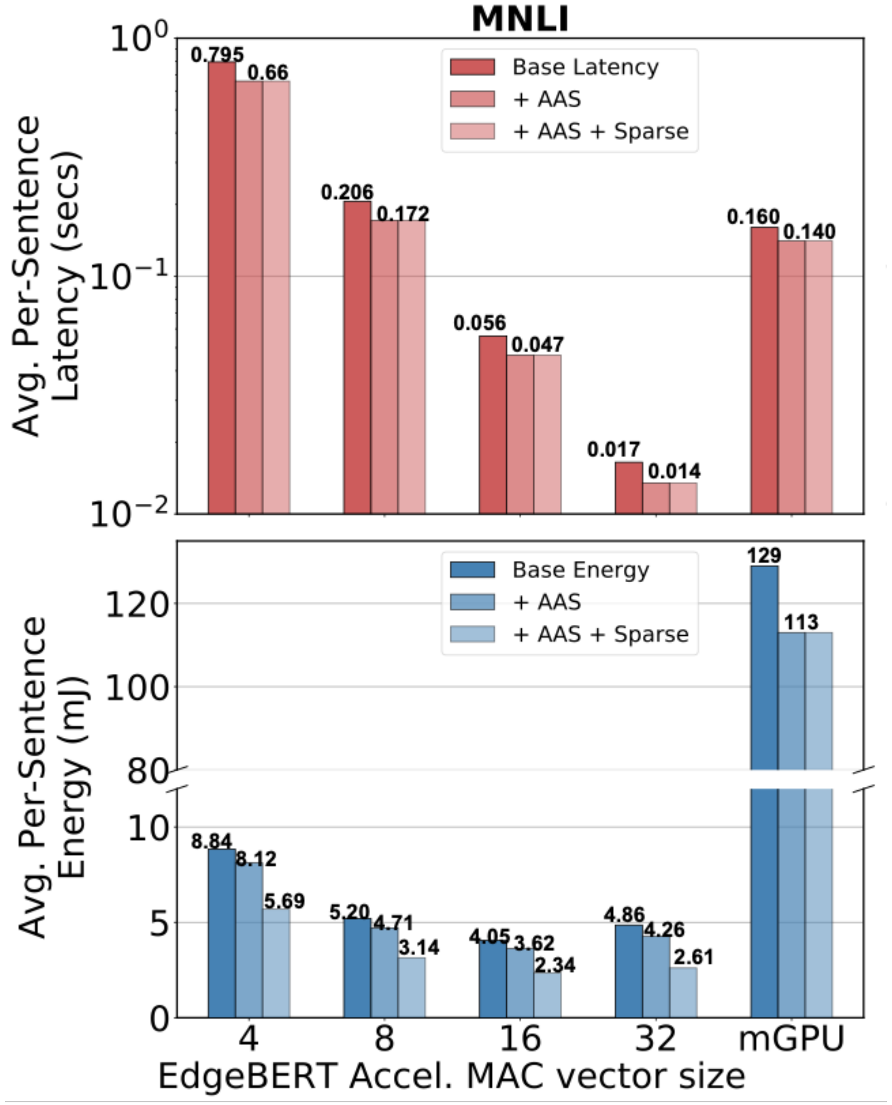
Outline

- Motivation
- EdgeBERT Optimizations
- Synergistic Evaluation
- Hardware Architecture
- Hardware Evaluation
- Conclusion

Impact of Adaptive Attention Span (AAS) and Sparse Execution

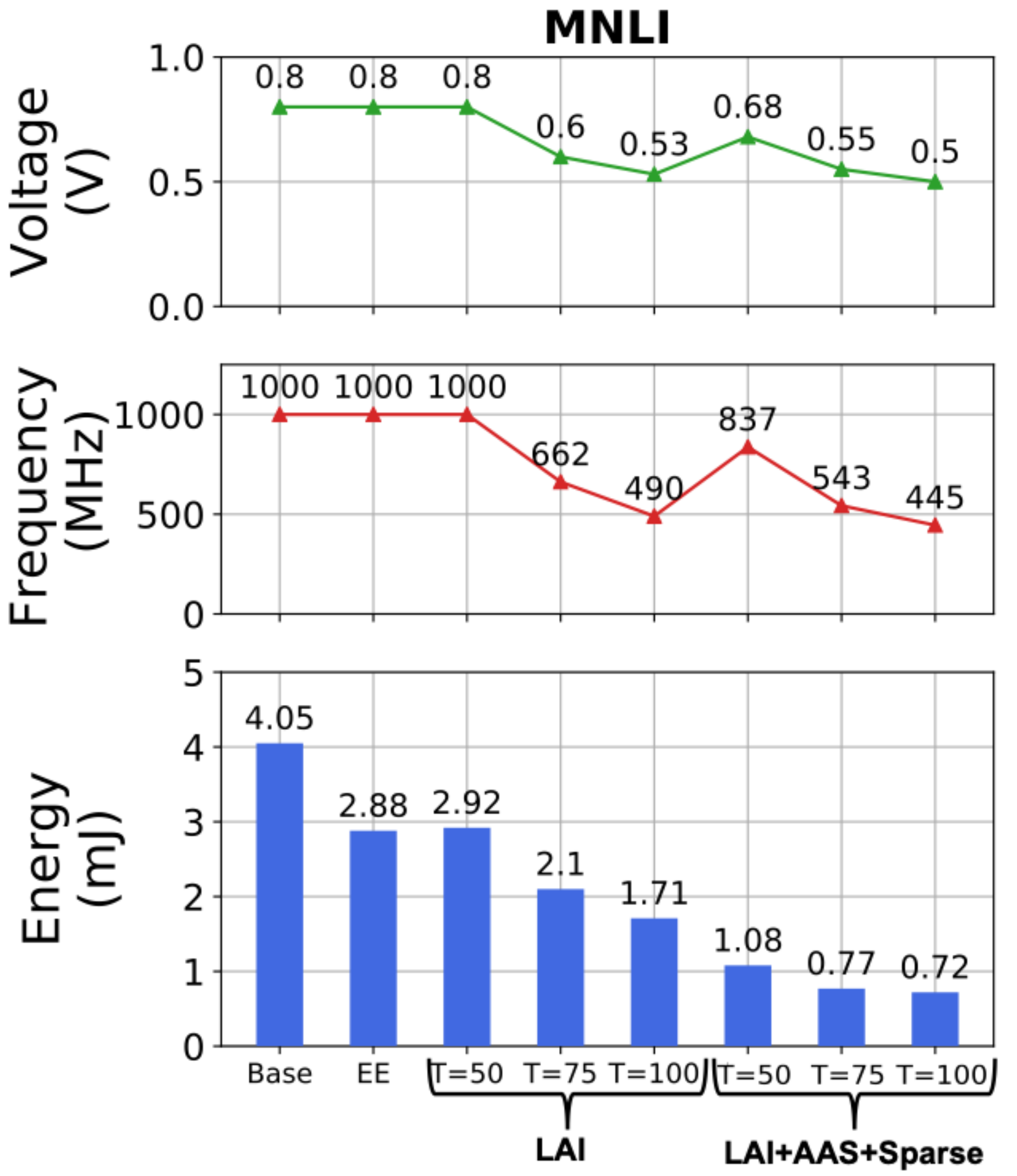


Impact of Adaptive Attention Span (AAS) and Sparse Execution

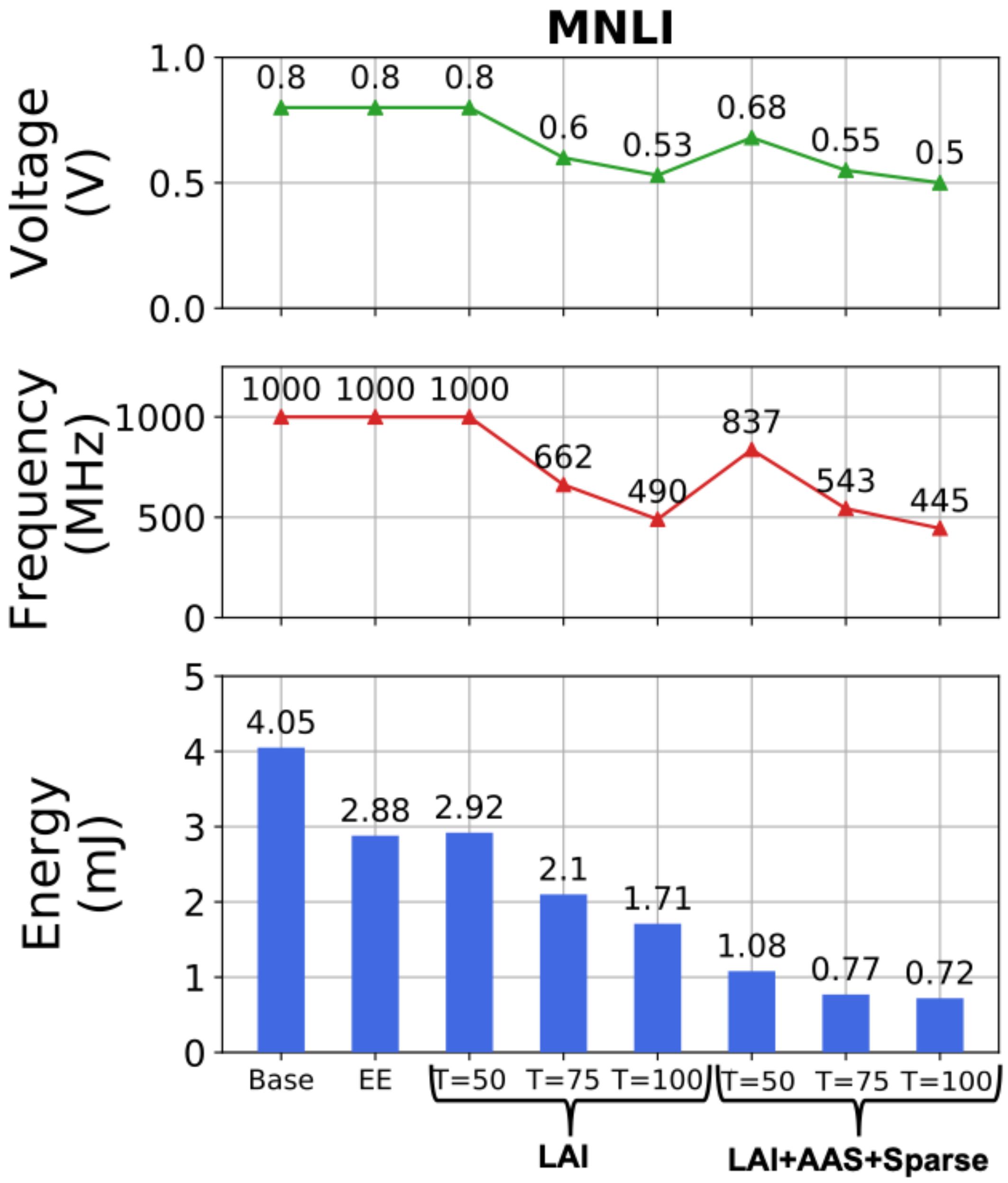


- ➢ **Latency decreases by ~3.5X as vector size doubles**
- ➢ **Sparse execution reduces energy by 1.4X**
- ➢ **MAC vector size of 16 is the most energy-efficient**

DVFS-based Latency-Aware Inference



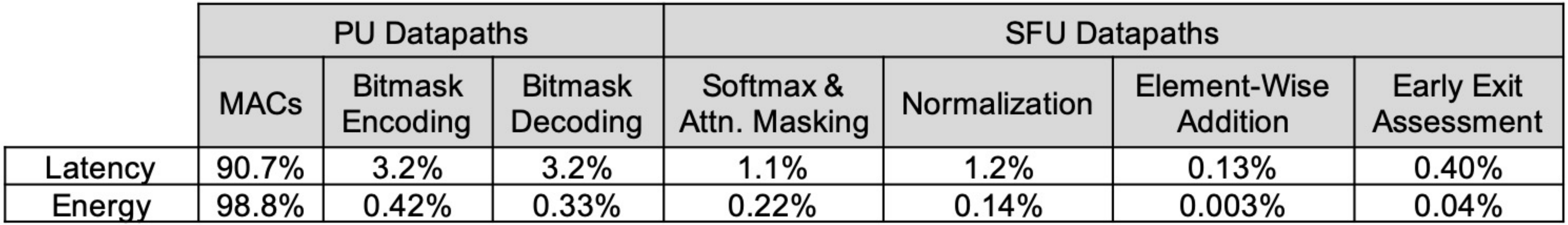
DVFS-based Latency-Aware Inference



- 7X and 2.5X energy savings compared to the non-optimized and conventional EE inference approaches, respectively

- For stricter latency targets (e.g. < 20ms), proposed scheme can be used with a larger MAC vector size (i.e. n ≥ 32)

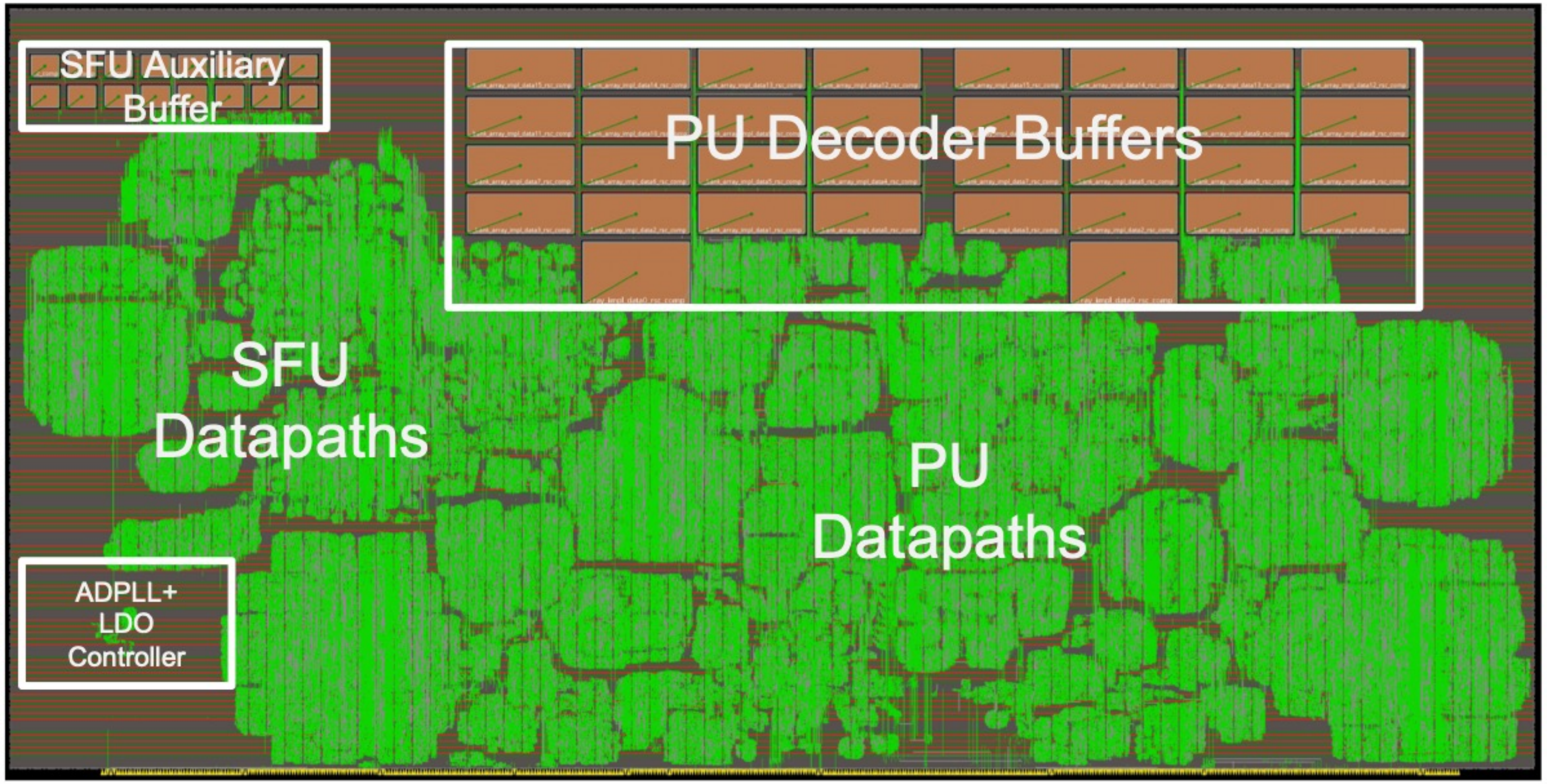
Latency and Energy Breakdown within EdgeBERT HW Units



	PU Datapaths			SFU Datapaths			
	MACs	Bitmask Encoding	Bitmask Decoding	Softmax & Attn. Masking	Normalization	Element-Wise Addition	Early Exit Assessment
Latency	90.7%	3.2%	3.2%	1.1%	1.2%	0.13%	0.40%
Energy	98.8%	0.42%	0.33%	0.22%	0.14%	0.003%	0.04%

- **Most computations are spent in the PU datapath which also accounts for the majority of the energy consumption**

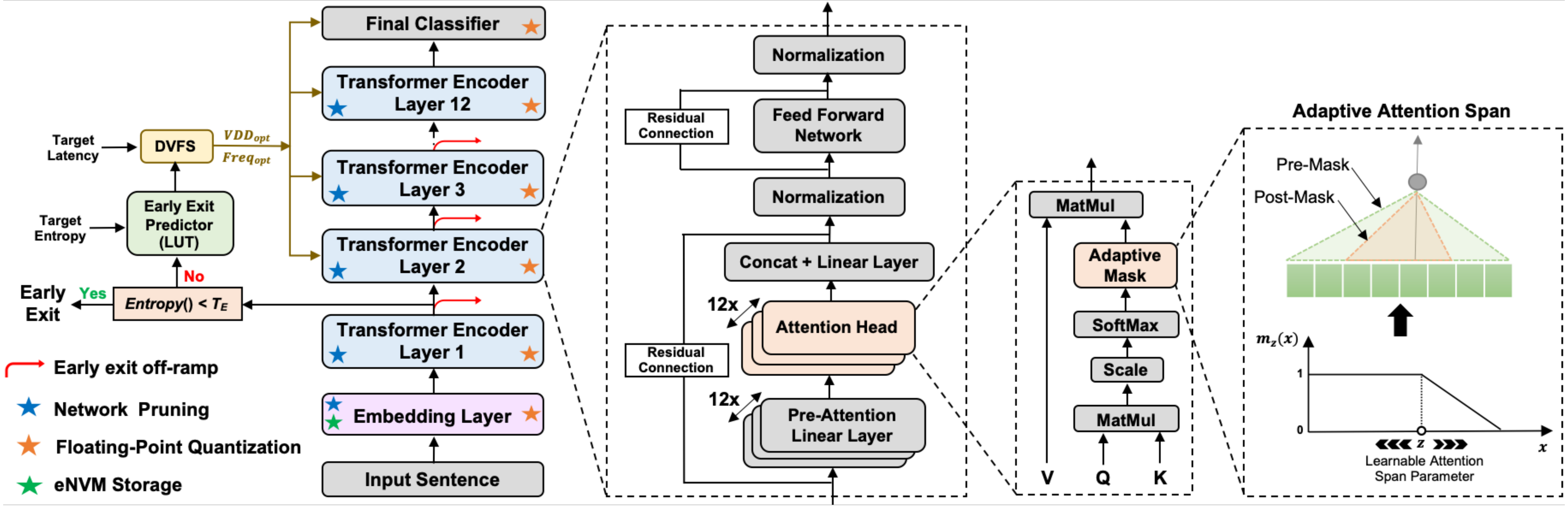
Accelerator GF12nm Summary



Blocks	Area (mm^2)	Power (mW)
PU Datapaths	0.52	36.9
SFU Datapaths	0.21	9.44
SRAM Buffers	0.50	33.6
ReRAM Buffers	0.15	3.48
ADPLL	0.01	2.46
Total	1.39	85.9

- **The 12nm EdgeBERT accelerator consumes 86mW of power and occupies 1.4mm^2 of area**

Conclusion



EdgeBERT is a cross-stack (algorithm, architecture, solid-state) set of optimizations for minimizing the energy consumption of multi-task NLP inference at a sentence granularity under the constraint of an application end-to-end latency target.

Thank You

Contact: Thierry Tambe (ttambe@g.harvard.edu)

EdgeBERT HW/SW infrastructure has been opened sourced at:

- **https://github.com/harvard-acc/EdgeBERT**
- **https://zenodo.org/record/5138730**