2nd May 2024

Large Language Model (LLM) related Hardware Optimizations

LLM in a flash: Efficient Large Language Model Inference with Limited Memory.
Exponentially Faster Language Modelling.
Inference with Reference: Lossless Acceleration of Large Language Models.

X-Former: In-Memory Acceleration of Transformers.
A Fast and Flexible FPGA-based Accelerator for Natural Language Processing Neural Networks.

Presenters

Rahul Vigneswaran K
CS23MTECH02002

Nikhil Kumar Patel
CS23MTECH11013

Course Instructors

Dr. Rajesh Kedia
Dr. Shirshendu Das

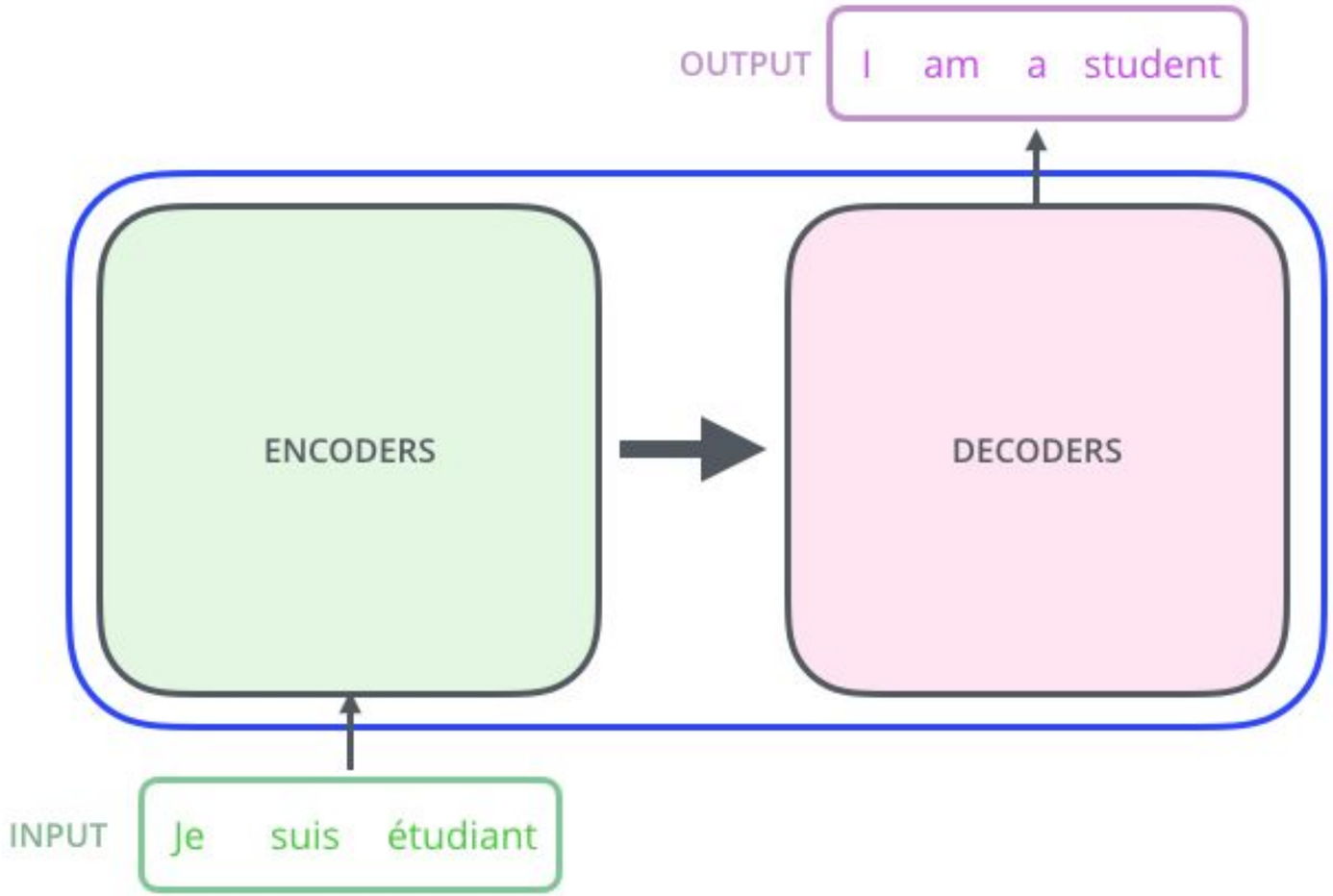
*Presented as part of the coursework for Hardware Architecture for Deep Learning (CS6490)

Table Of Contents

Transformer 101

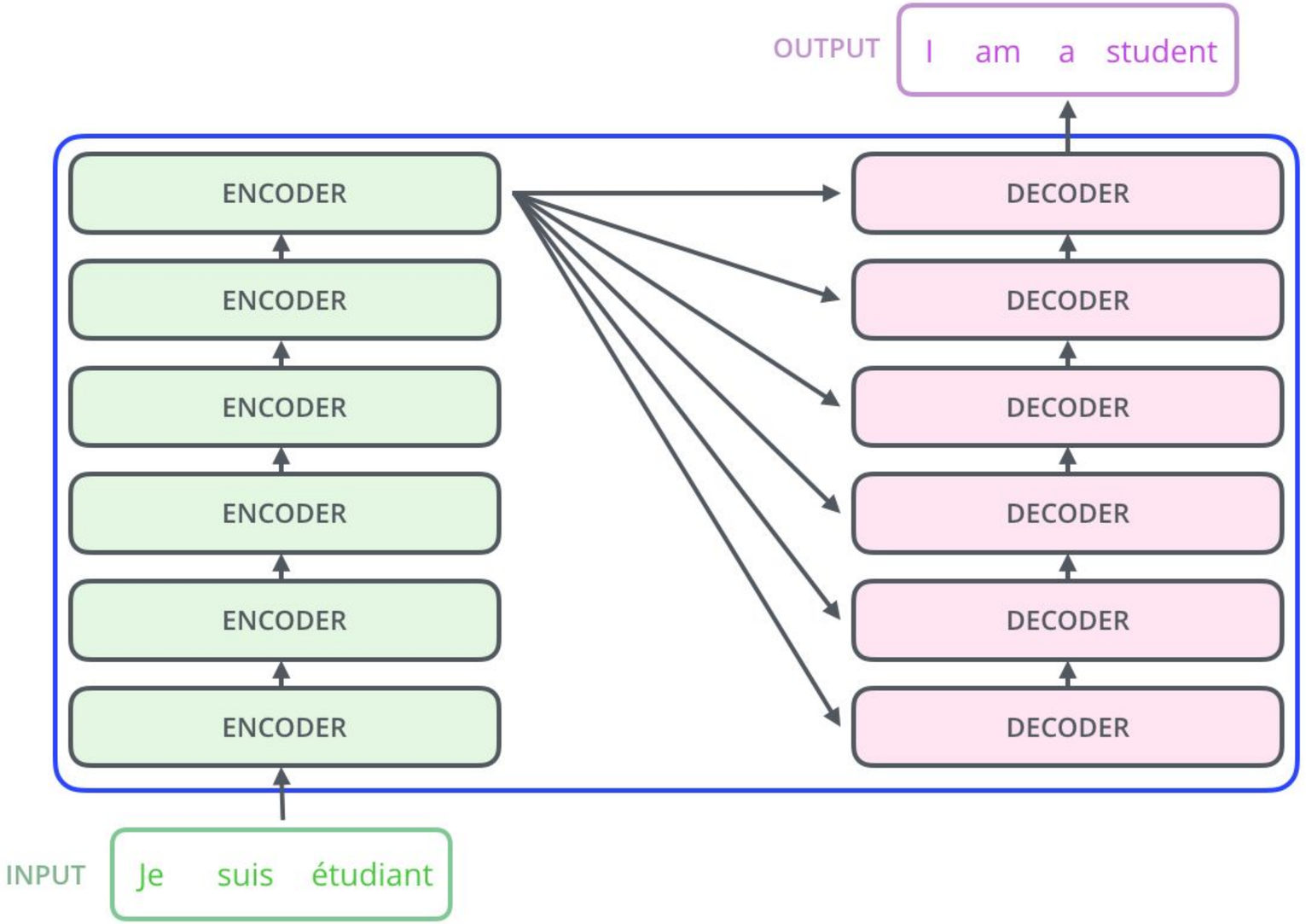
Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

Transformer 101



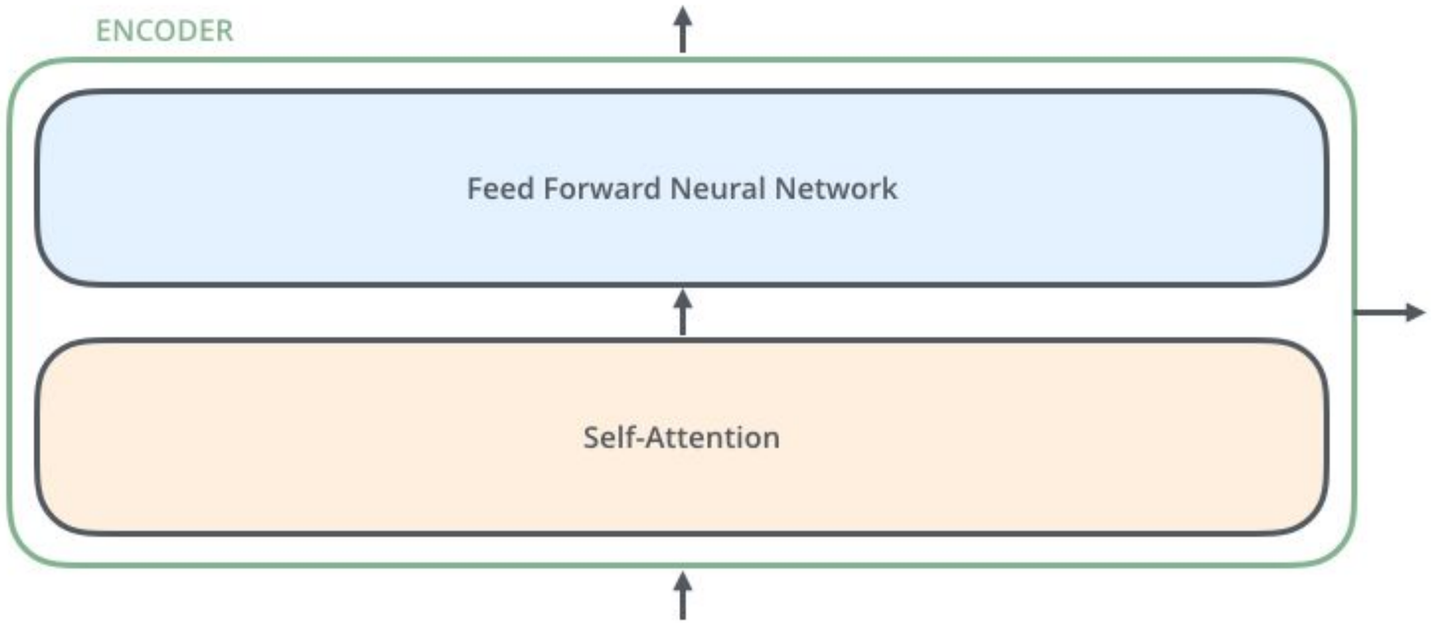
Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

Transformer 101



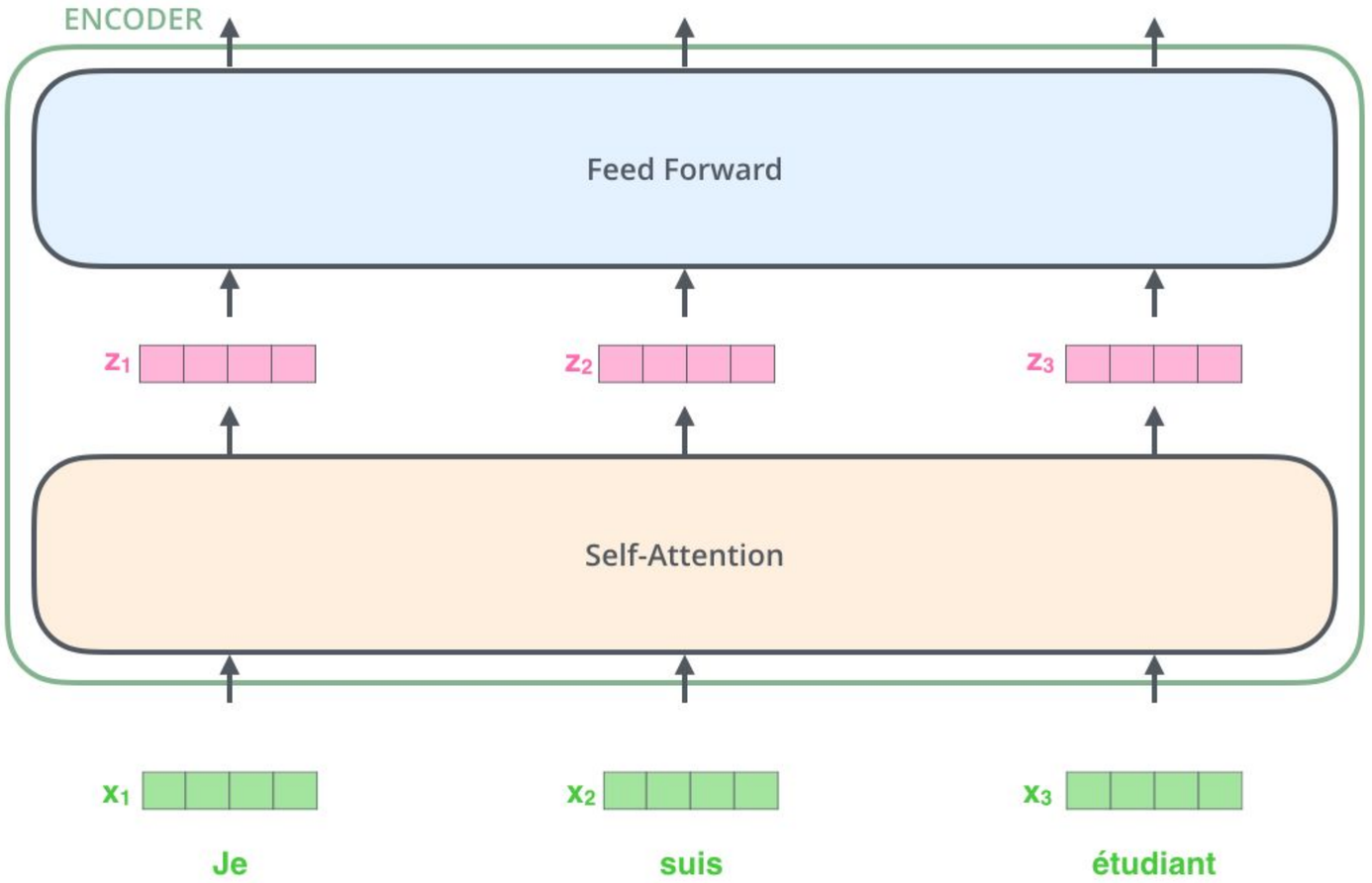
Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

Transformer 101



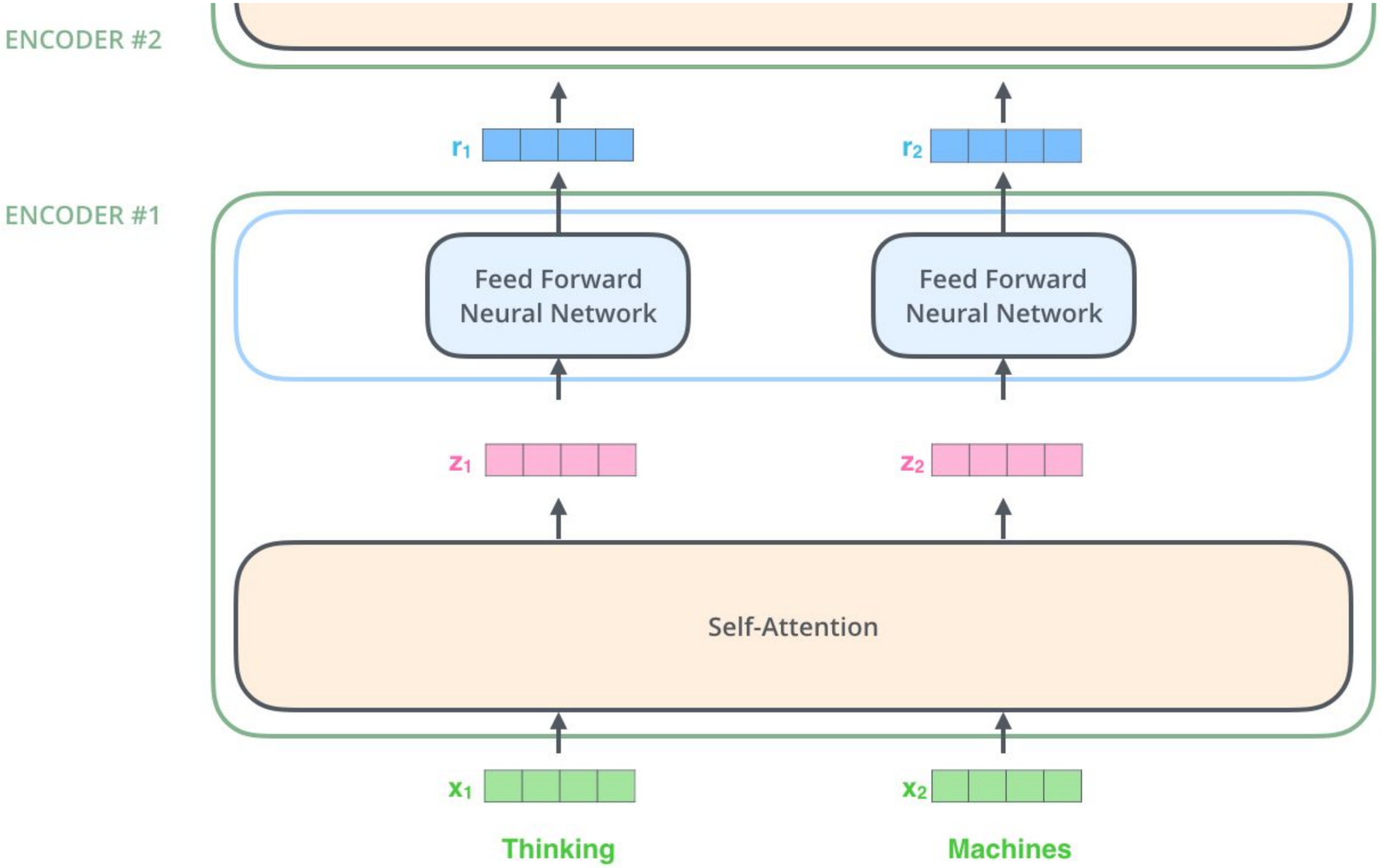
Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

Transformer 101



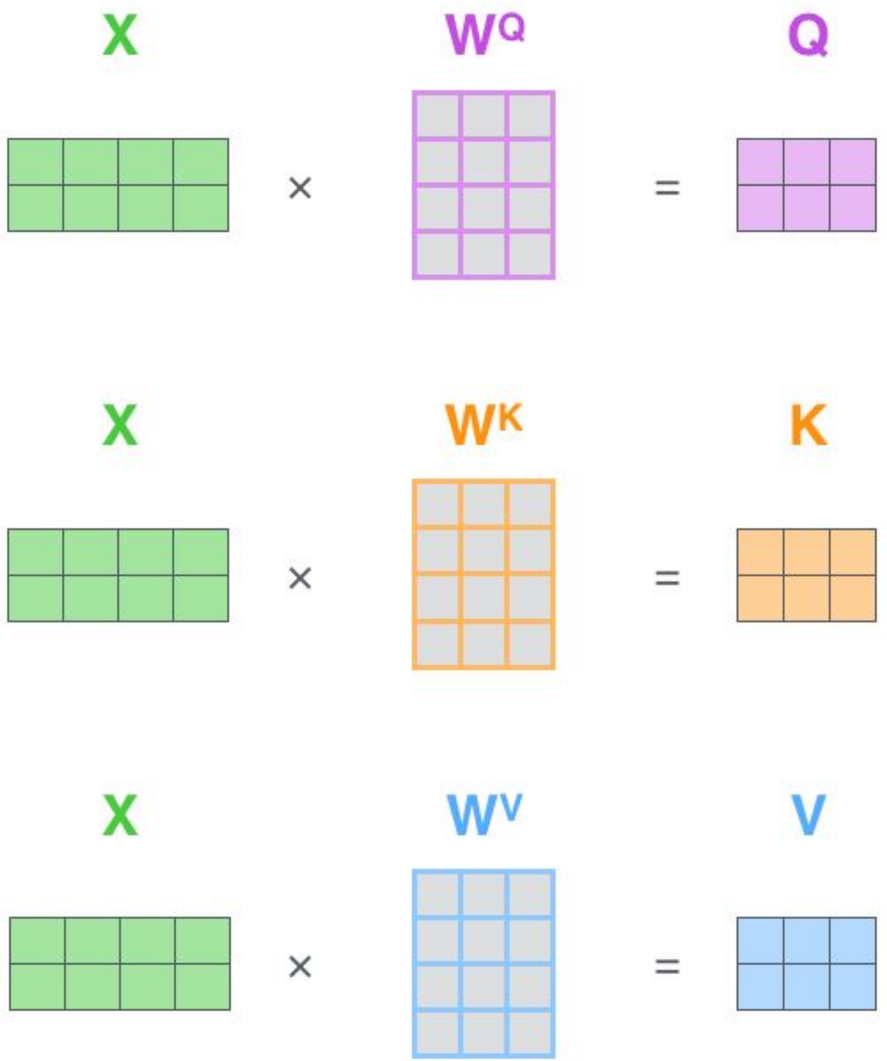
Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

Transformer 101



Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

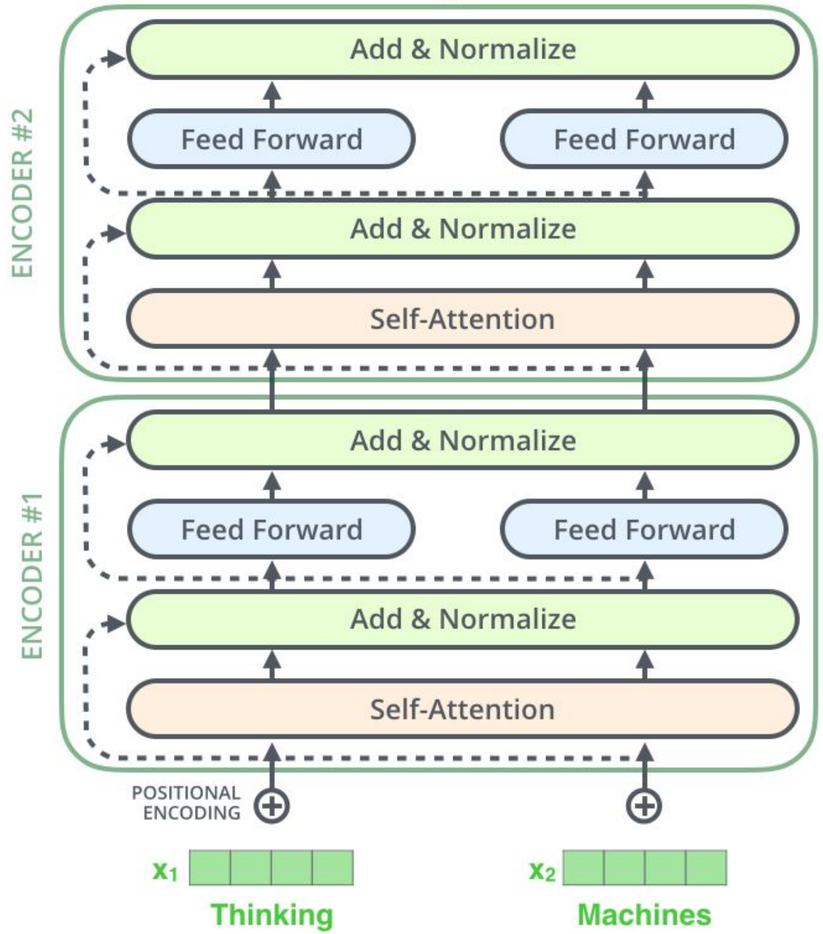
Transformer 101



$$\text{softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) V = Z$$

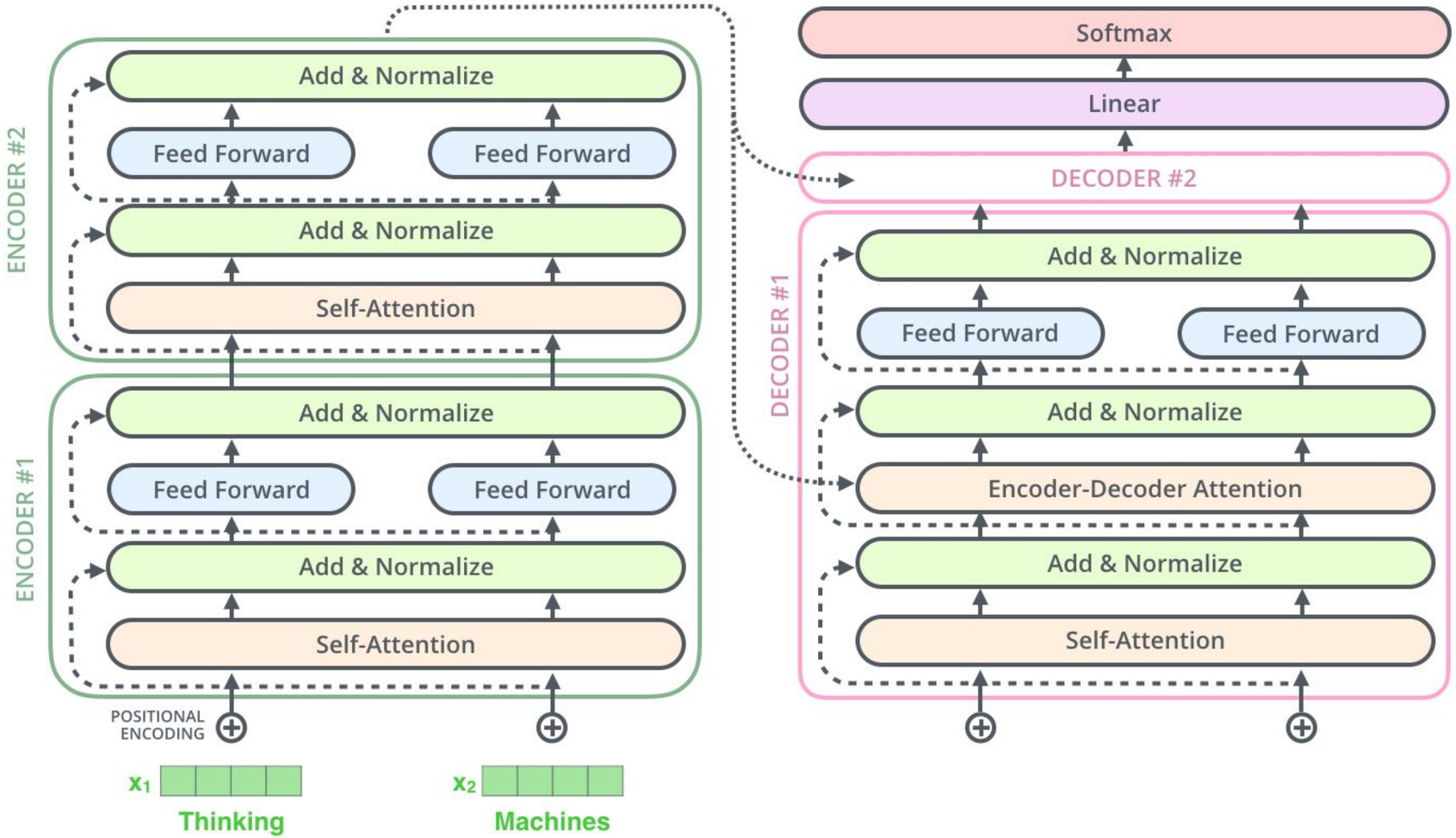
Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

Transformer 101



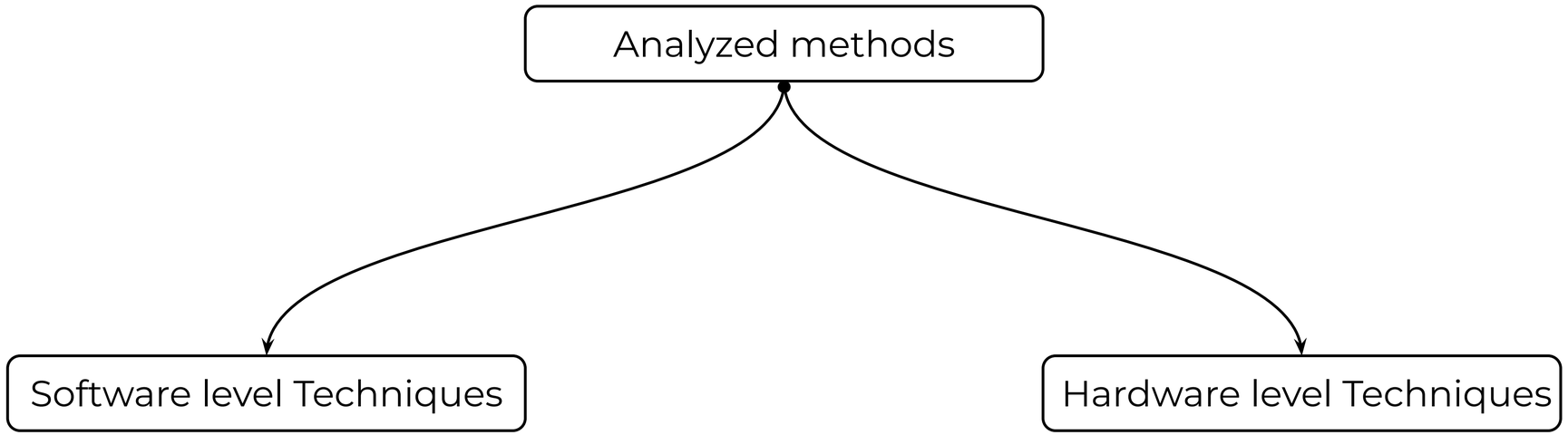
Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

Transformer 101

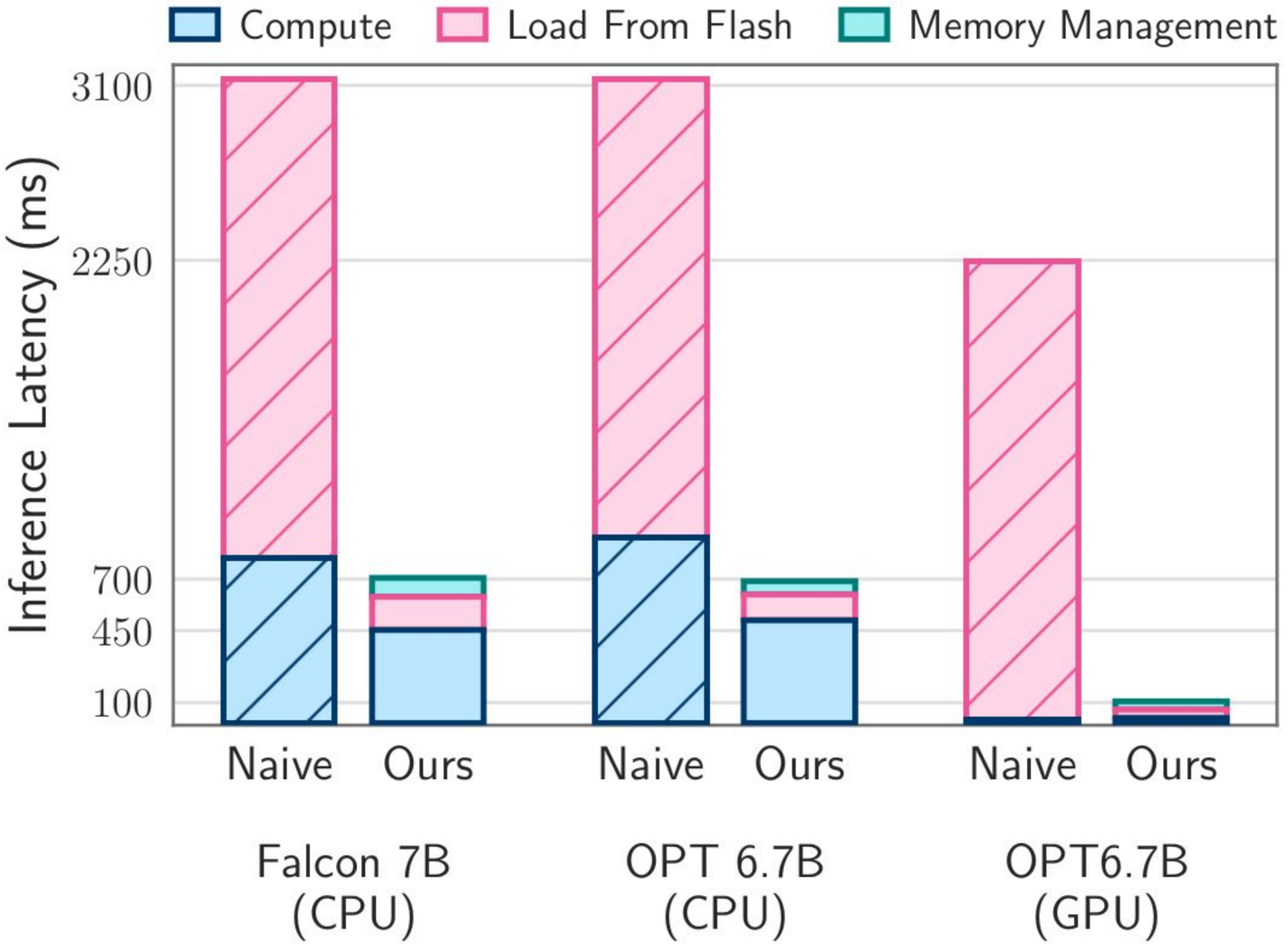


Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

Overview

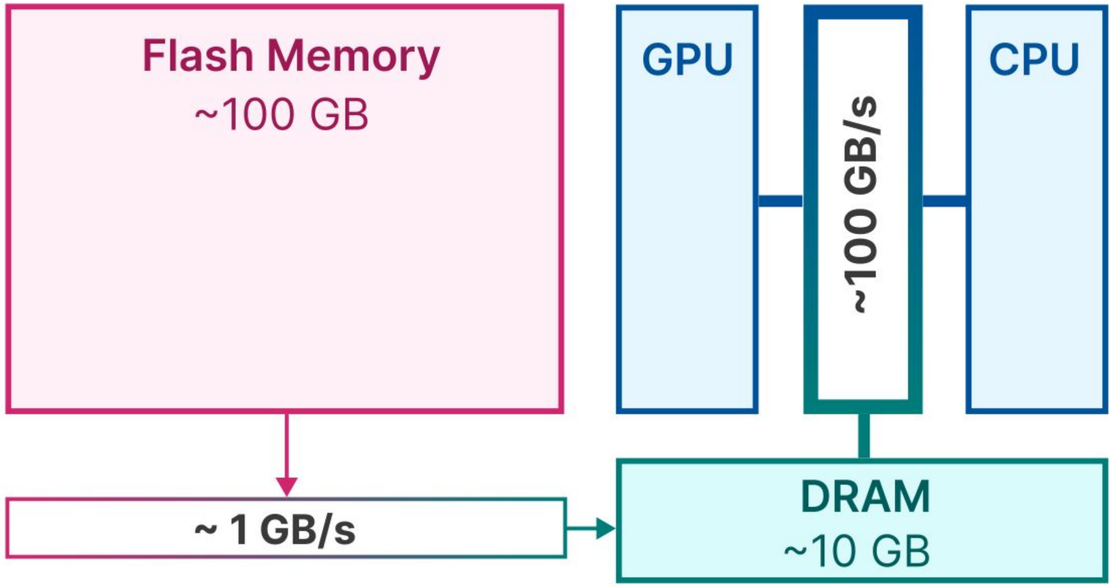

Analyzed methods
Software level Techniques
Hardware level Techniques

LLM in a flash: Efficient Large Language Model Inference with Limited Memory



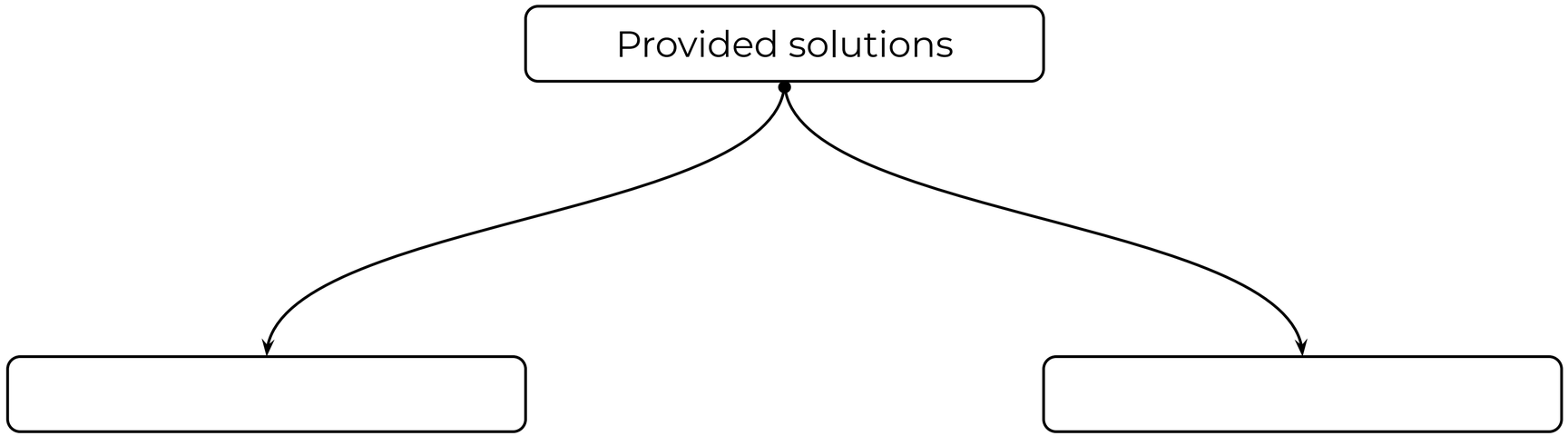
Alizadeh-Vahid, Keivan et al. LLM in a flash: Efficient Large Language Model Inference with Limited Memory. ArXiv 2023.

LLM in a flash: Efficient Large Language Model Inference with Limited Memory

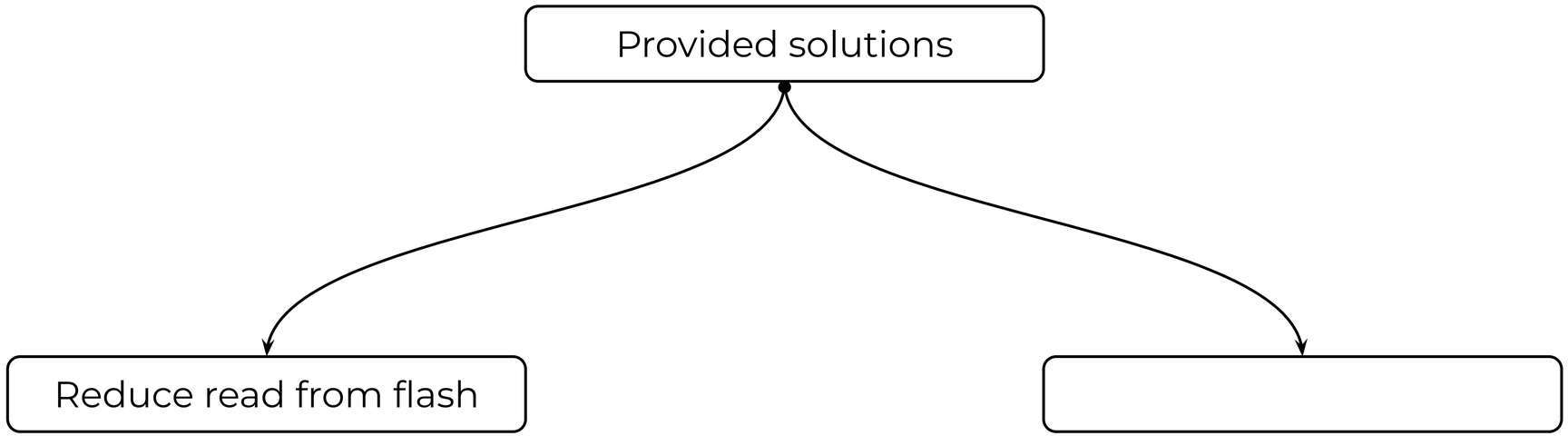


Alizadeh-Vahid, Keivan et al. LLM in a flash: Efficient Large Language Model Inference with Limited Memory. ArXiv 2023.

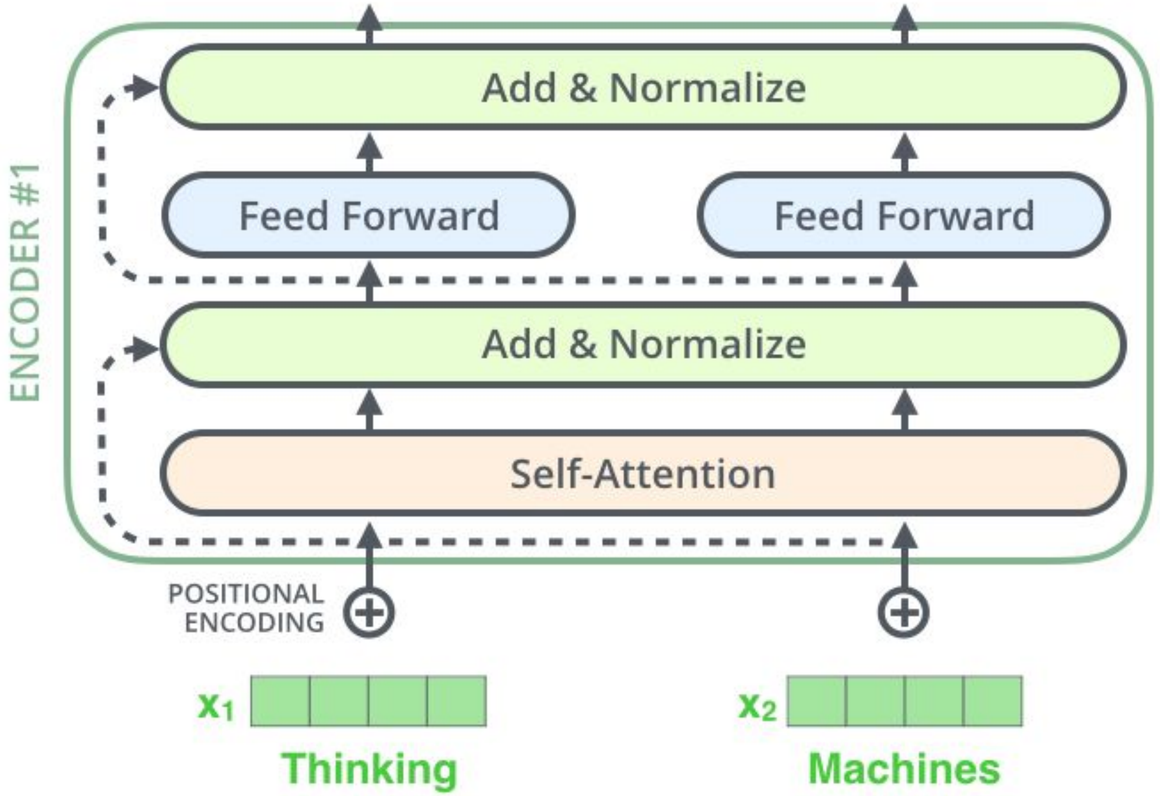
LLM in a flash: Efficient Large Language Model Inference with Limited Memory

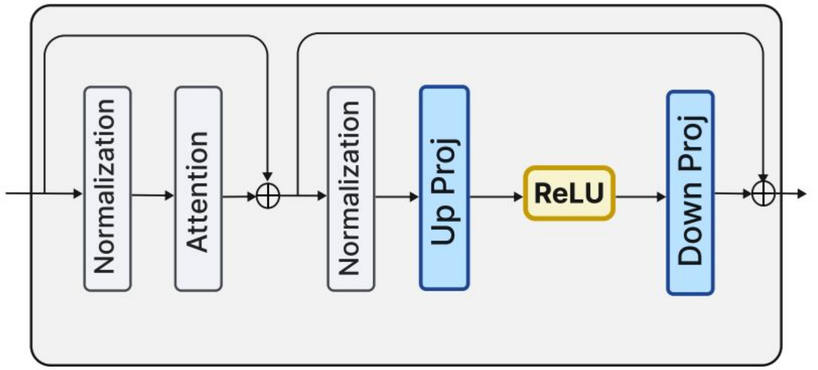


LLM in a flash: Efficient Large Language Model Inference with Limited Memory



LLM in a flash: Efficient Large Language Model Inference with Limited Memory

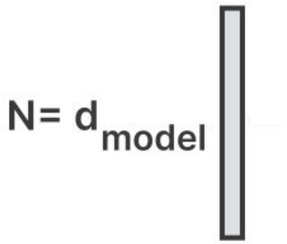


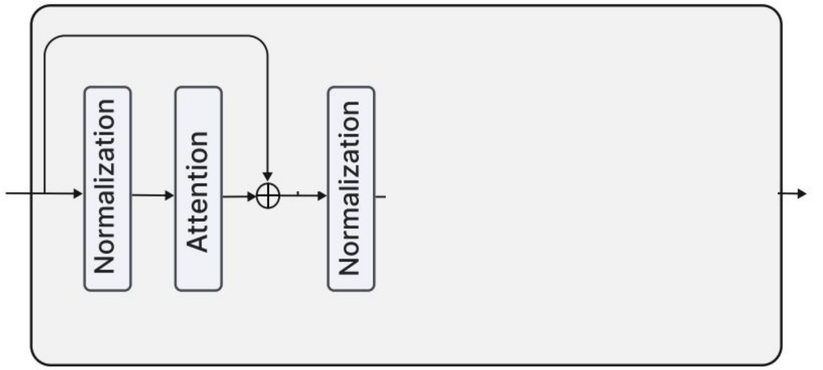


Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

Mirzadeh, Iman, et al. Relu strikes back: Exploiting activation sparsity in large language models. arXiv 2023.

Flash : Reduce read from flash

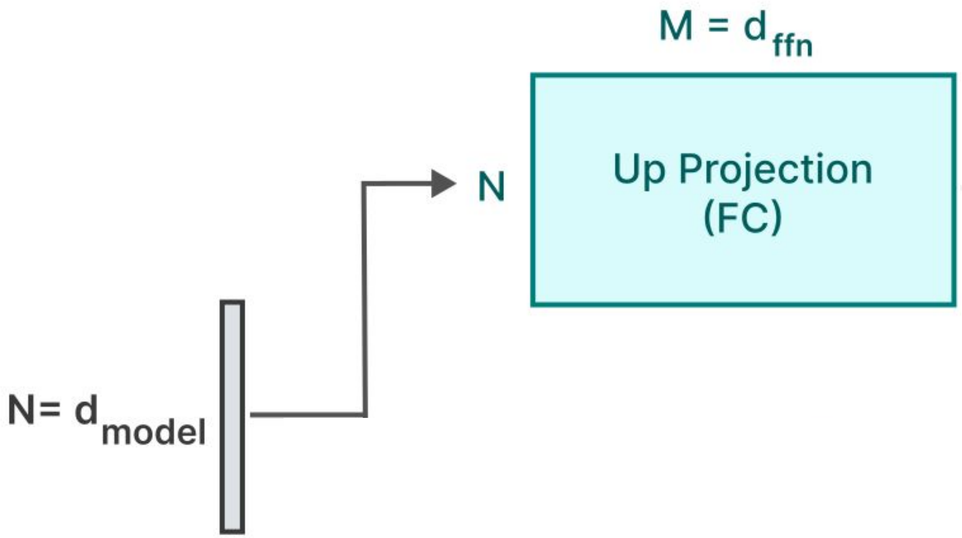


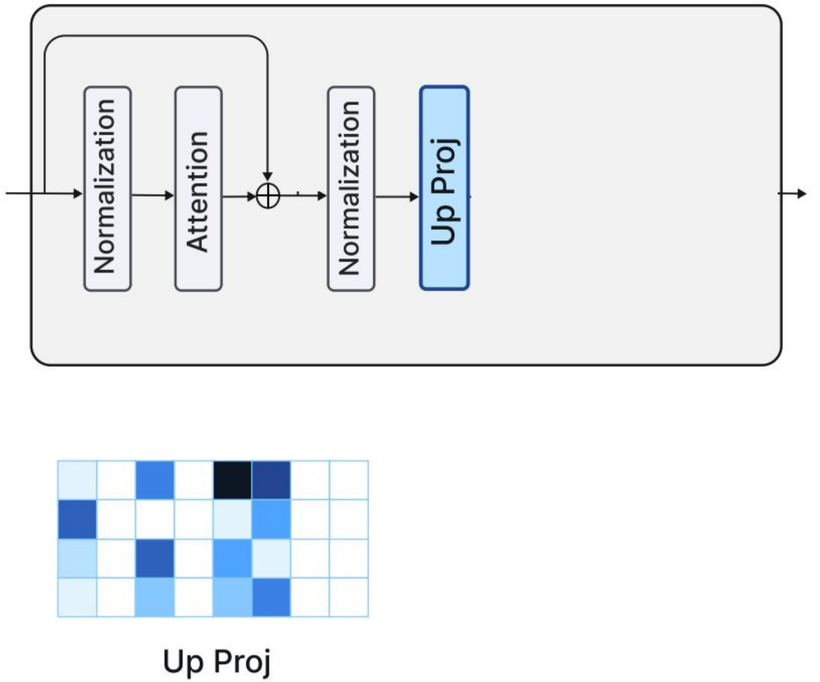


Alizadeh-Vahid, Keivan et al. LLM in a flash: Efficient Large Language Model Inference with Limited Memory. ArXiv 2023.

Mirzadeh, Iman, et al. Relu strikes back: Exploiting activation sparsity in large language models. arXiv 2023.

Flash : Reduce read from flash

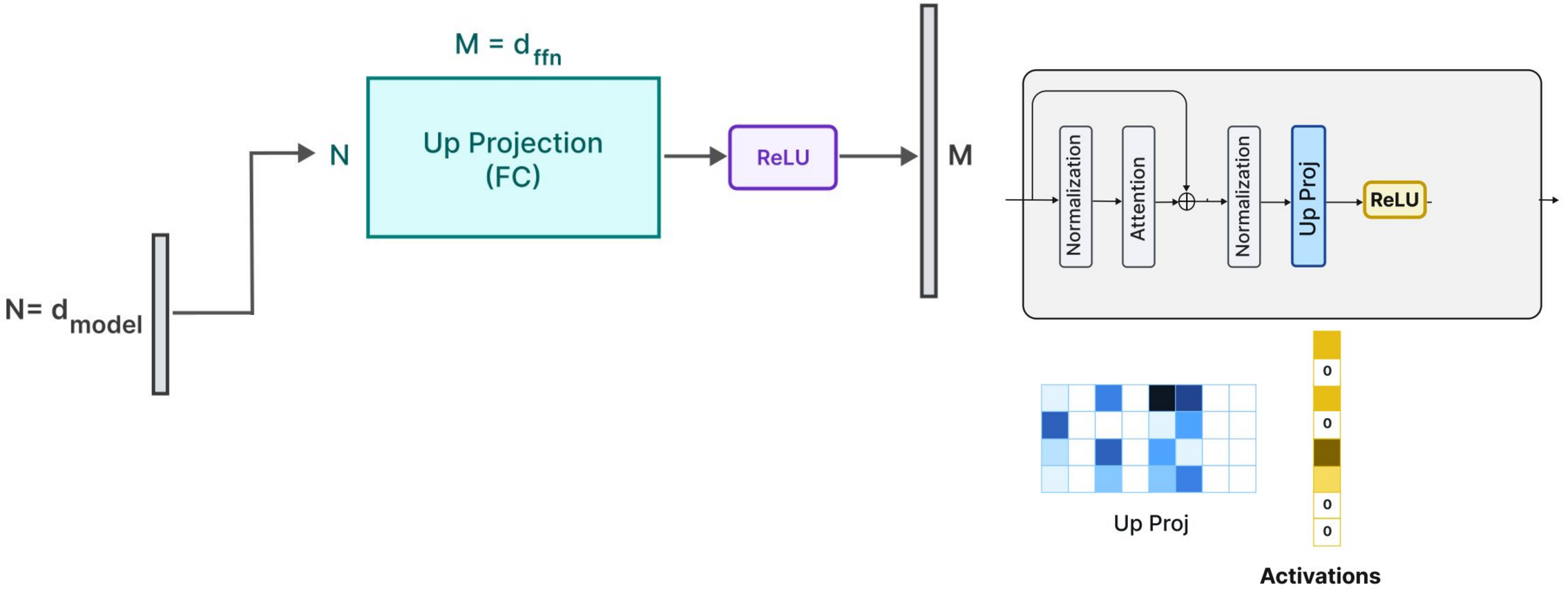




Alizadeh-Vahid, Keivan et al. LLM in a flash: Efficient Large Language Model Inference with Limited Memory. ArXiv 2023.

Mirzadeh, Iman, et al. Relu strikes back: Exploiting activation sparsity in large language models. arXiv 2023.

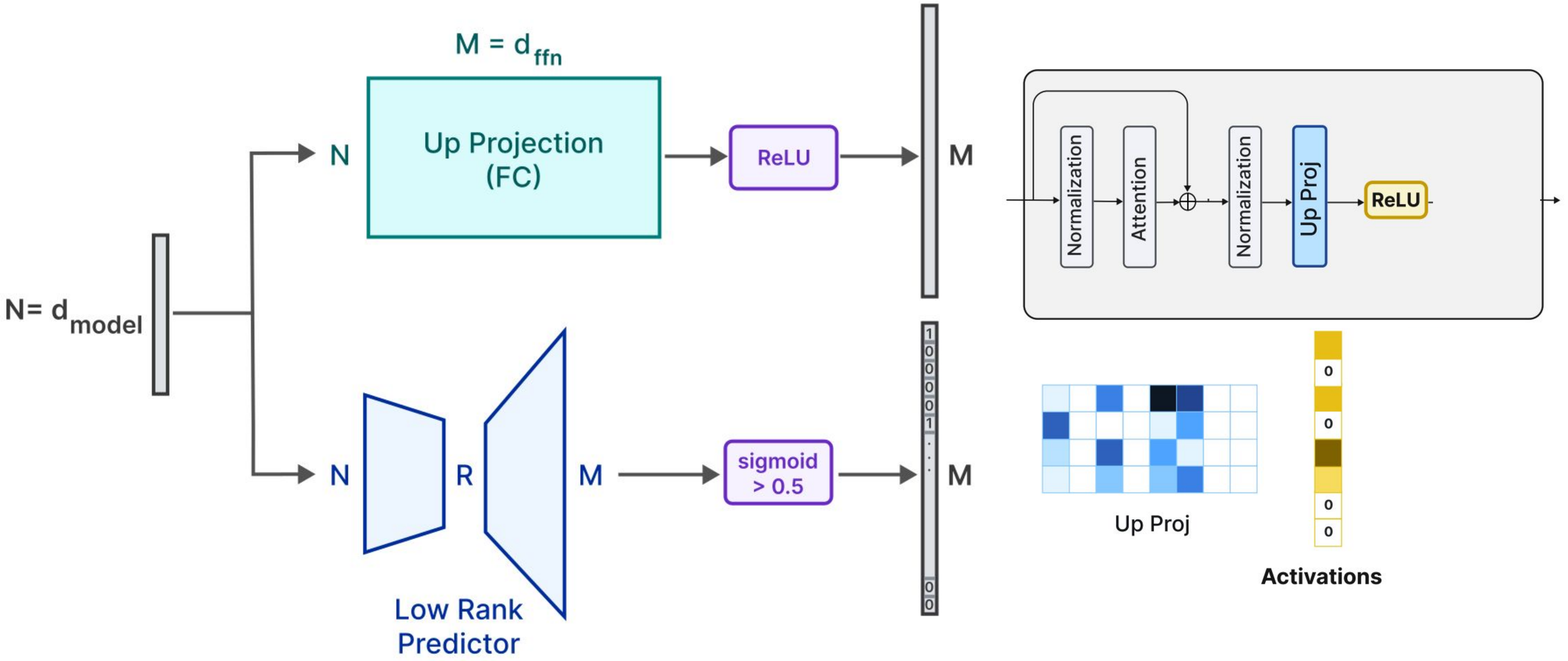
Flash : Reduce read from flash



Alizadeh-Vahid, Keivan et al. LLM in a flash: Efficient Large Language Model Inference with Limited Memory. ArXiv 2023.

Mirzadeh, Iman, et al. Relu strikes back: Exploiting activation sparsity in large language models. arXiv 2023.

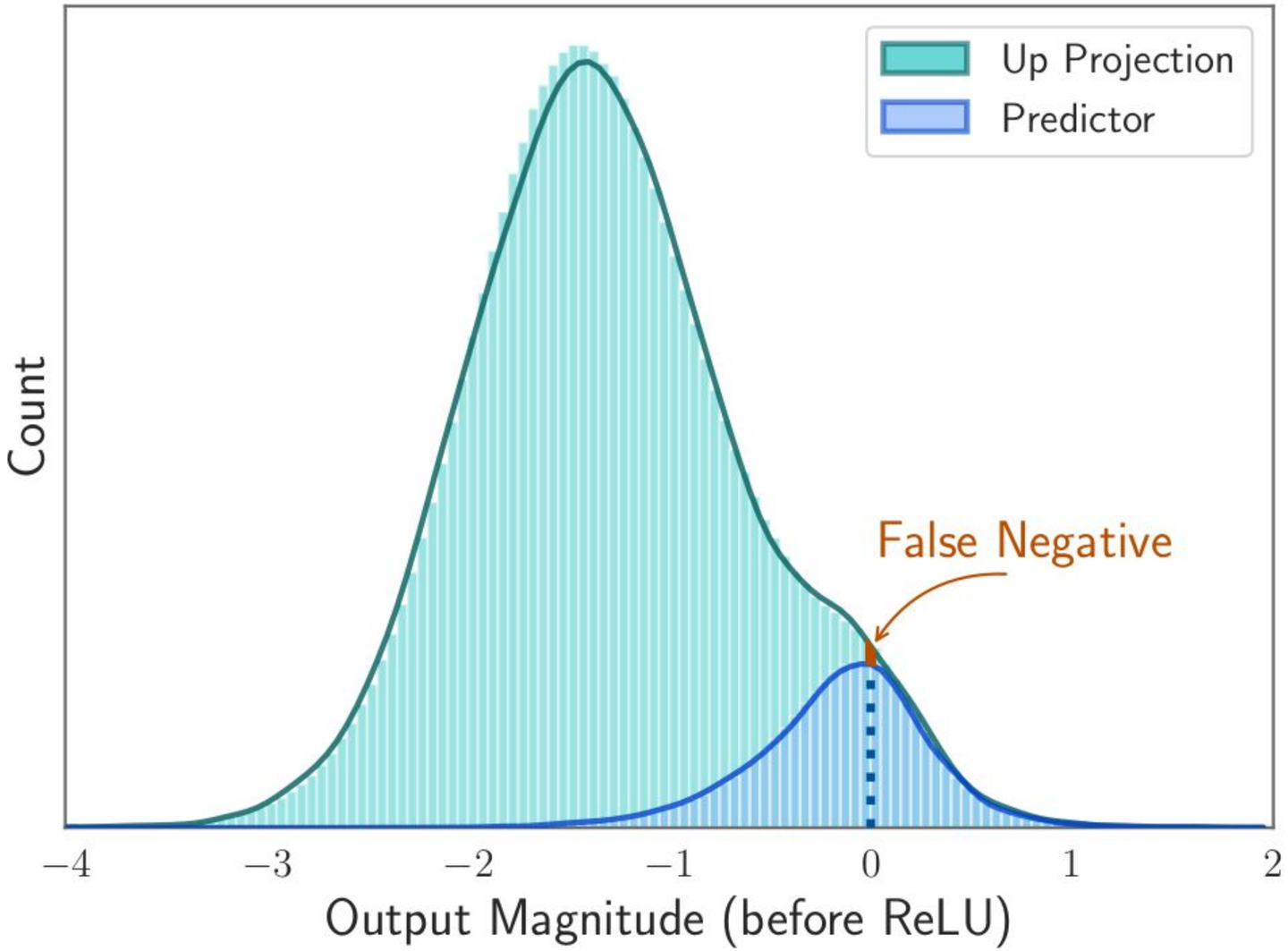
Flash : Reduce read from flash



Alizadeh-Vahid, Keivan et al. LLM in a flash: Efficient Large Language Model Inference with Limited Memory. ArXiv 2023.

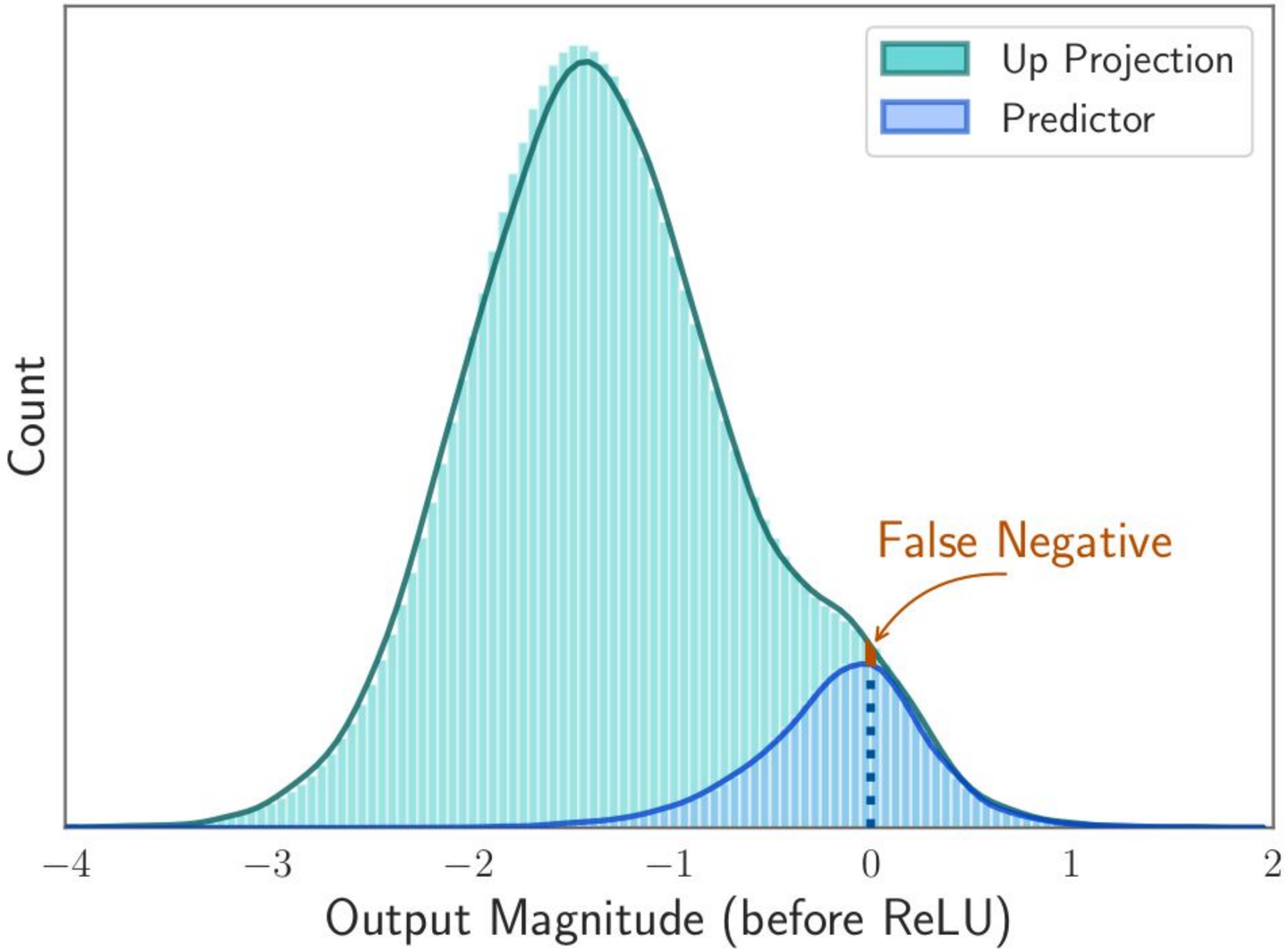
Mirzadeh, Iman, et al. Relu strikes back: Exploiting activation sparsity in large language models. arXiv 2023.

Flash : Reduce read from flash



Alizadeh-Vahid, Keivan et al. LLM in a flash: Efficient Large Language Model Inference with Limited Memory. ArXiv 2023.

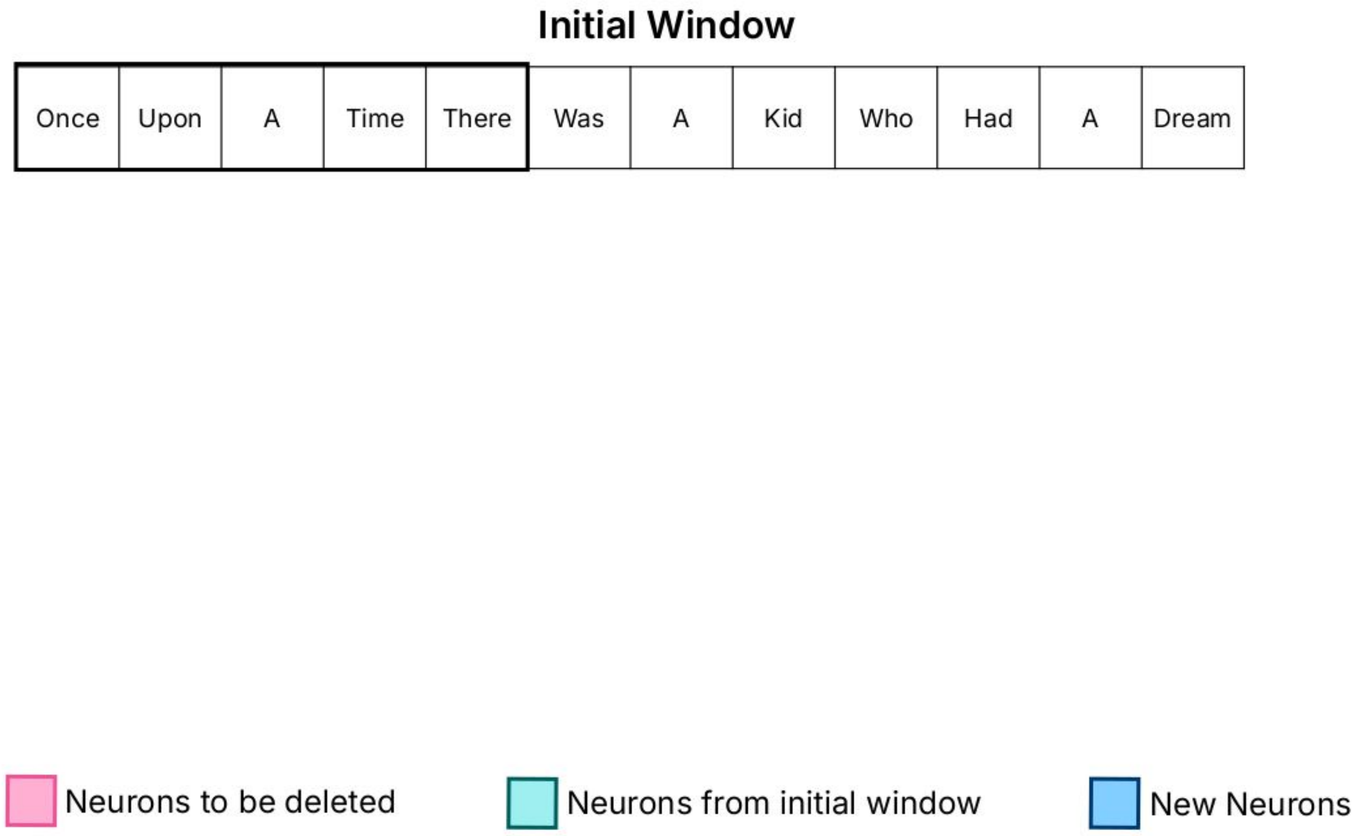
Flash : Reduce read from flash



Zero-Shot Task	OPT 6.7B	with Predictor
Arc Easy	66.1	66.2
Arc Challenge	30.6	30.6
HellaSwag	50.3	49.8

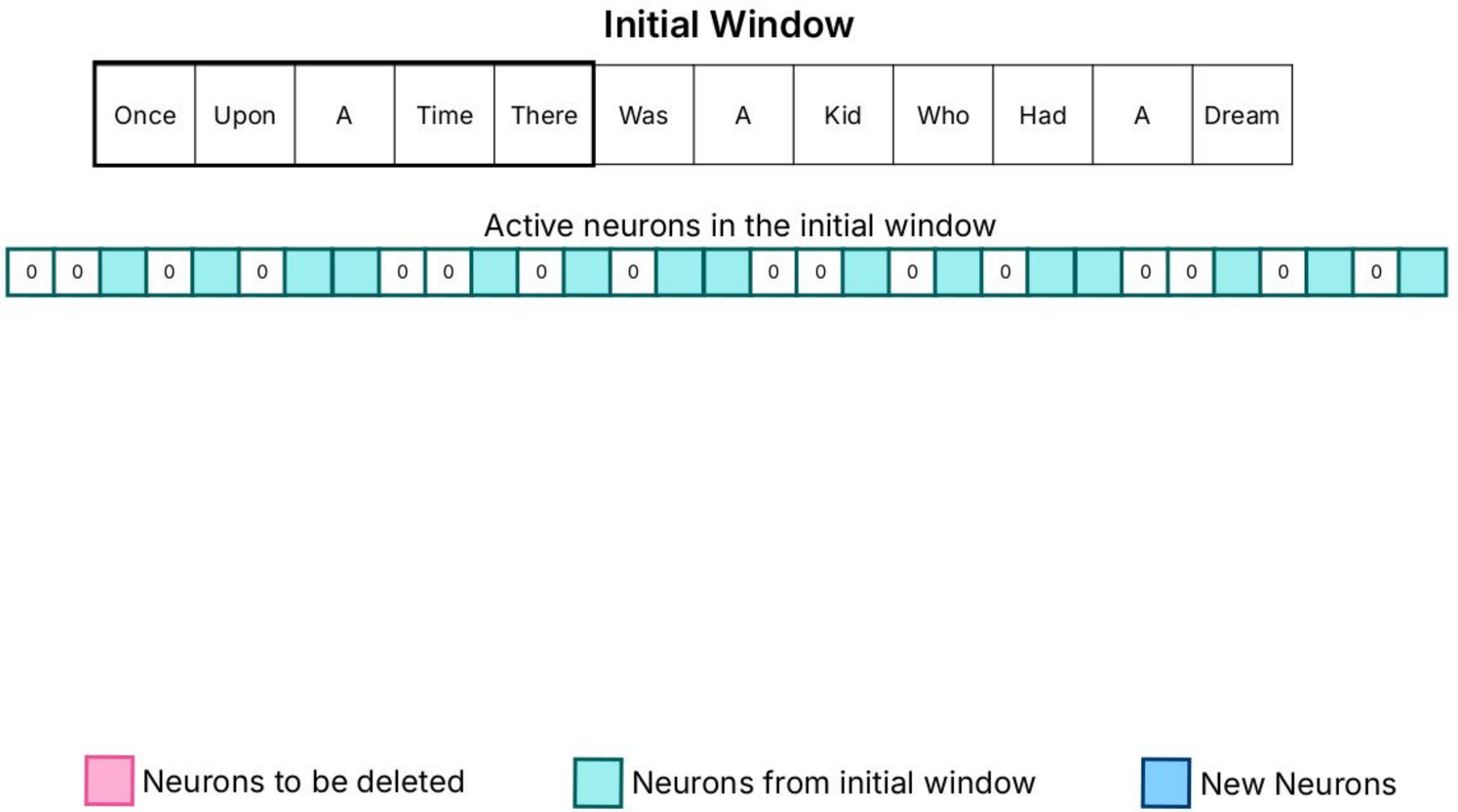
Alizadeh-Vahid, Keivan et al. LLM in a flash: Efficient Large Language Model Inference with Limited Memory. ArXiv 2023.

Flash : Reduce read from flash



Alizadeh-Vahid, Keivan et al. LLM in a flash: Efficient Large Language Model Inference with Limited Memory. ArXiv 2023.

Flash : Reduce read from flash

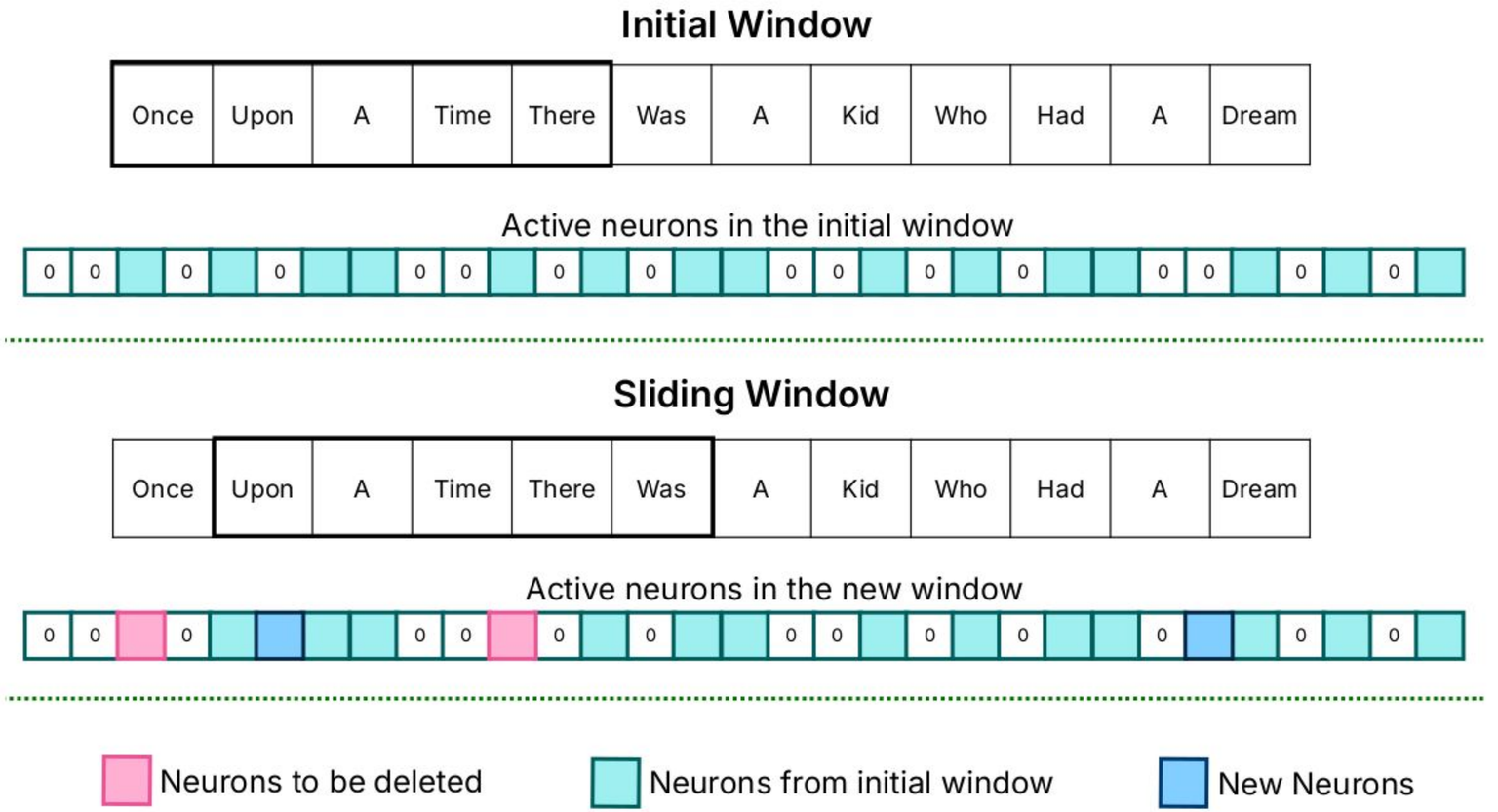


Alizadeh-Vahid, Keivan et al. LLM in a flash: Efficient Large Language Model Inference with Limited Memory. ArXiv 2023.

Flash : Reduce read from flash

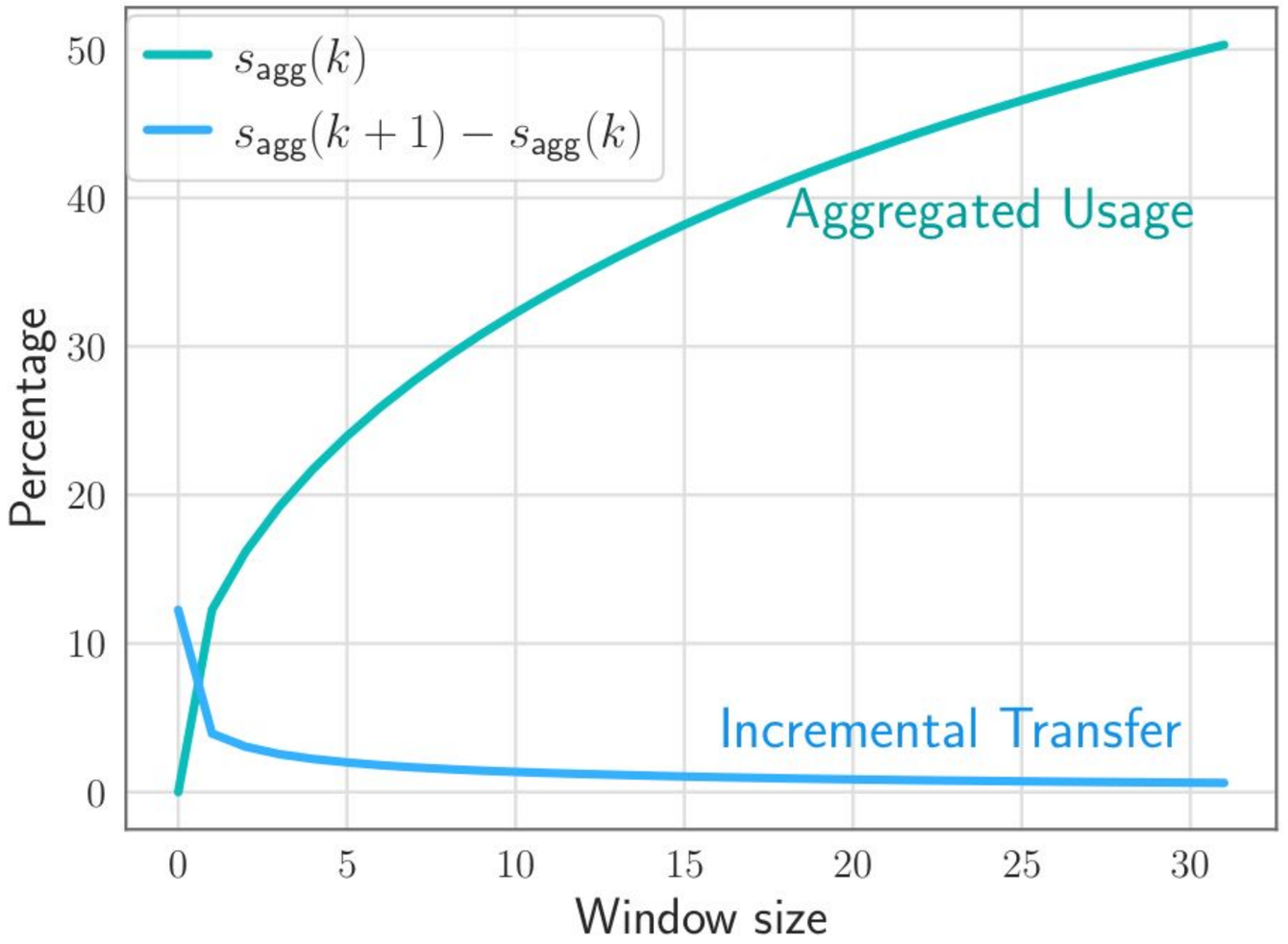
Alizadeh-Vahid, Keivan et al. LLM in a flash: Efficient Large Language Model Inference with Limited Memory. ArXiv 2023.

Flash : Reduce read from flash



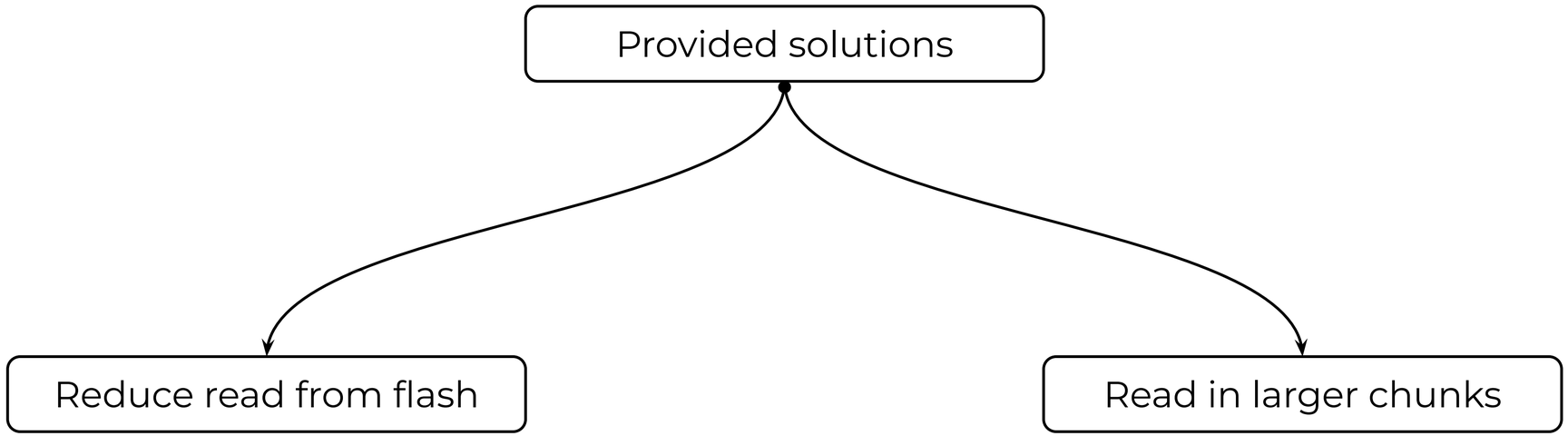
Alizadeh-Vahid, Keivan et al. LLM in a flash: Efficient Large Language Model Inference with Limited Memory. ArXiv 2023.

Flash : Reduce read from flash

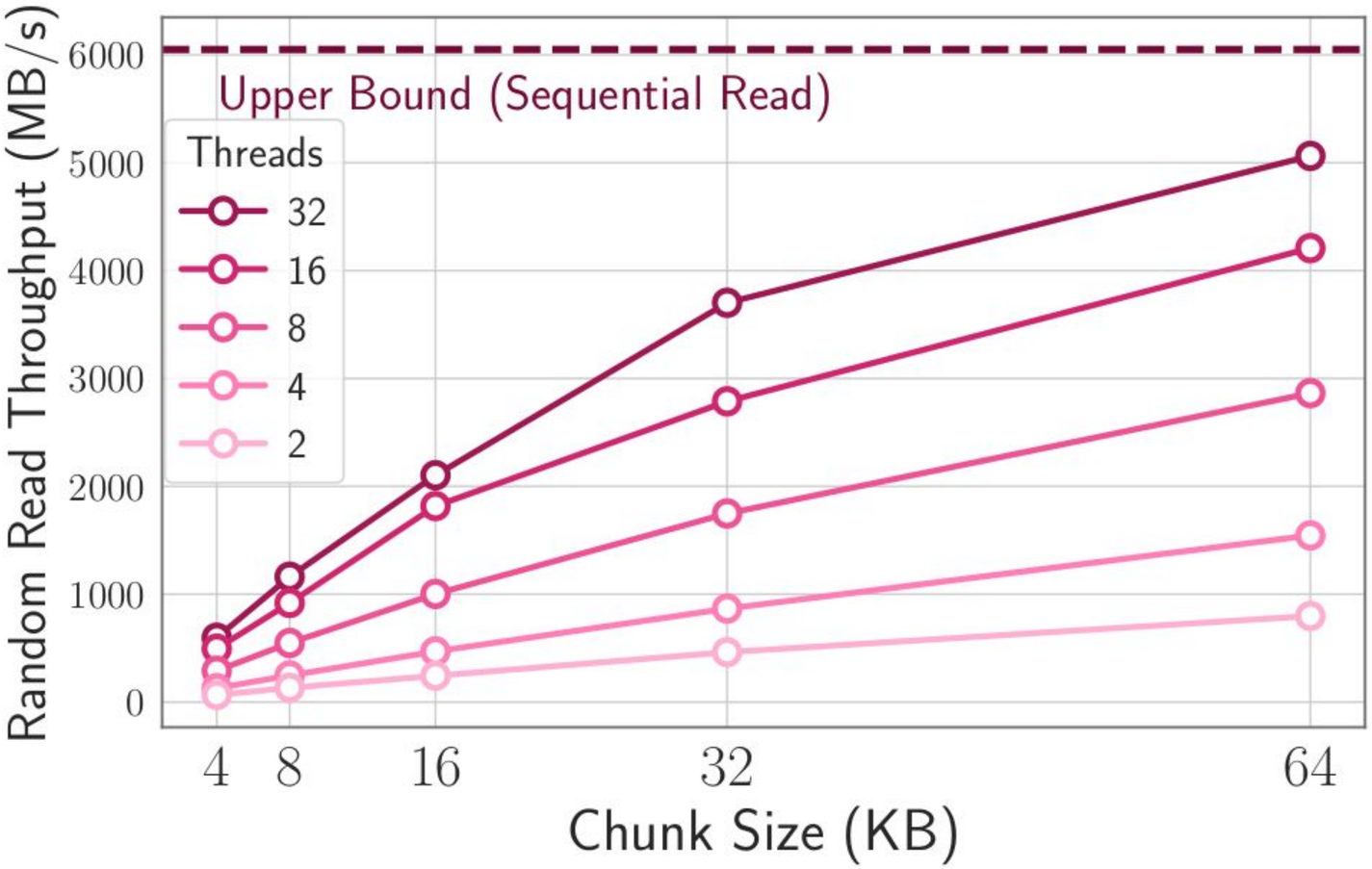


Alizadeh-Vahid, Keivan et al. LLM in a flash: Efficient Large Language Model Inference with Limited Memory. ArXiv 2023.

LLM in a flash: Efficient Large Language Model Inference with Limited Memory

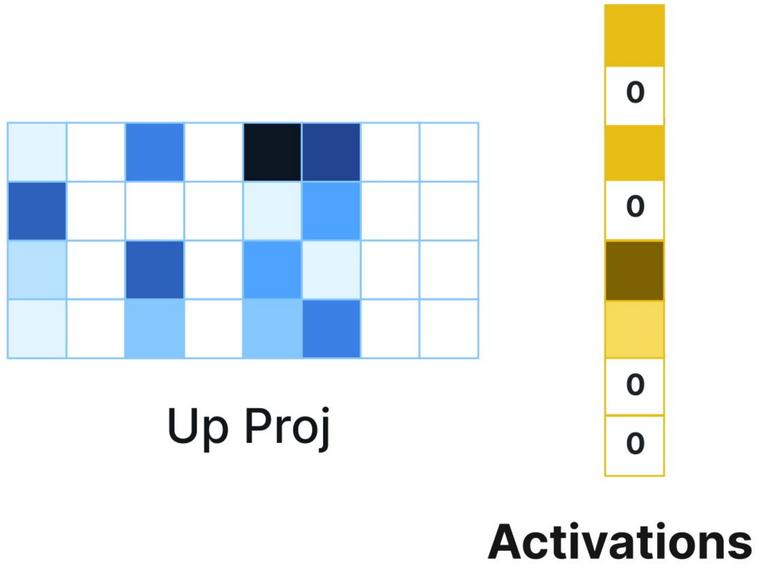


Flash : Read in larger chunks



Alizadeh-Vahid, Keivan et al. LLM in a flash: Efficient Large Language Model Inference with Limited Memory. ArXiv 2023.

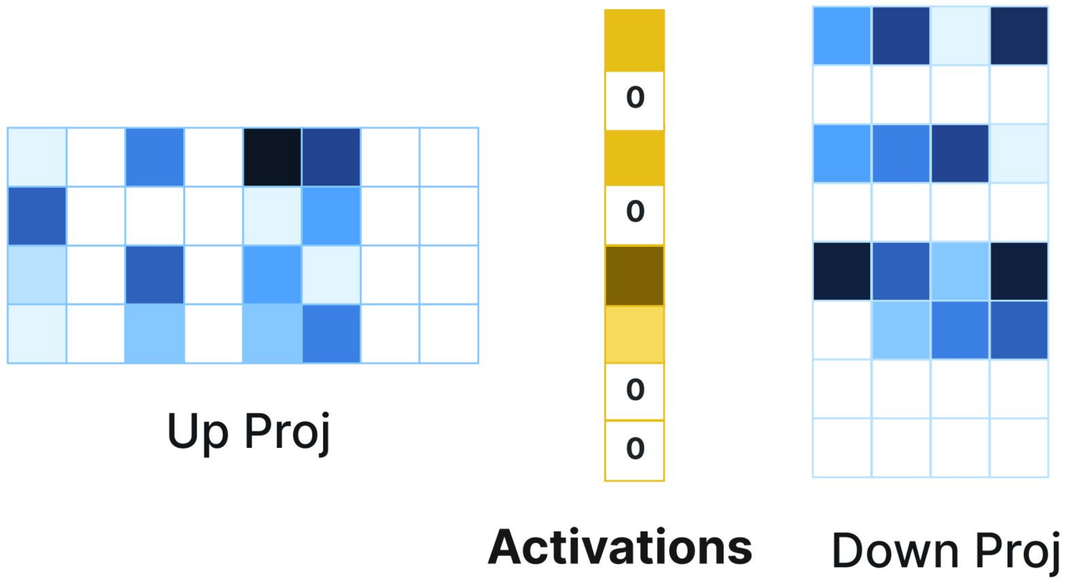
Flash : Read in larger chunks



Alizadeh-Vahid, Keivan et al. LLM in a flash: Efficient Large Language Model Inference with Limited Memory. ArXiv 2023.

Mirzadeh, Iman, et al. Relu strikes back: Exploiting activation sparsity in large language models. arXiv 2023.

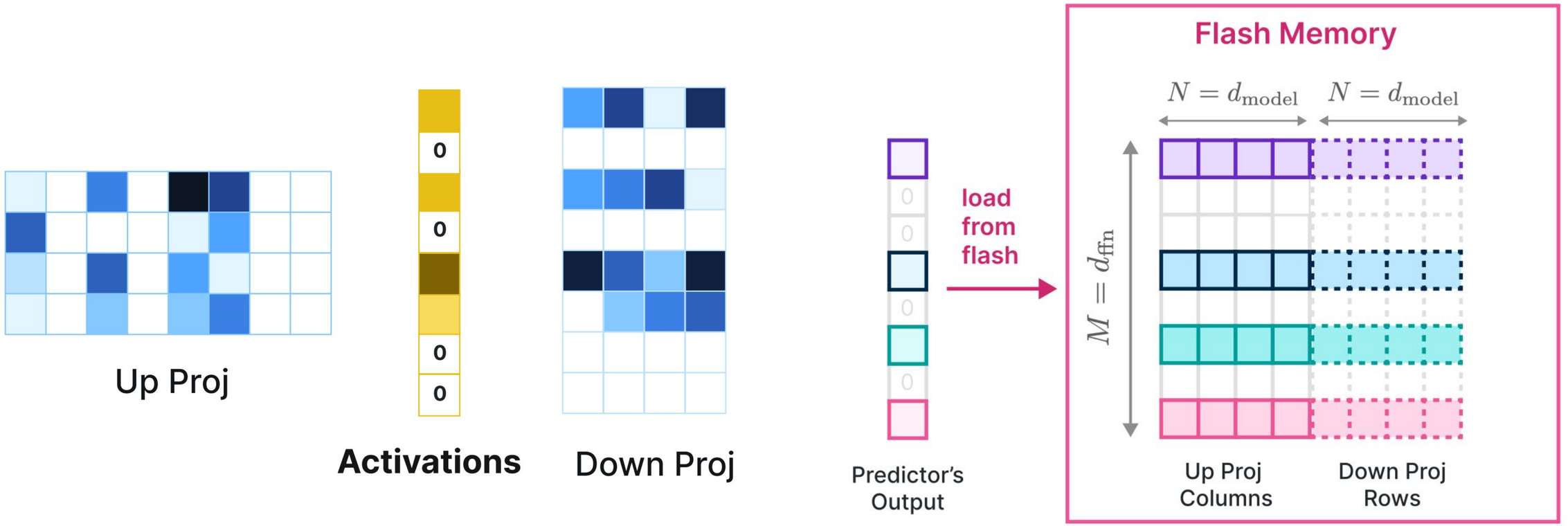
Flash : Read in larger chunks



Alizadeh-Vahid, Keivan et al. LLM in a flash: Efficient Large Language Model Inference with Limited Memory. ArXiv 2023.

Mirzadeh, Iman, et al. Relu strikes back: Exploiting activation sparsity in large language models. arXiv 2023.

Flash : Read in larger chunks



Alizadeh-Vahid, Keivan et al. LLM in a flash: Efficient Large Language Model Inference with Limited Memory. ArXiv 2023.

Mirzadeh, Iman, et al. Relu strikes back: Exploiting activation sparsity in large language models. arXiv 2023.

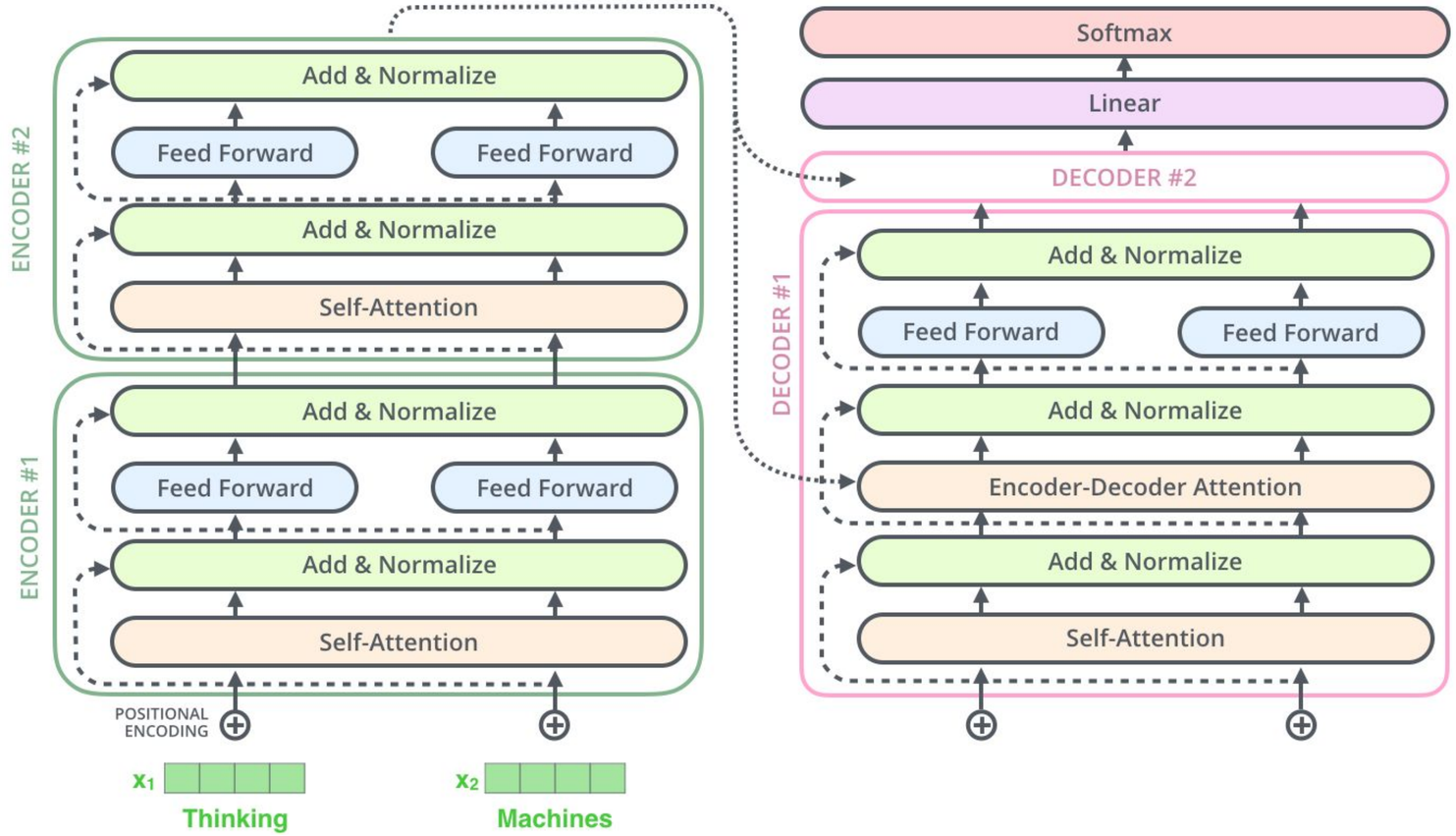
LLM in a flash: Efficient Large Language Model Inference with Limited Memory

Table 2: The I/O latency of OPT 6.7B 16 bit on M1 max for different techniques when half the memory is available

Configuration				Performance Metrics			
Hybrid	**Predictor**	**Windowing**	**Bundling**	**DRAM (GB)**	**Flash→ DRAM(GB)**	**Throughput (GB/s)**	**I/O Latency (ms)**
✗	✗	✗	✗	0	13.4 GB	6.10 GB/s	2130 ms
✓	✗	✗	✗	6.7	6.7 GB	6.10 GB/s	1090 ms
✓	✓	✗	✗	4.8	0.9 GB	1.25 GB/s	738 ms
✓	✓	✓	✗	6.5	0.2 GB	1.25 GB/s	164 ms
✓	✓	✓	✓	6.5	0.2 GB	2.25 GB/s	87 ms

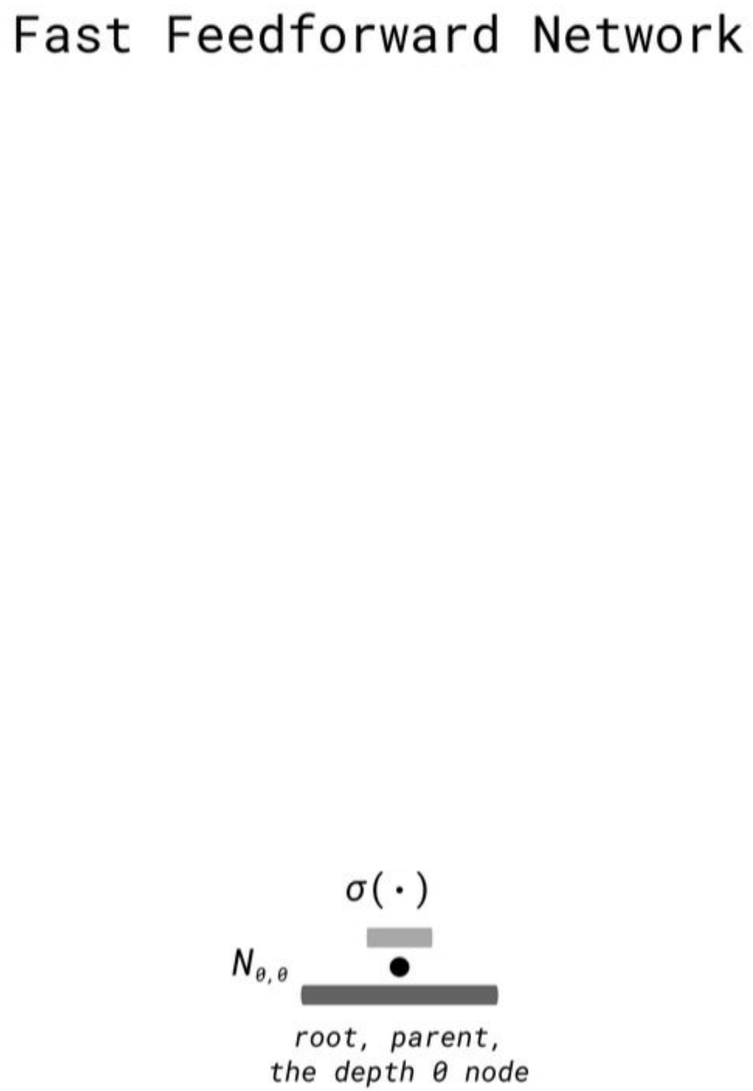
Alizadeh-Vahid, Keivan et al. LLM in a flash: Efficient Large Language Model Inference with Limited Memory. ArXiv 2023.

UltraFastBERT : Exponentially Faster Language Modelling



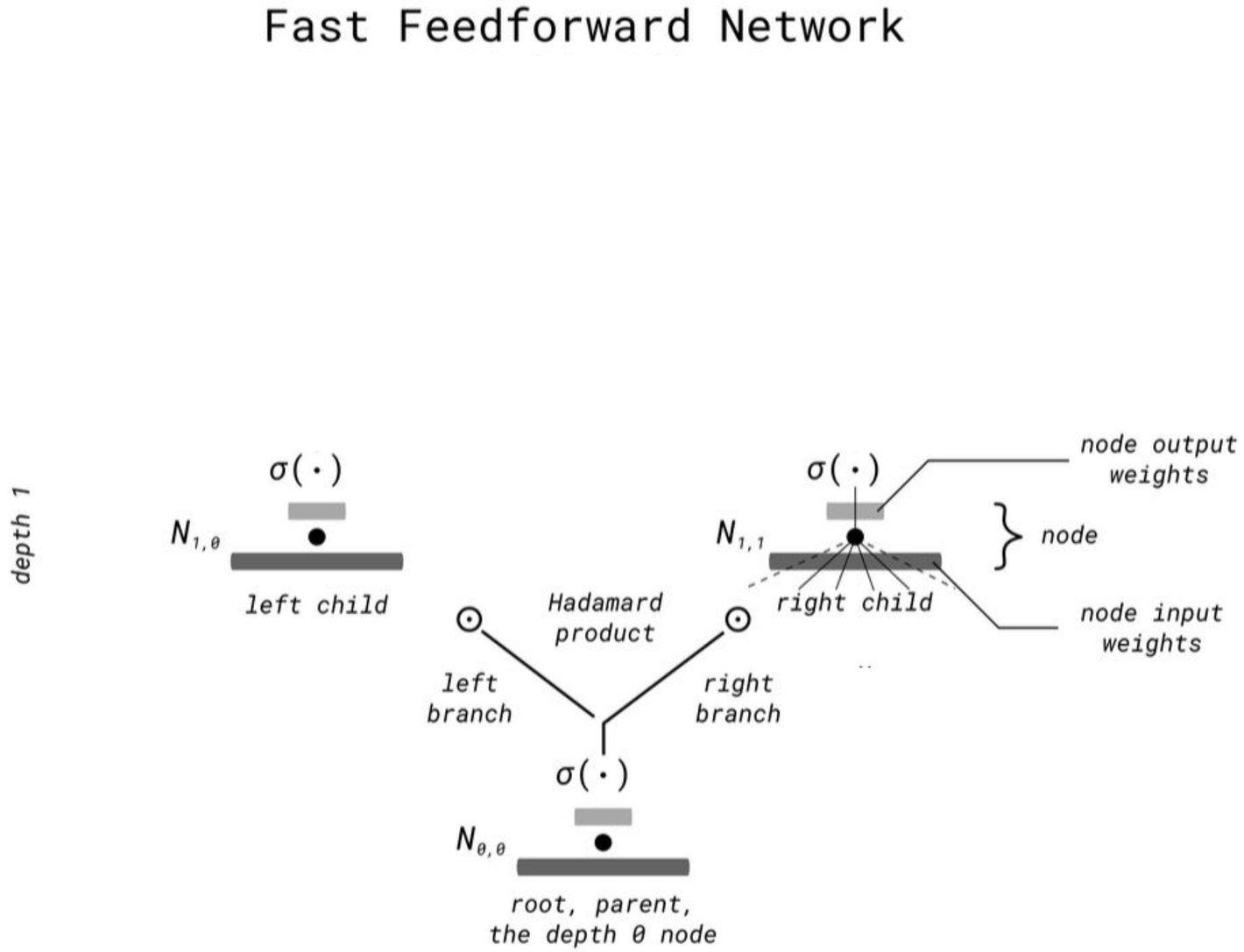
Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

UltraFastBERT : Exponentially Faster Language Modelling



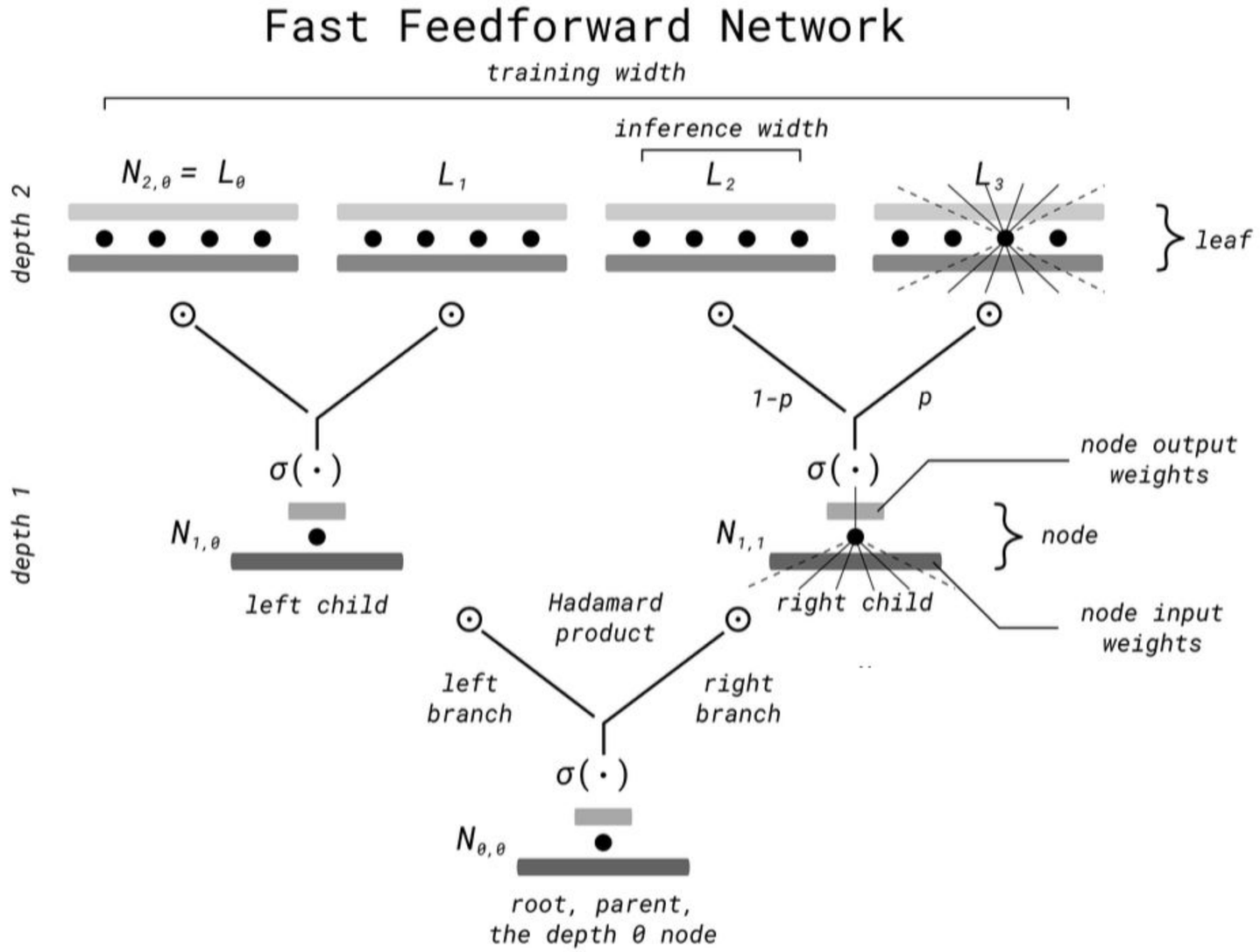
Belcak, Peter, and Roger Wattenhofer. Fast feedforward networks. arXiv 2023.

UltraFastBERT : Exponentially Faster Language Modelling



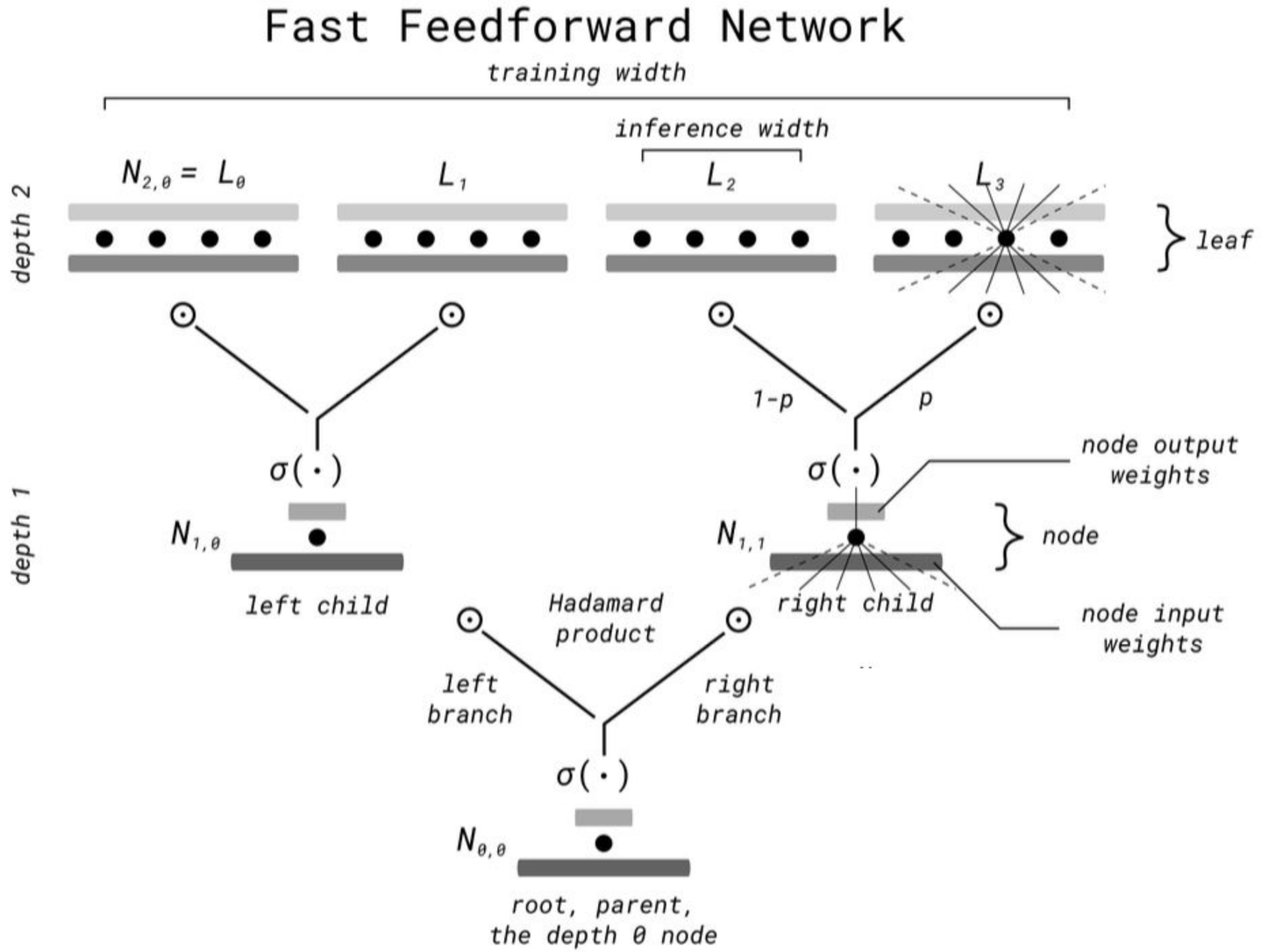
Belcak, Peter, and Roger Wattenhofer. Fast feedforward networks. arXiv 2023.

UltraFastBERT : Exponentially Faster Language Modelling



Belcak, Peter, and Roger Wattenhofer. Fast feedforward networks. arXiv 2023.

UltraFastBERT : Exponentially Faster Language Modelling

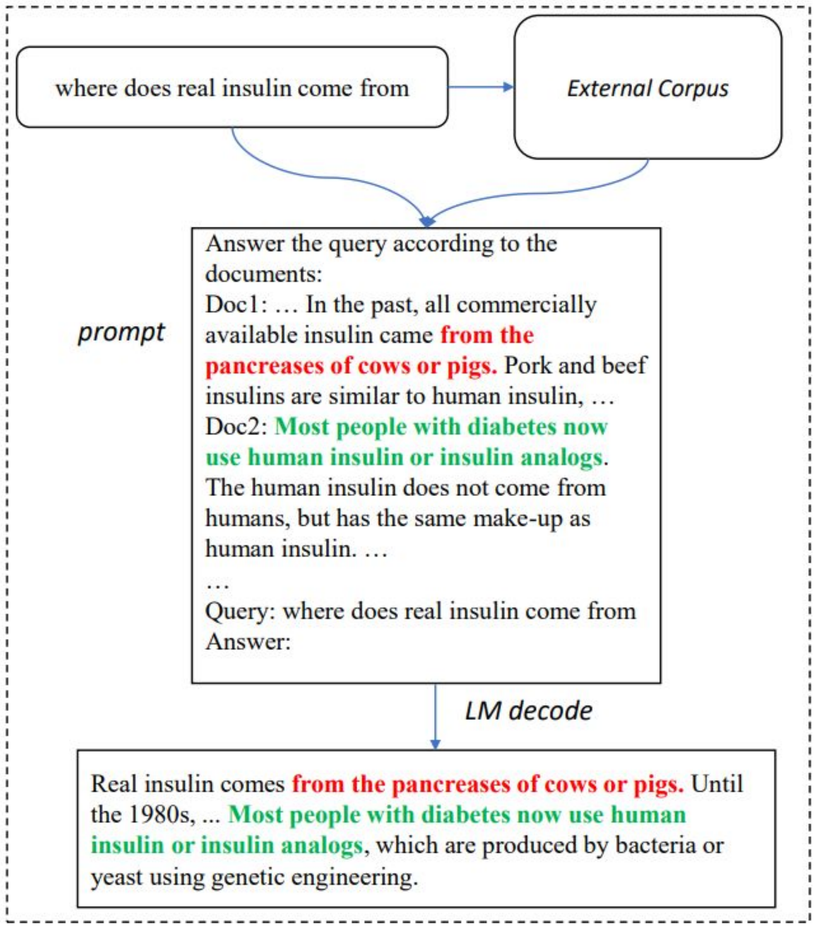


CPU: 78x
GPU: 40x
Neurons used: 0.3%

Belcak, Peter, and Roger Wattenhofer. Fast feedforward networks. arXiv 2023.

Belcak, Peter, and Roger Wattenhofer. Exponentially Faster Language Modelling. arXiv 2023.

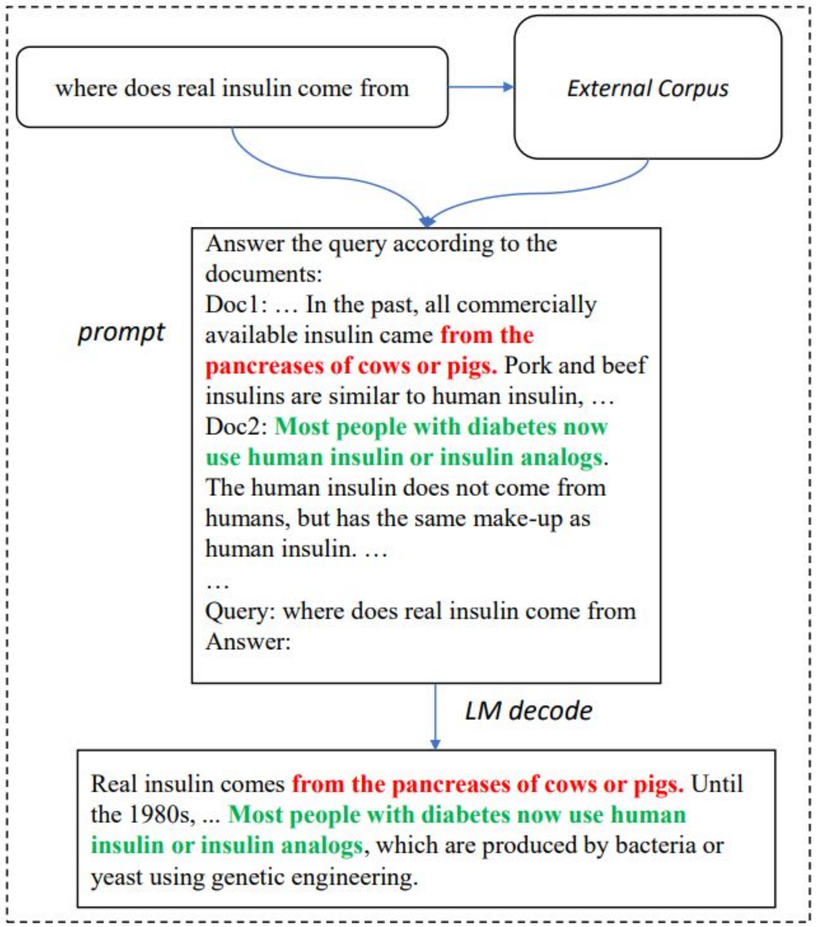
Inference with Reference: Lossless Acceleration of Large Language Models



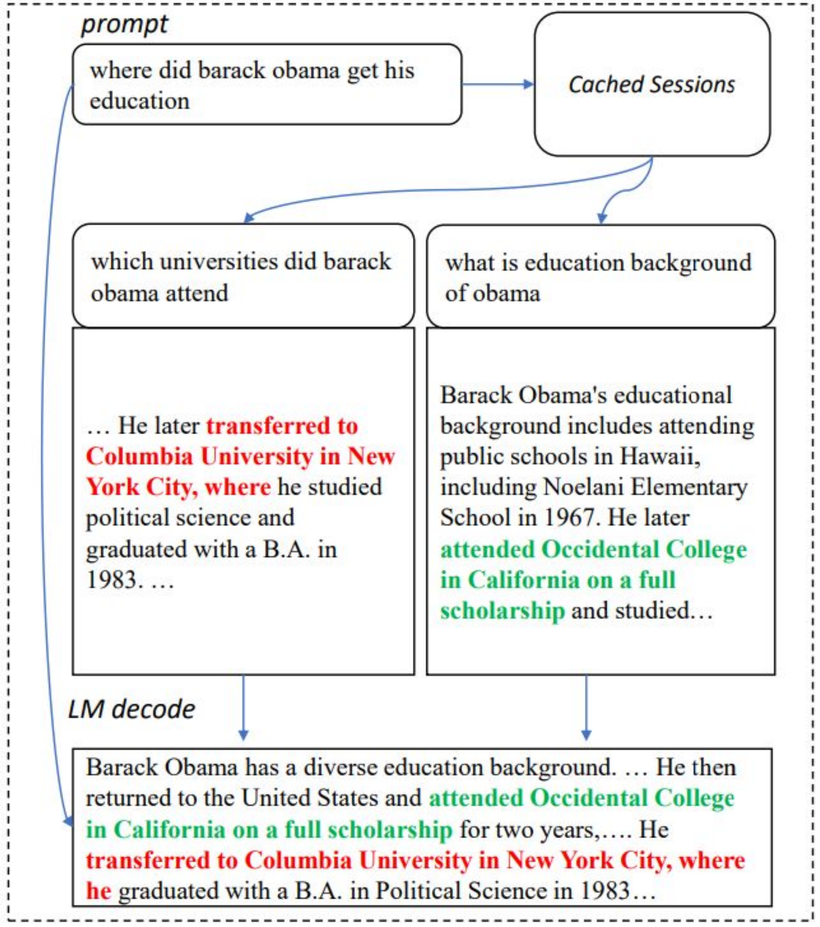
Retrieval-augmented.

Yang, Nan, et al. Inference with reference: Lossless acceleration of large language models. arXiv2023.

Inference with Reference: Lossless Acceleration of Large Language Models



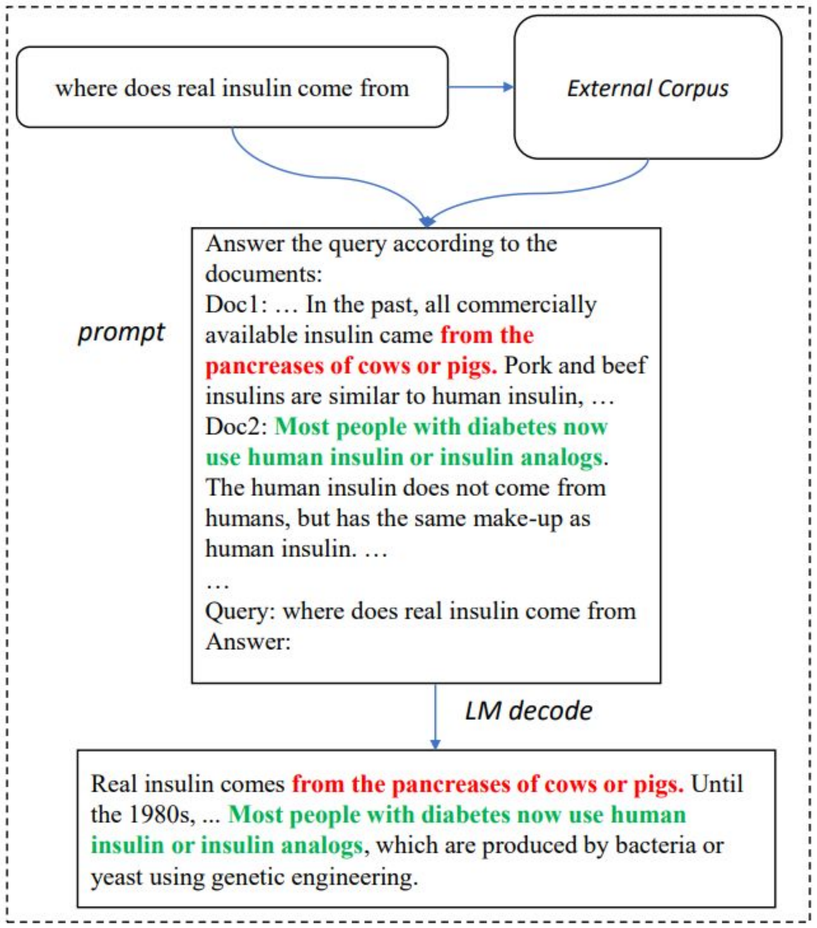
Retrieval-augmented.



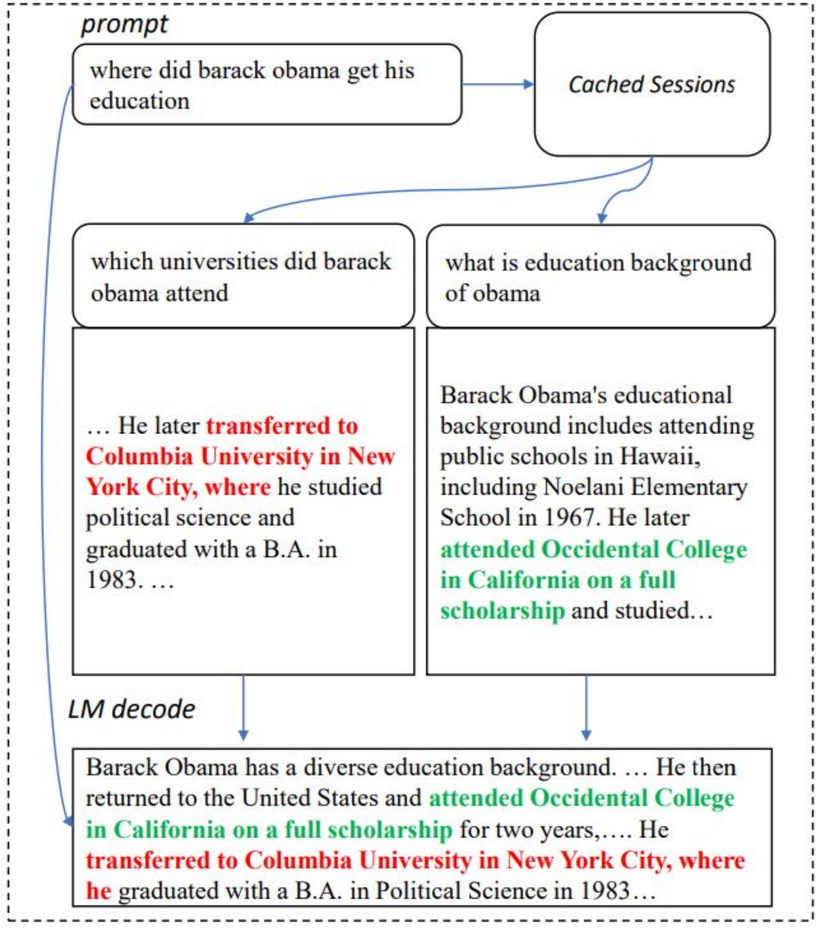
Cache-assisted.

Yang, Nan, et al. Inference with reference: Lossless acceleration of large language models. arXiv2023.

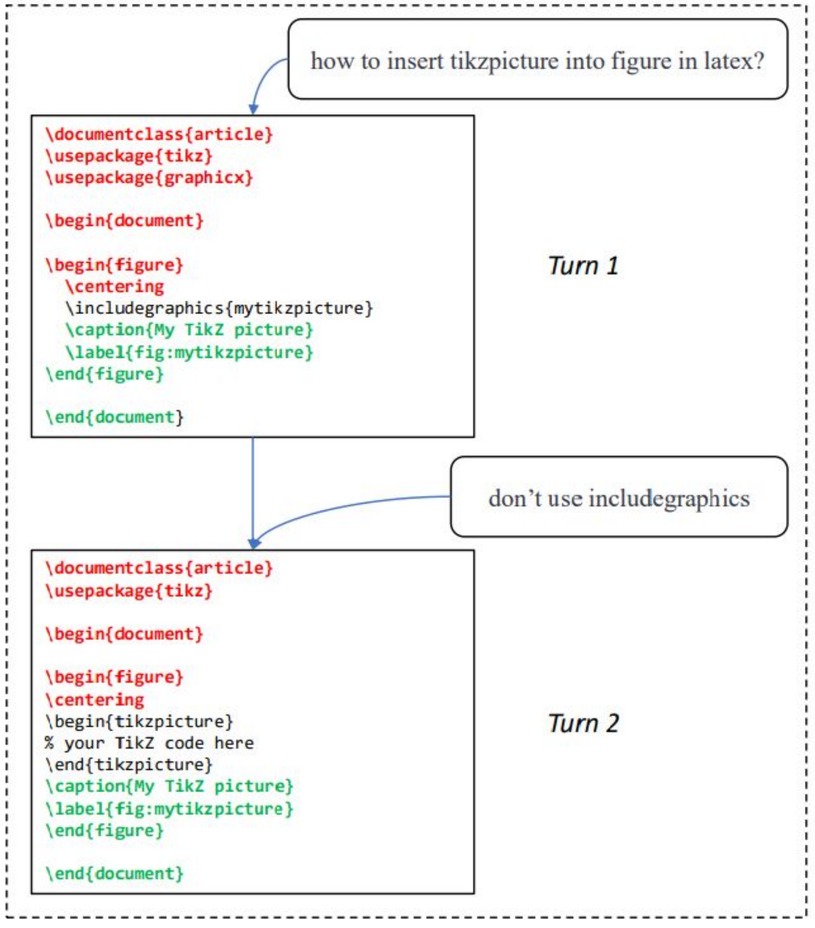
Inference with Reference: Lossless Acceleration of Large Language Models



Retrieval-augmented.



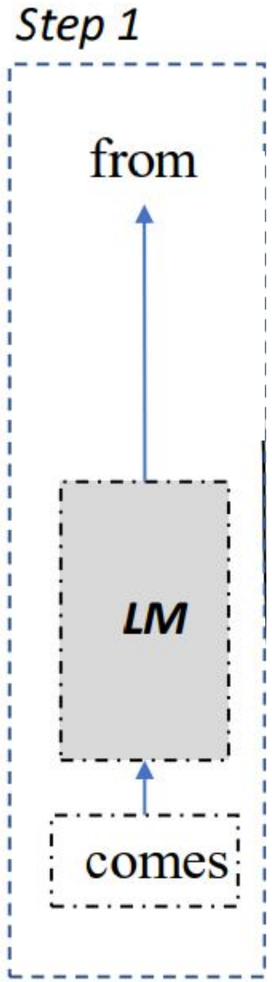
Cache-assisted.



Multi-turn conversations.

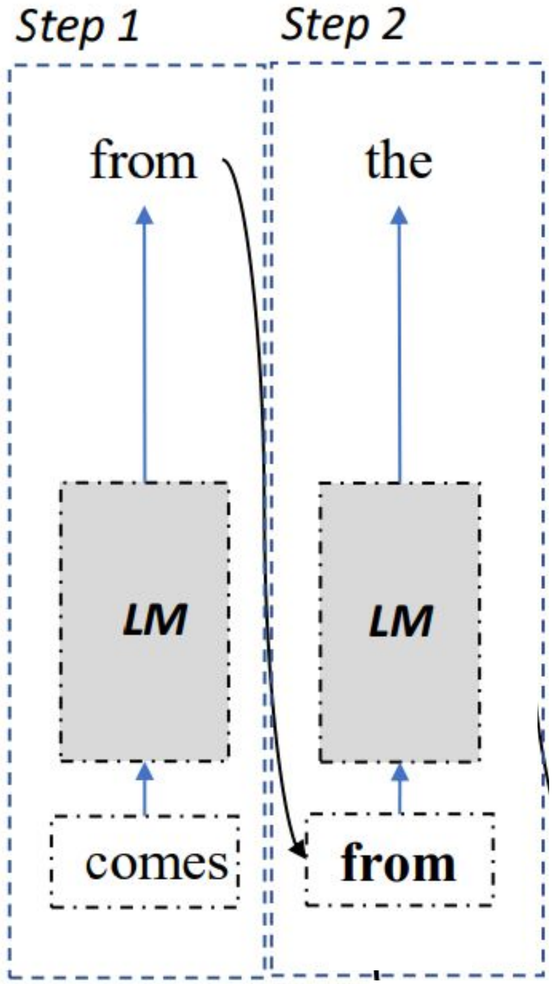
Yang, Nan, et al. Inference with reference: Lossless acceleration of large language models. arXiv2023.

Inference with Reference: Lossless Acceleration of Large Language Models



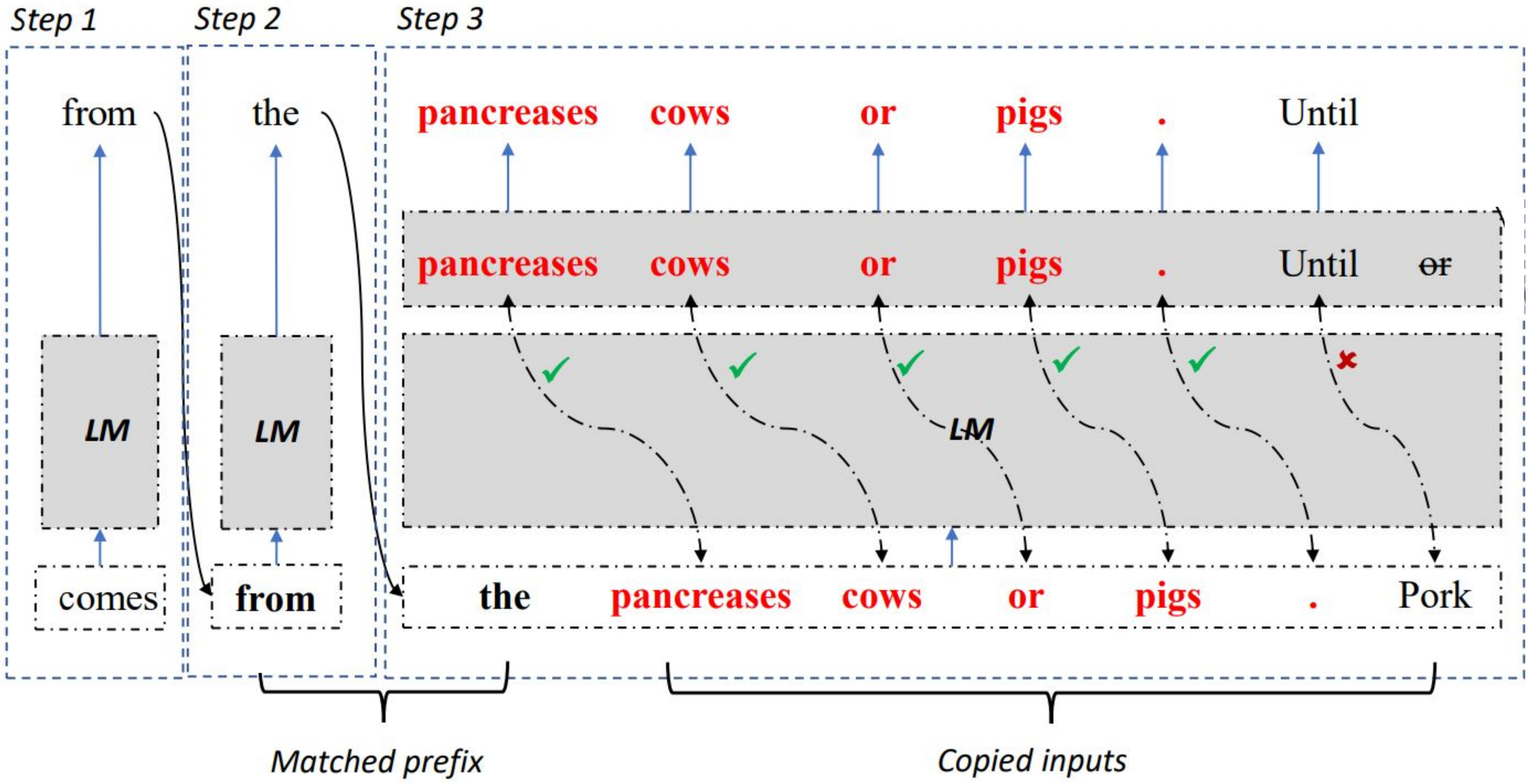
Yang, Nan, et al. Inference with reference: Lossless acceleration of large language models. arXiv2023.

Inference with Reference: Lossless Acceleration of Large Language Models



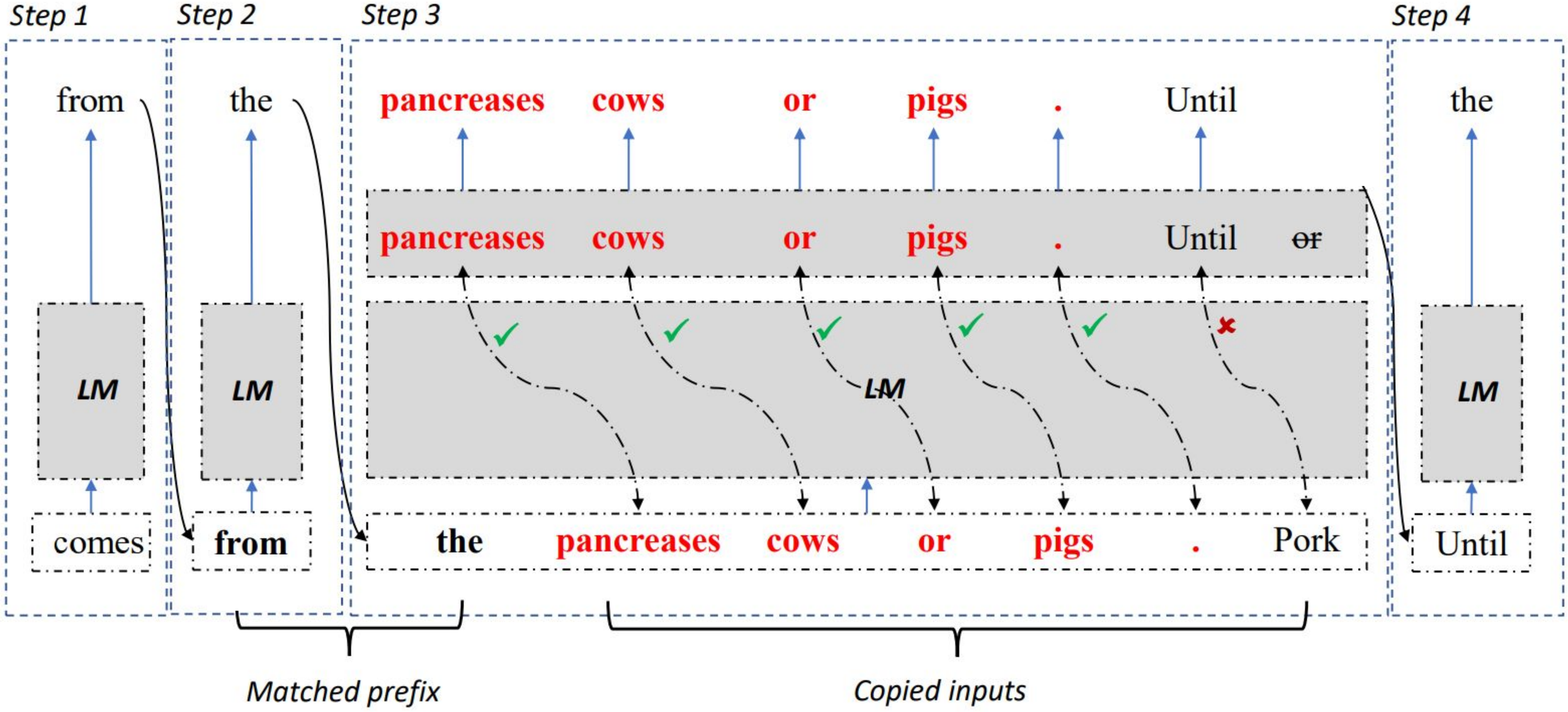
Yang, Nan, et al. Inference with reference: Lossless acceleration of large language models. arXiv2023.

Inference with Reference: Lossless Acceleration of Large Language Models



Yang, Nan, et al. Inference with reference: Lossless acceleration of large language models. arXiv2023.

Inference with Reference: Lossless Acceleration of Large Language Models

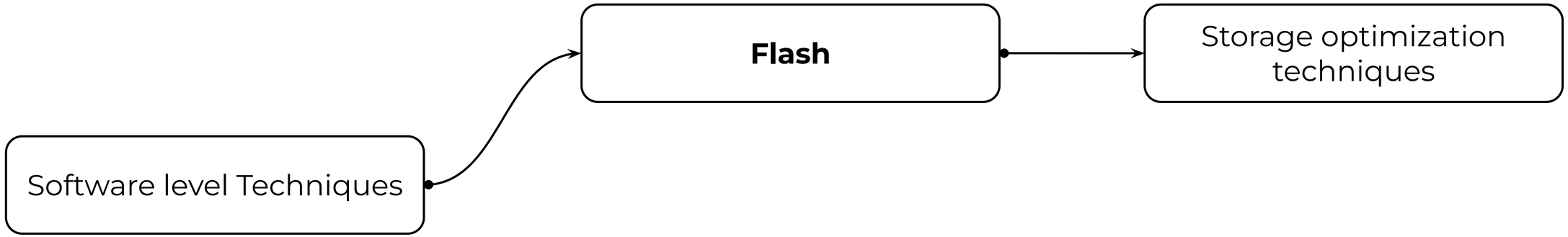


Yang, Nan, et al. Inference with reference: Lossless acceleration of large language models. arXiv2023.

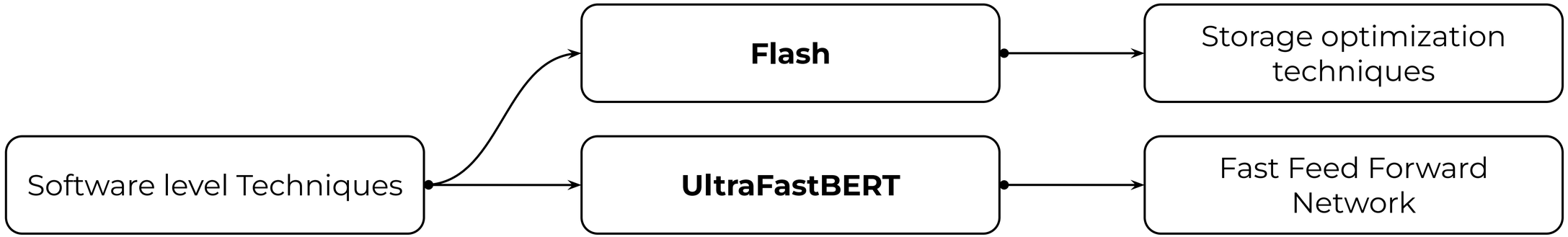
So far.

Software level Techniques

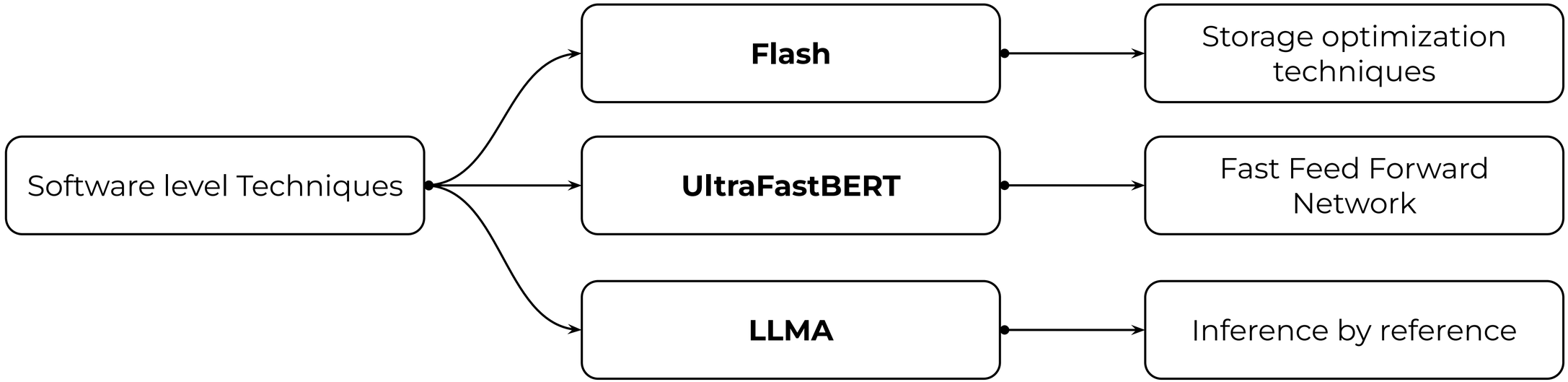
So far.


Flash
Storage optimization techniques
Software level Techniques

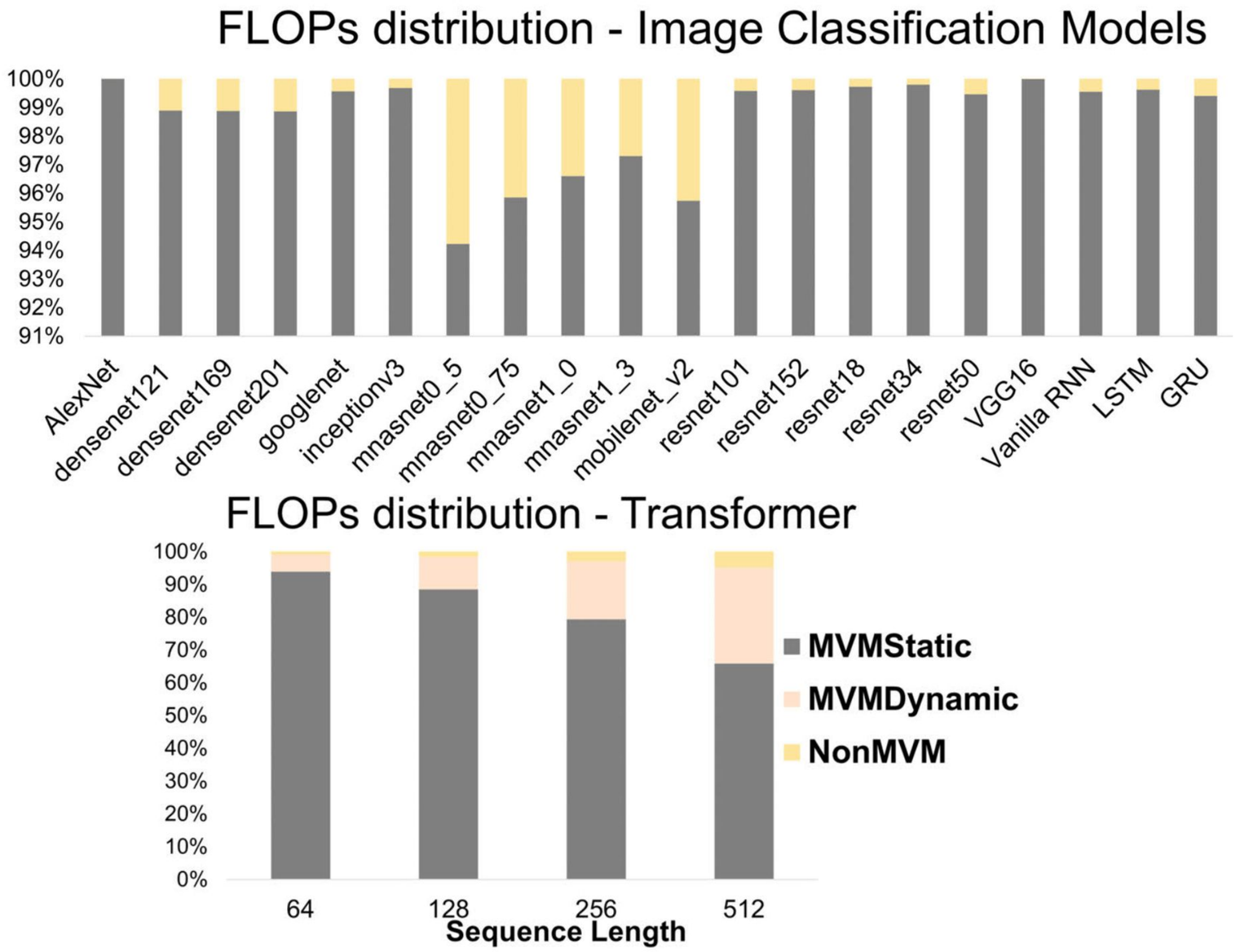
So far.


Software level Techniques
Flash
UltraFastBERT
Storage optimization techniques
Fast Feed Forward Network

So far.

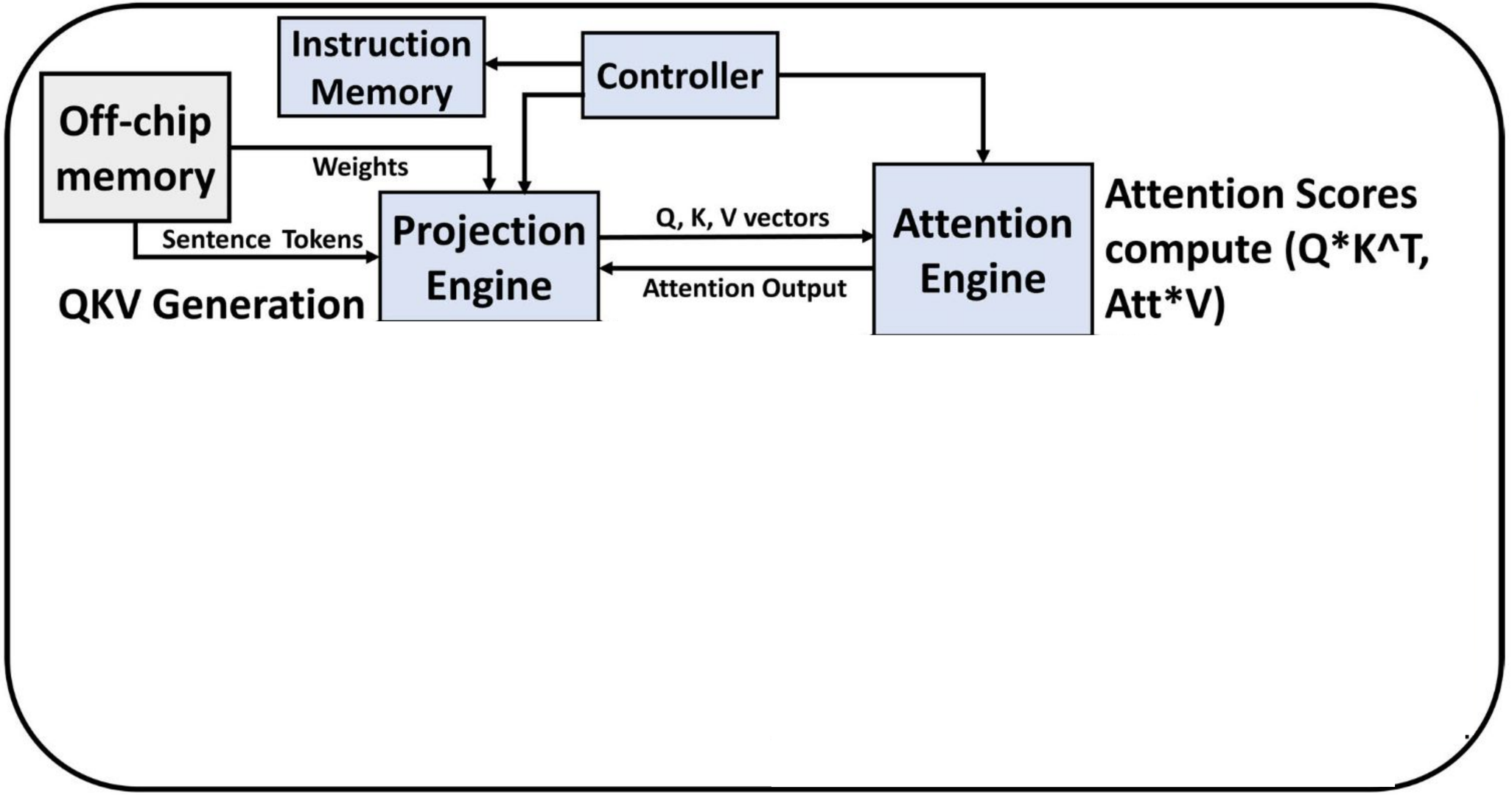

Software level Techniques
Flash
UltraFastBERT
LLMA
Storage optimization techniques
Fast Feed Forward Network
Inference by reference

X-Former: In-Memory Acceleration of Transformers



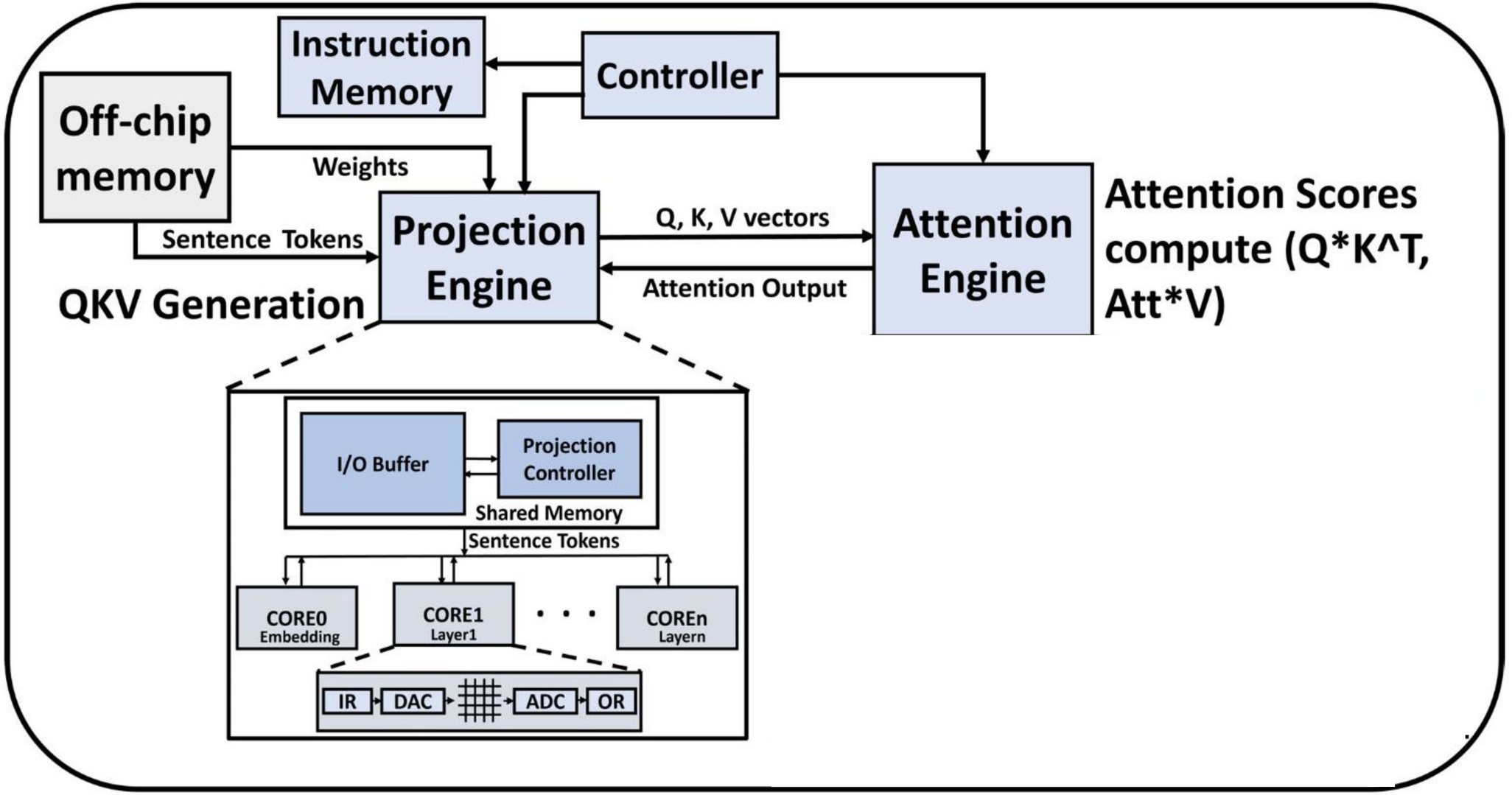
Sridharan, Shrihari, et al. X-former: In-memory acceleration of transformers. IEEE TVLSI Systems 2023.

X-Former: In-Memory Acceleration of Transformers



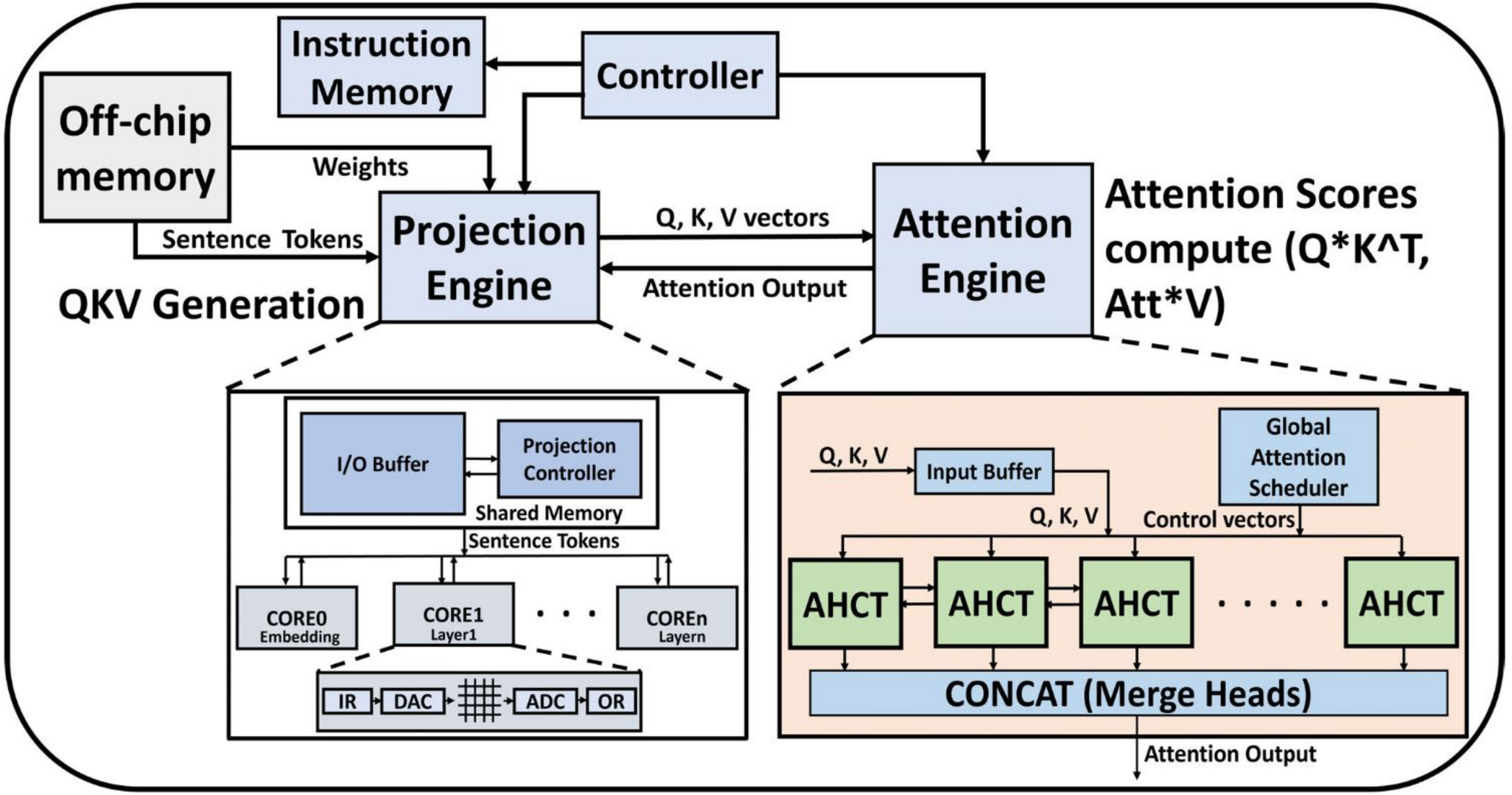
Sridharan, Shrihari, et al. X-former: In-memory acceleration of transformers. IEEE TVLSI Systems 2023.

X-Former: In-Memory Acceleration of Transformers



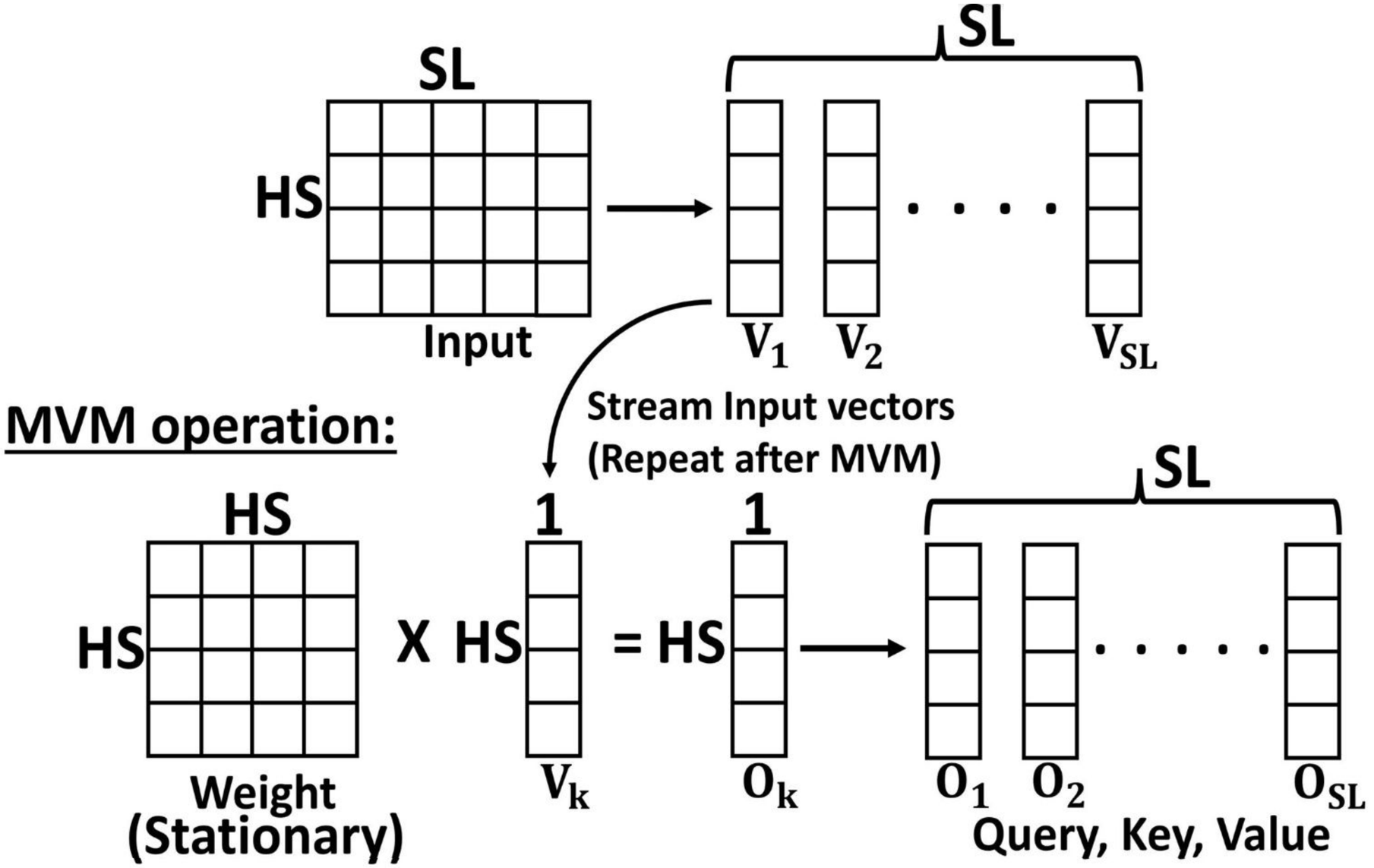
Sridharan, Shrihari, et al. X-former: In-memory acceleration of transformers. IEEE TVLSI Systems 2023.

X-Former: In-Memory Acceleration of Transformers



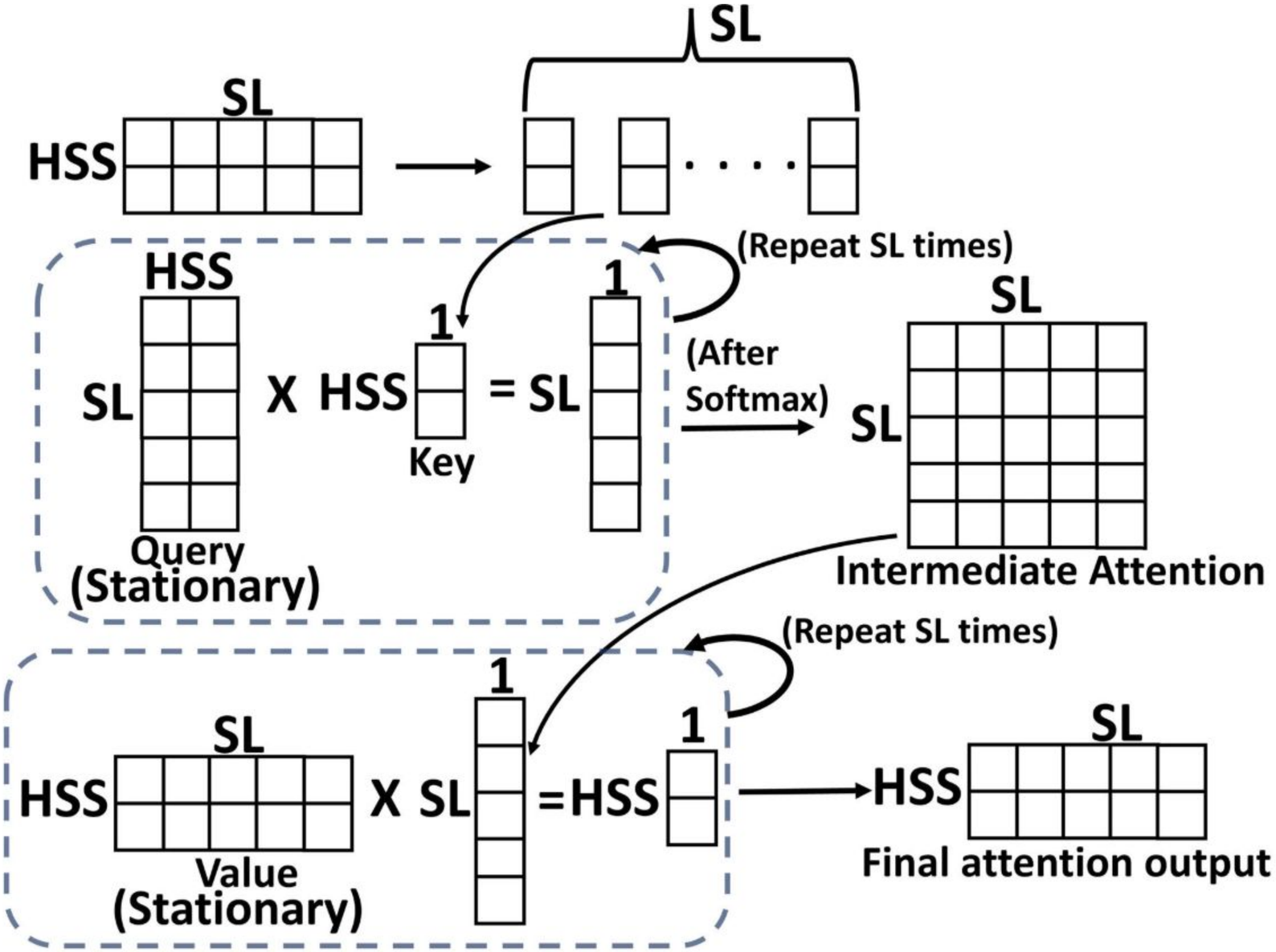
Sridharan, Shrihari, et al. X-former: In-memory acceleration of transformers. IEEE TVLSI Systems 2023.

X-Former: In-Memory Acceleration of Transformers



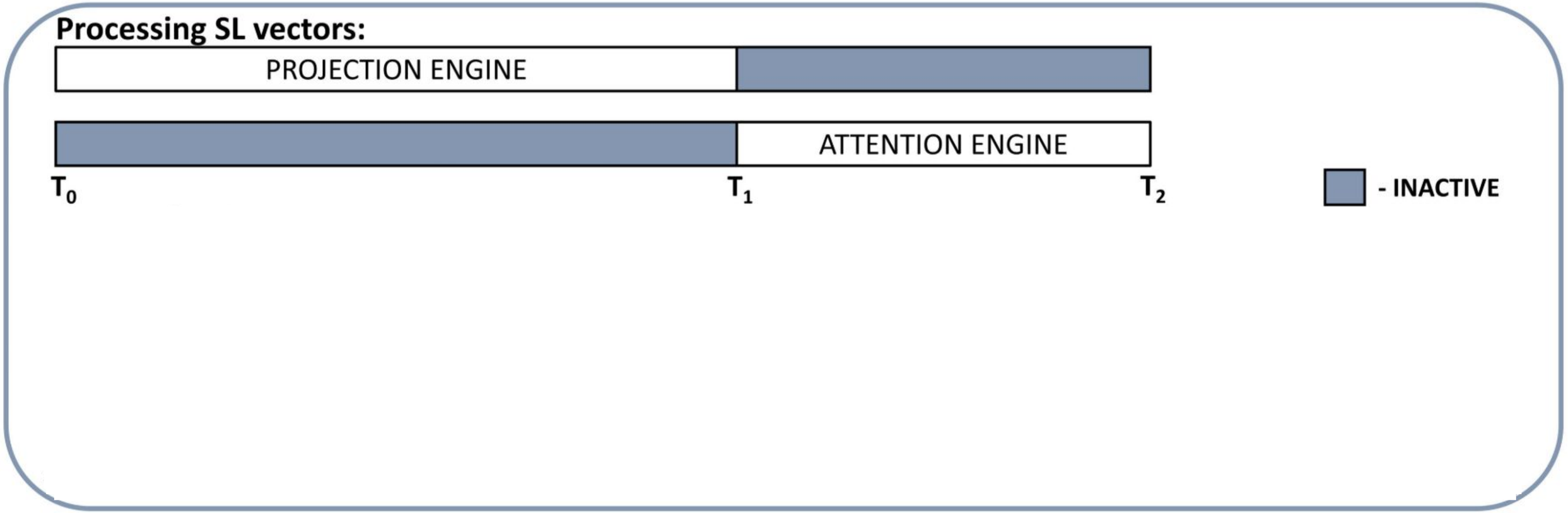
Sridharan, Shrihari, et al. X-former: In-memory acceleration of transformers. IEEE TVLSI Systems 2023.

X-Former: In-Memory Acceleration of Transformers



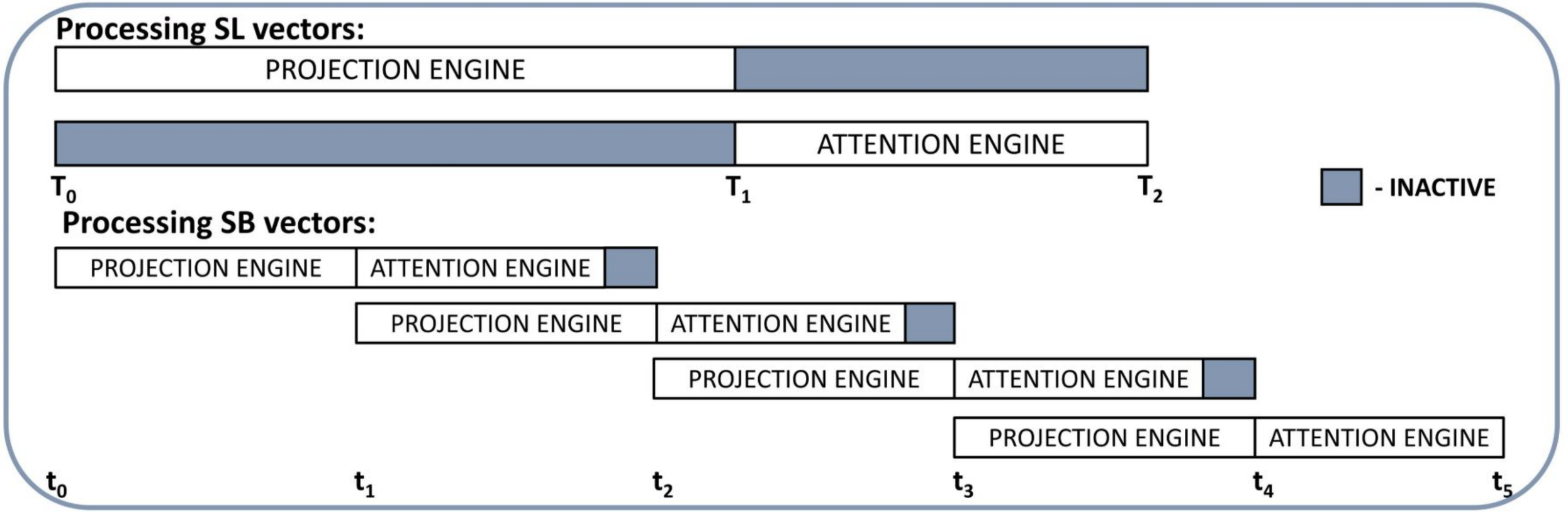
Sridharan, Shrihari, et al. X-former: In-memory acceleration of transformers. IEEE TVLSI Systems 2023.

X-Former: In-Memory Acceleration of Transformers



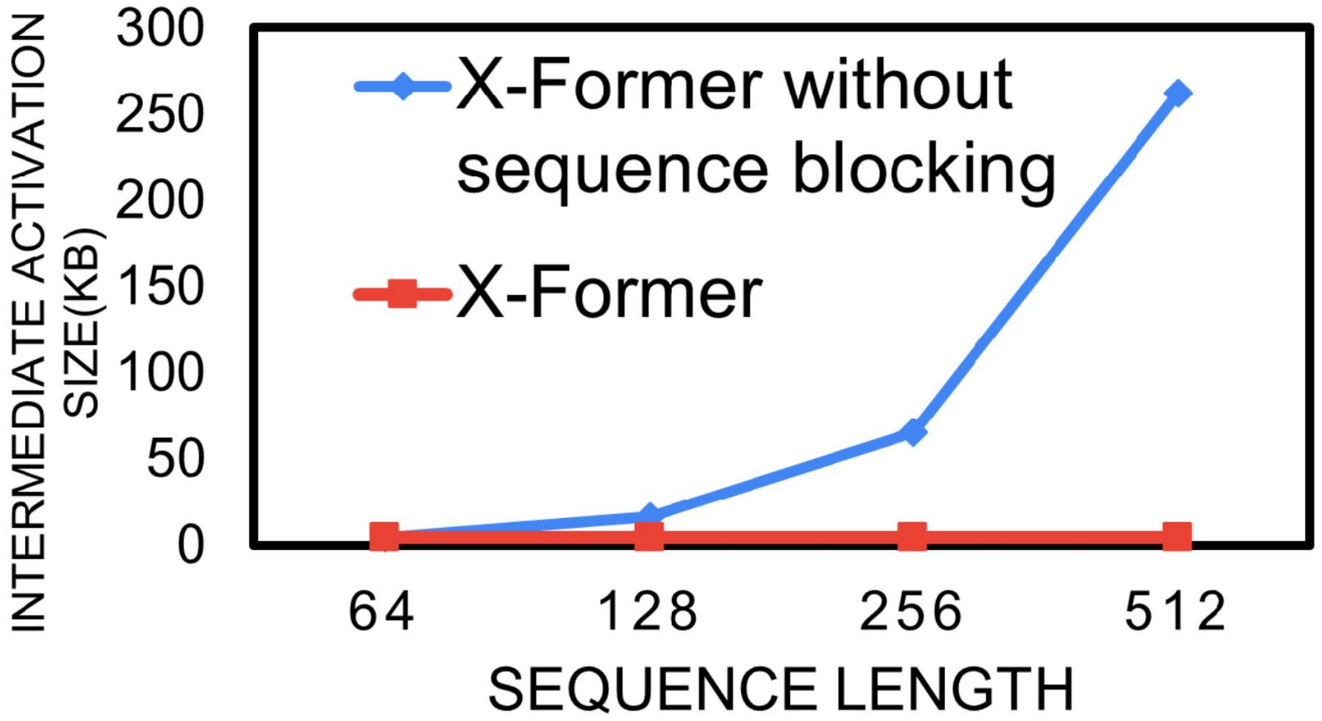
Sridharan, Shrihari, et al. X-former: In-memory acceleration of transformers. IEEE TVLSI Systems 2023.

X-Former: In-Memory Acceleration of Transformers



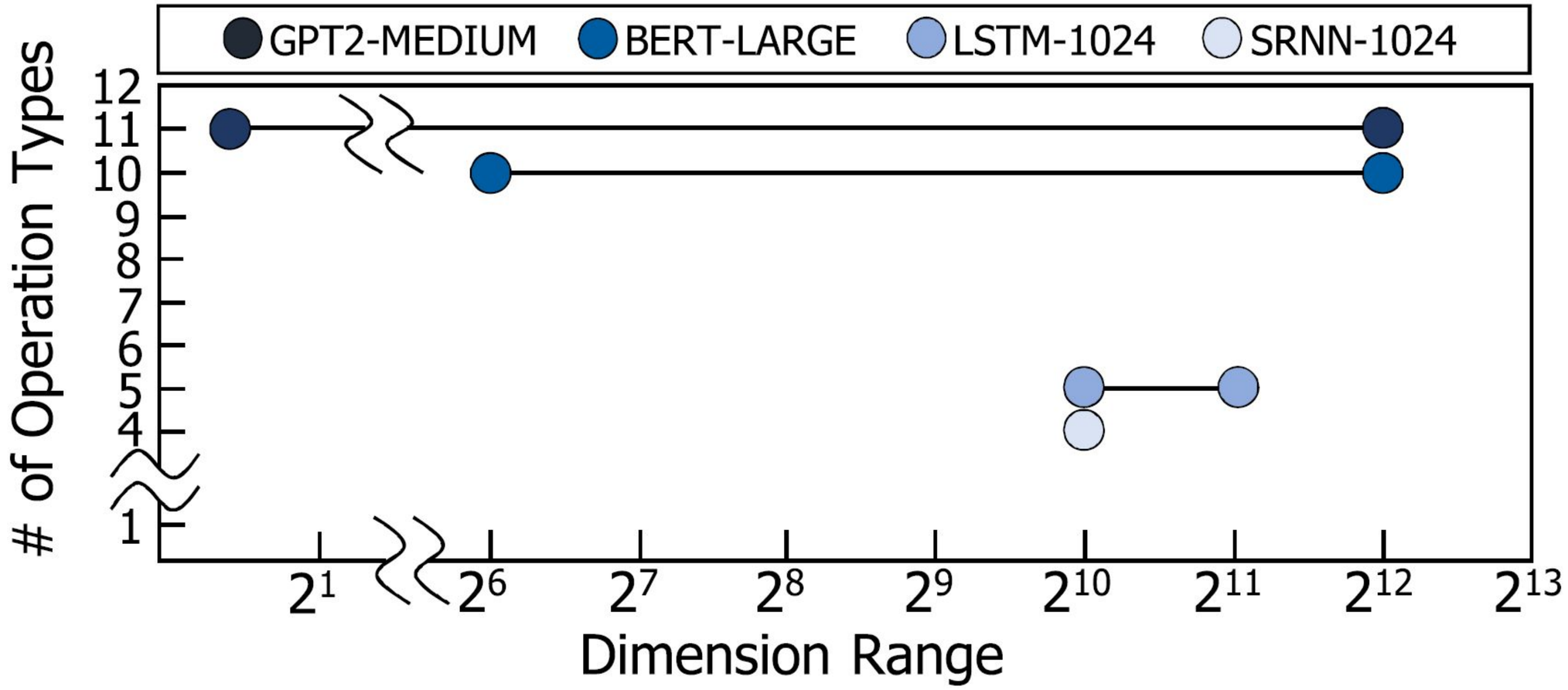
Sridharan, Shrihari, et al. X-former: In-memory acceleration of transformers. IEEE TVLSI Systems 2023.

X-Former: In-Memory Acceleration of Transformers



Sridharan, Shrihari, et al. X-former: In-memory acceleration of transformers. IEEE TVLSI Systems 2023.

A Fast and Flexible FPGA-based Accelerator for Natural Language Processing Neural Networks



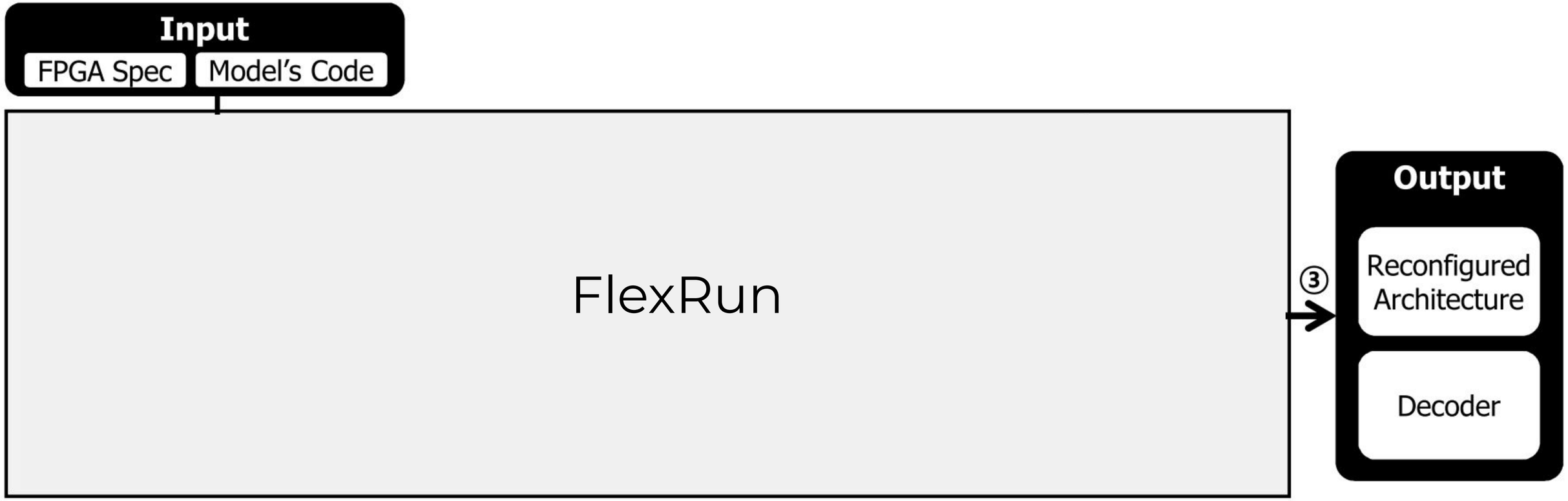
Hur, Suyeon, et al. A fast and flexible FPGA-based accelerator for natural language processing neural networks. ACM TACO 2023.

A Fast and Flexible FPGA-based Accelerator for Natural Language Processing Neural Networks

NLP model	Matrix Operations			Vector Operations					
	gemv	gemm	transpose	activation	exp	add/sub	mul	reduction	square/sqrt/div
SRNN	✓	-	-	tanh	-	✓	✓	-	-
LSTM	✓	-	-	sig/tanh	-	✓	✓	-	-
GRU	✓	-	-	sig/tanh	-	✓	✓	-	-
Transformer	✓	✓	✓	ReLU	✓	✓	✓	✓	✓
BERT	-	✓	✓	gelu	✓	✓	✓	✓	✓
GPT2	✓	✓	✓	gelu	✓	✓	✓	✓	✓

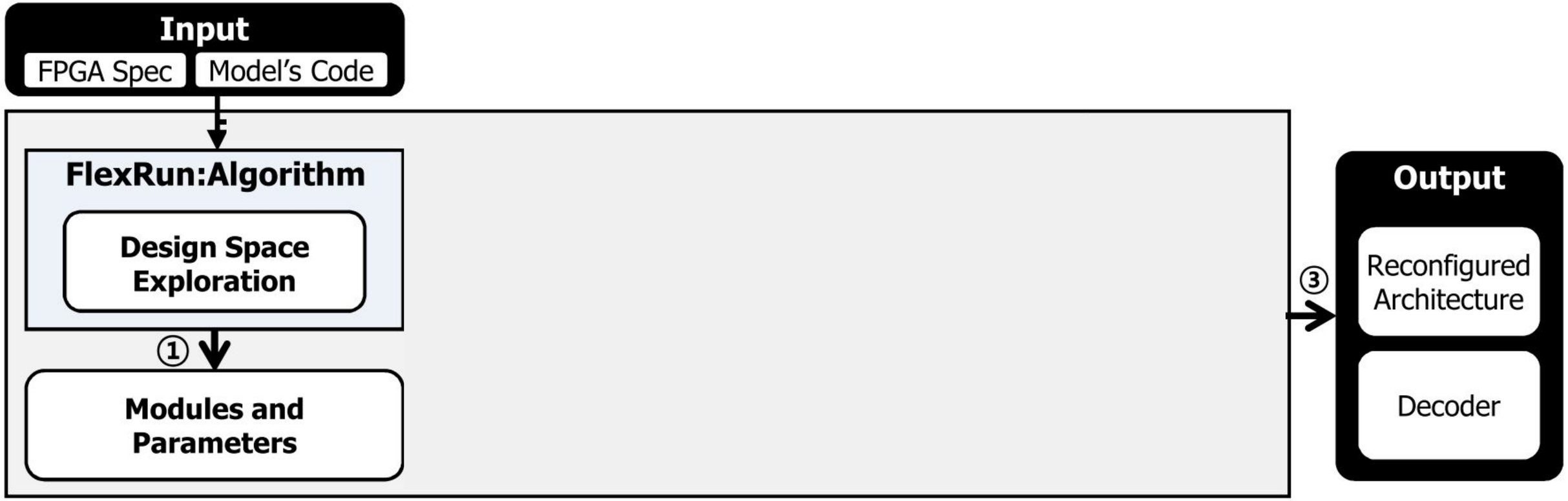
Hur, Suyeon, et al. A fast and flexible FPGA-based accelerator for natural language processing neural networks. ACM TACO 2023.

A Fast and Flexible FPGA-based Accelerator for Natural Language Processing Neural Networks



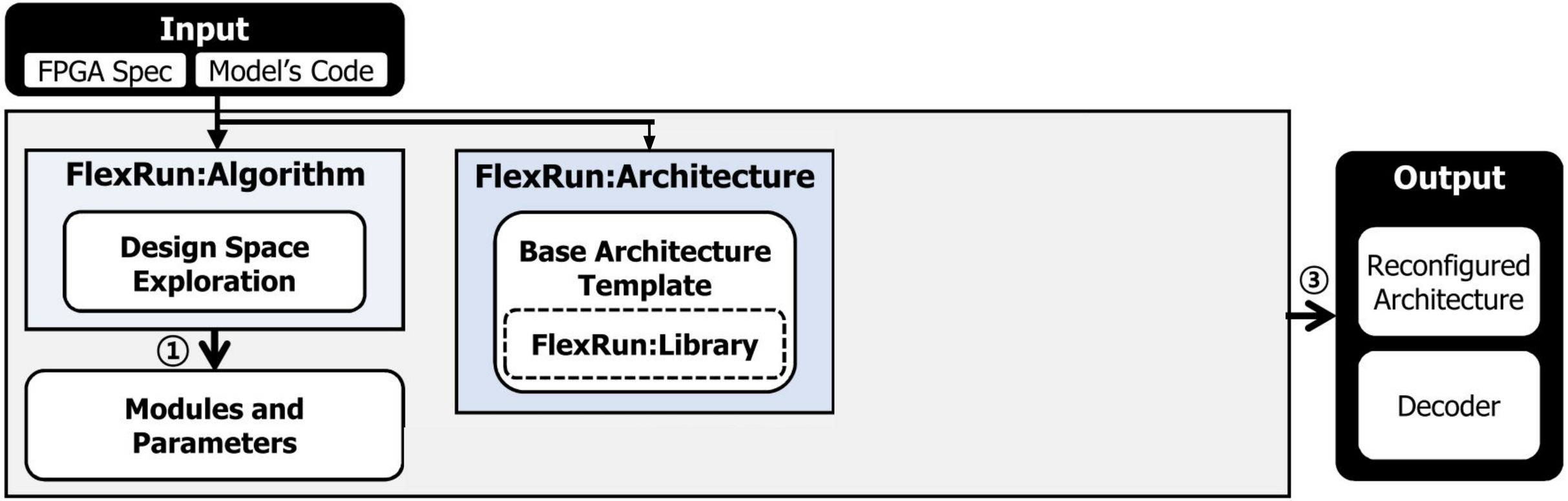
Hur, Suyeon, et al. A fast and flexible FPGA-based accelerator for natural language processing neural networks. ACM TACO 2023.

A Fast and Flexible FPGA-based Accelerator for Natural Language Processing Neural Networks



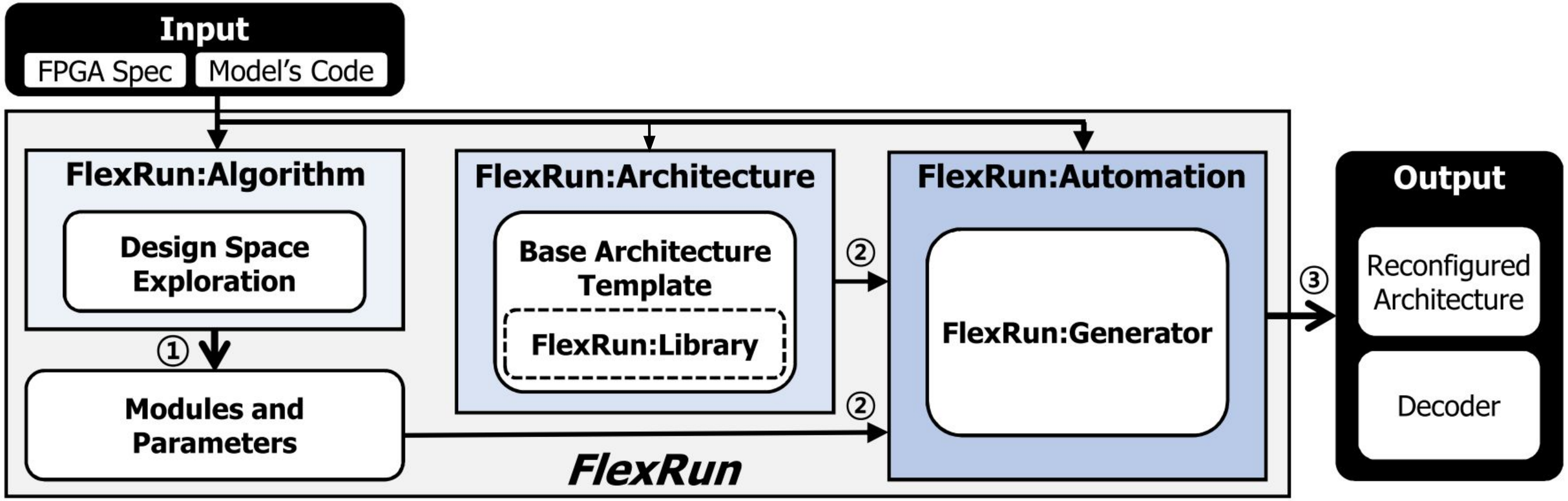
Hur, Suyeon, et al. A fast and flexible FPGA-based accelerator for natural language processing neural networks. ACM TACO 2023.

A Fast and Flexible FPGA-based Accelerator for Natural Language Processing Neural Networks



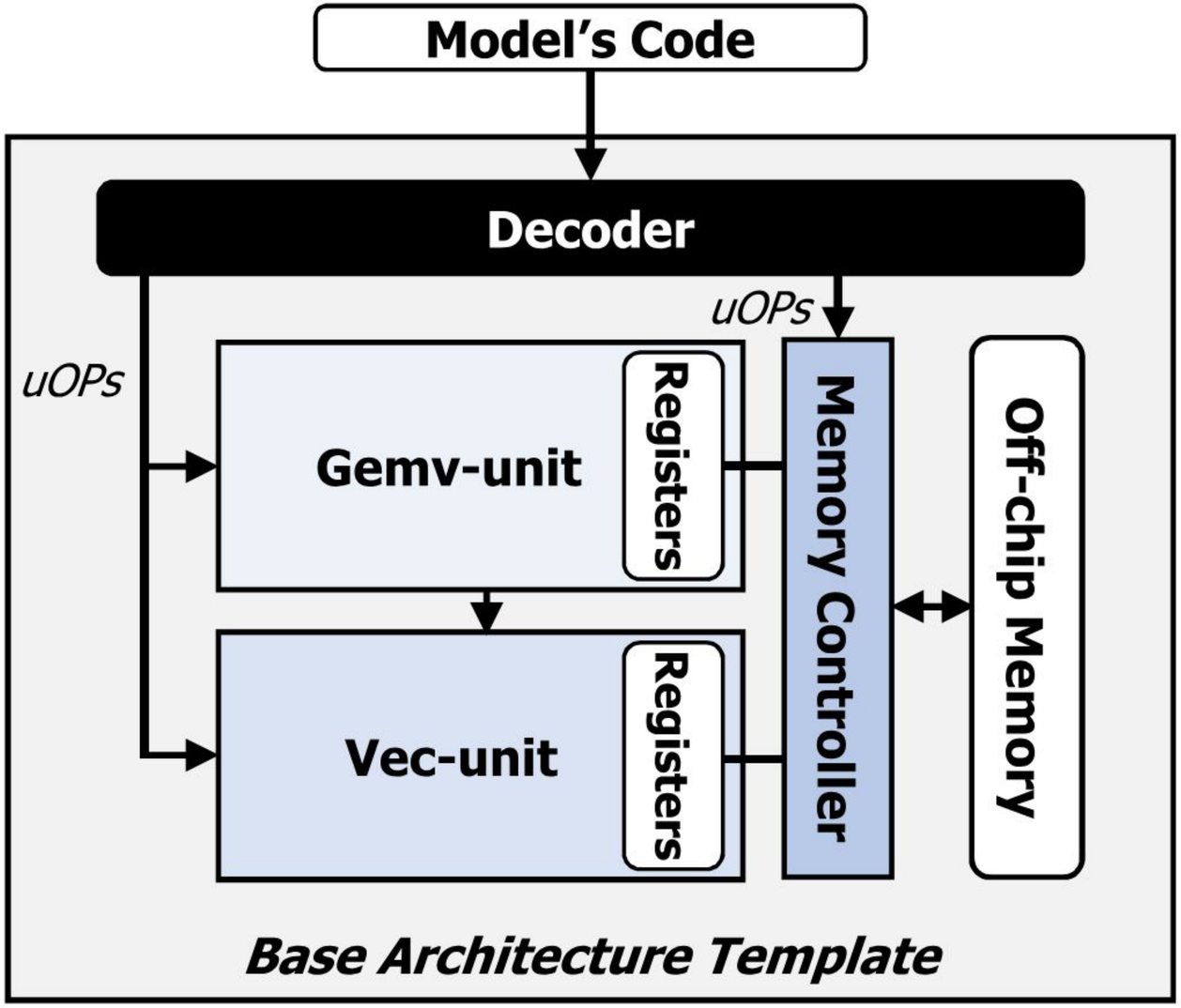
Hur, Suyeon, et al. A fast and flexible FPGA-based accelerator for natural language processing neural networks. ACM TACO 2023.

A Fast and Flexible FPGA-based Accelerator for Natural Language Processing Neural Networks



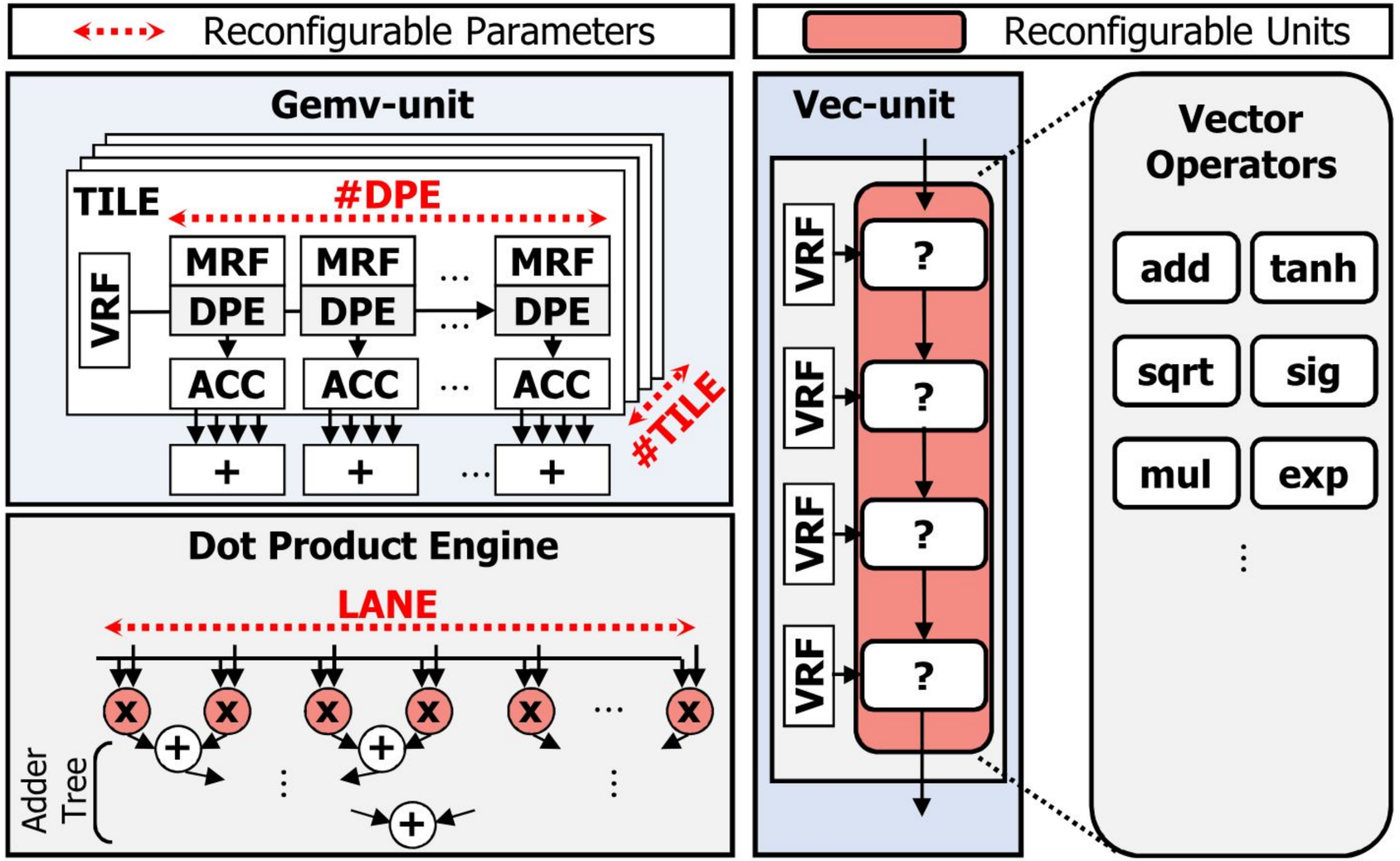
Hur, Suyeon, et al. A fast and flexible FPGA-based accelerator for natural language processing neural networks. ACM TACO 2023.

A Fast and Flexible FPGA-based Accelerator for Natural Language Processing Neural Networks



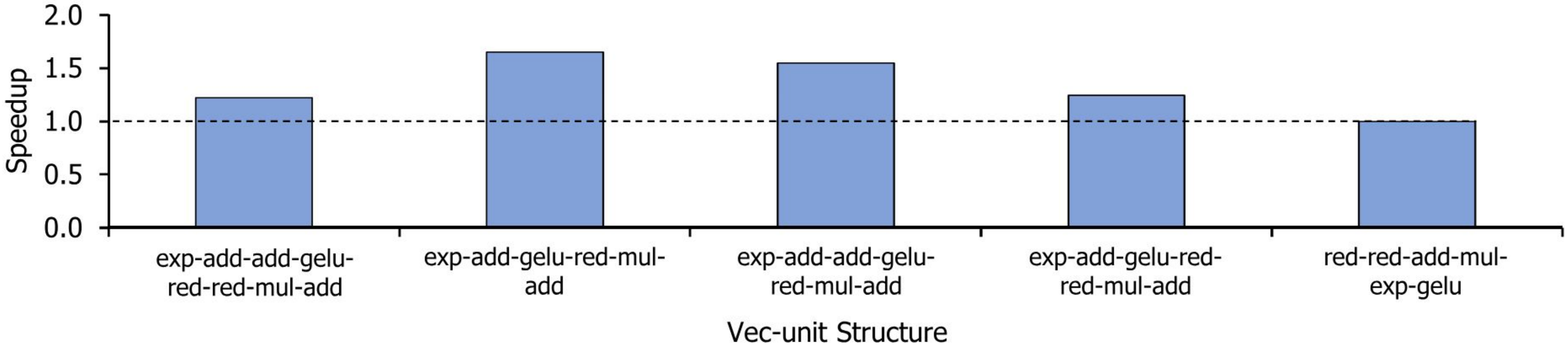
Hur, Suyeon, et al. A fast and flexible FPGA-based accelerator for natural language processing neural networks. ACM TACO 2023.

A Fast and Flexible FPGA-based Accelerator for Natural Language Processing Neural Networks



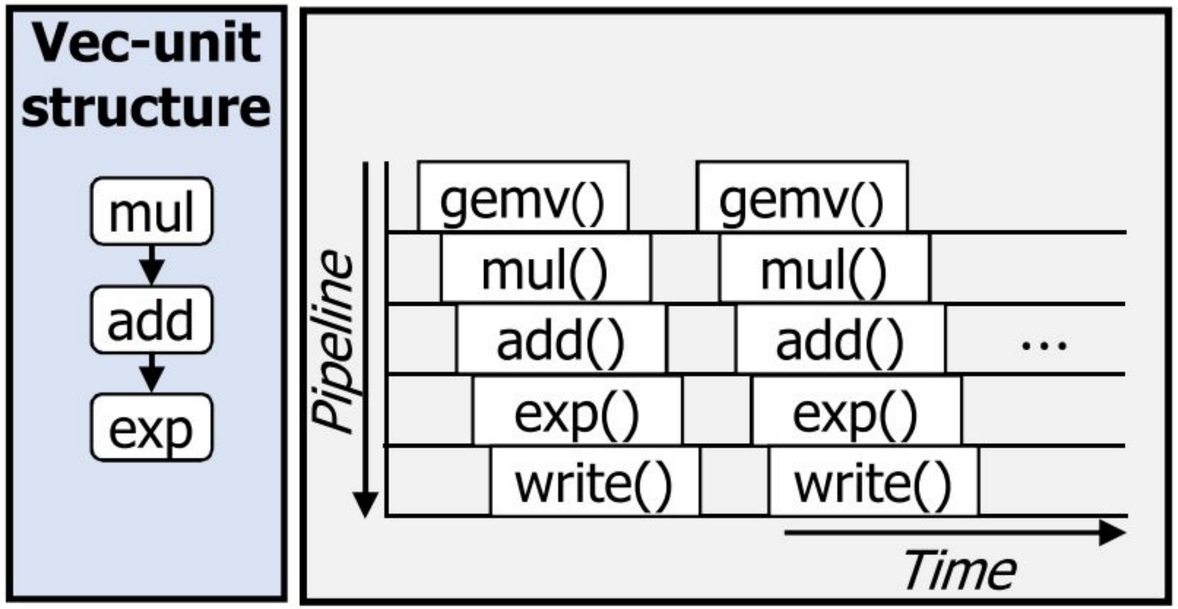
Hur, Suyeon, et al. A fast and flexible FPGA-based accelerator for natural language processing neural networks. ACM TACO 2023.

A Fast and Flexible FPGA-based Accelerator for Natural Language Processing Neural Networks

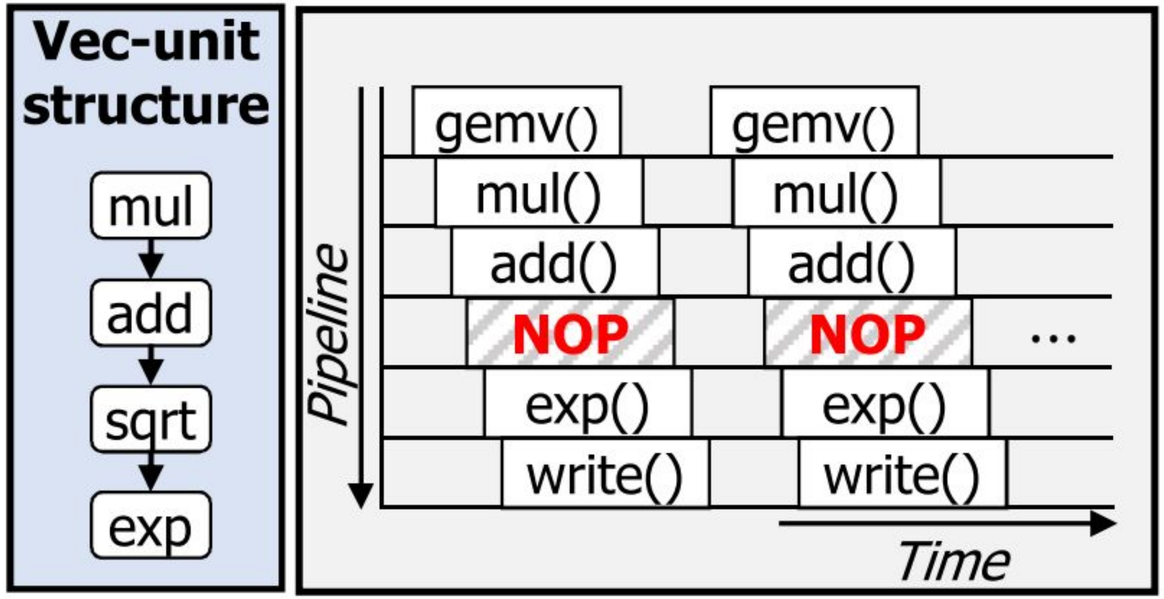


Hur, Suyeon, et al. A fast and flexible FPGA-based accelerator for natural language processing neural networks. ACM TACO 2023.

A Fast and Flexible FPGA-based Accelerator for Natural Language Processing Neural Networks



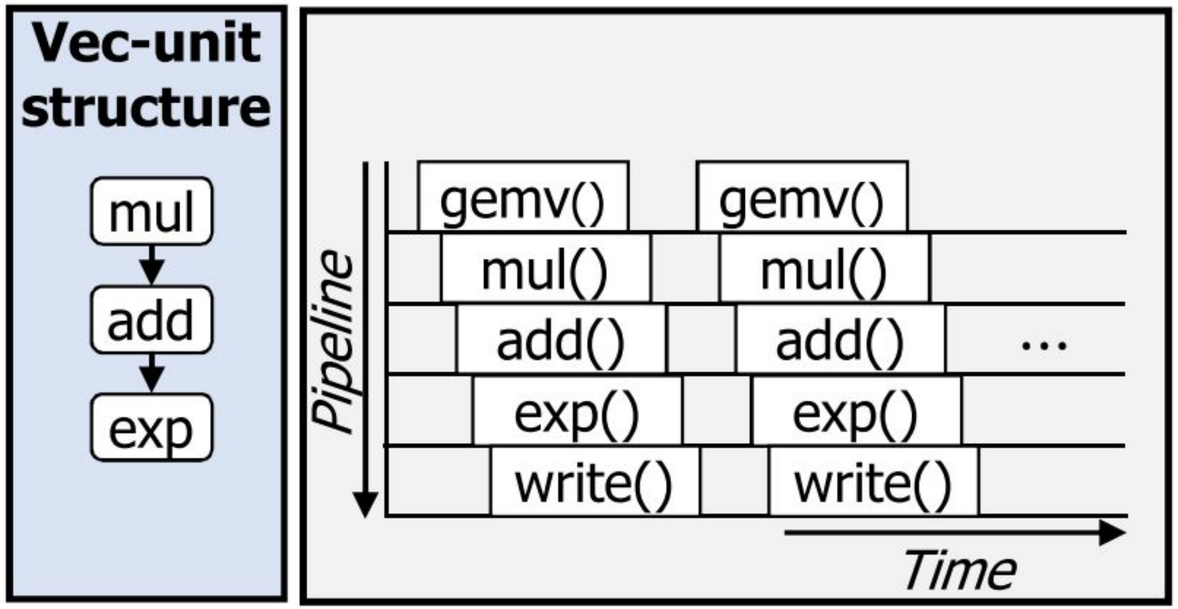
(a) Right types & order.



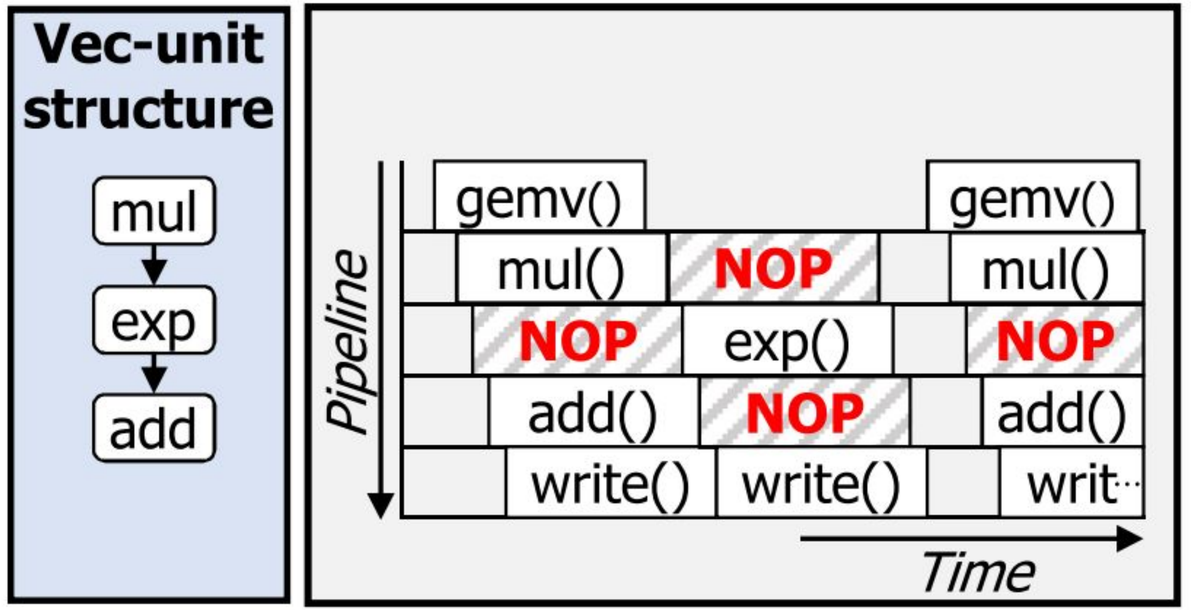
(b) Wrong types but right order.

Hur, Suyeon, et al. A fast and flexible FPGA-based accelerator for natural language processing neural networks. ACM TACO 2023.

A Fast and Flexible FPGA-based Accelerator for Natural Language Processing Neural Networks



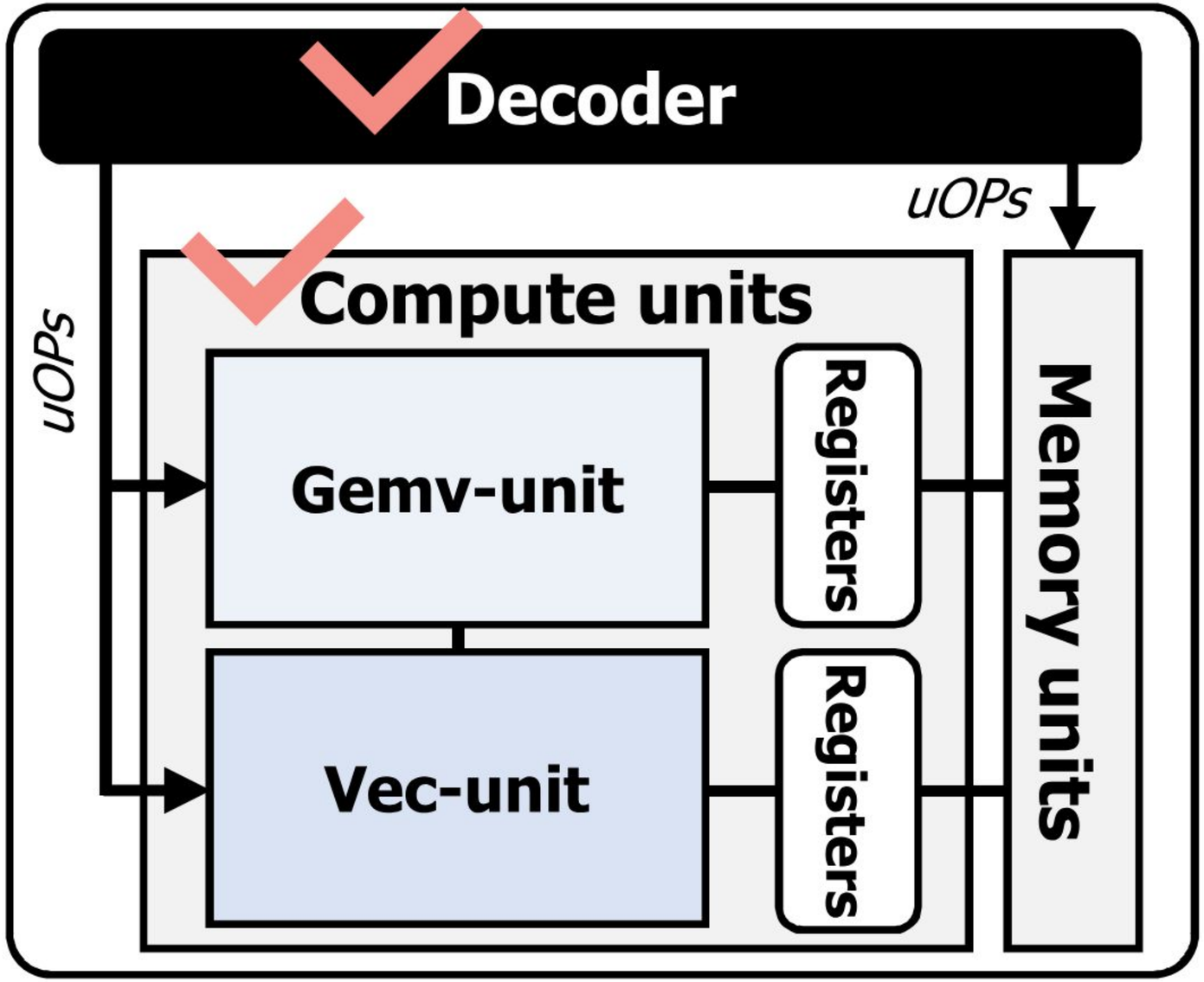
(a) Right types & order.



(c) Right types but wrong order.

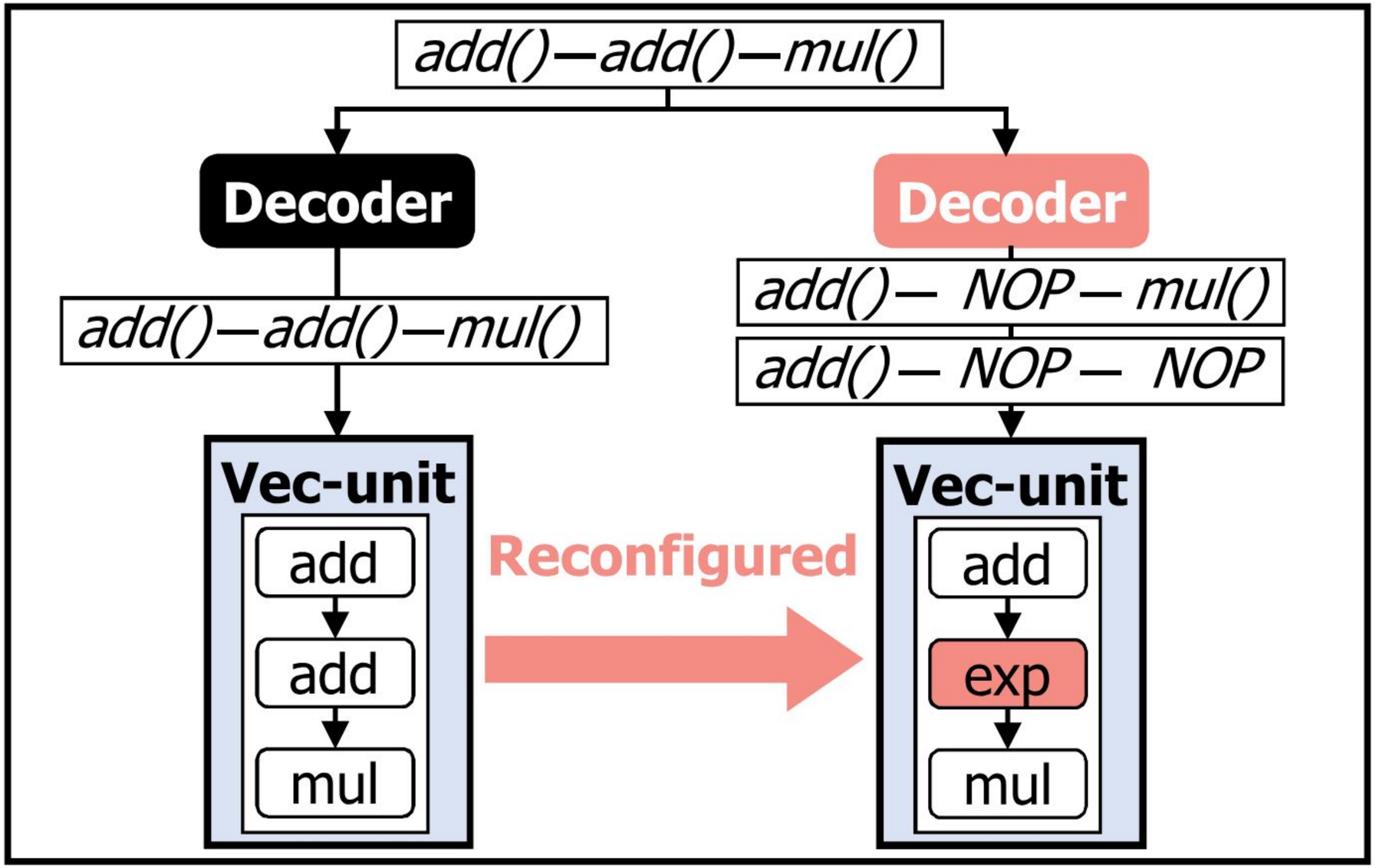
Hur, Suyeon, et al. A fast and flexible FPGA-based accelerator for natural language processing neural networks. ACM TACO 2023.

A Fast and Flexible FPGA-based Accelerator for Natural Language Processing Neural Networks



Hur, Suyeon, et al. A fast and flexible FPGA-based accelerator for natural language processing neural networks. ACM TACO 2023.

A Fast and Flexible FPGA-based Accelerator for Natural Language Processing Neural Networks



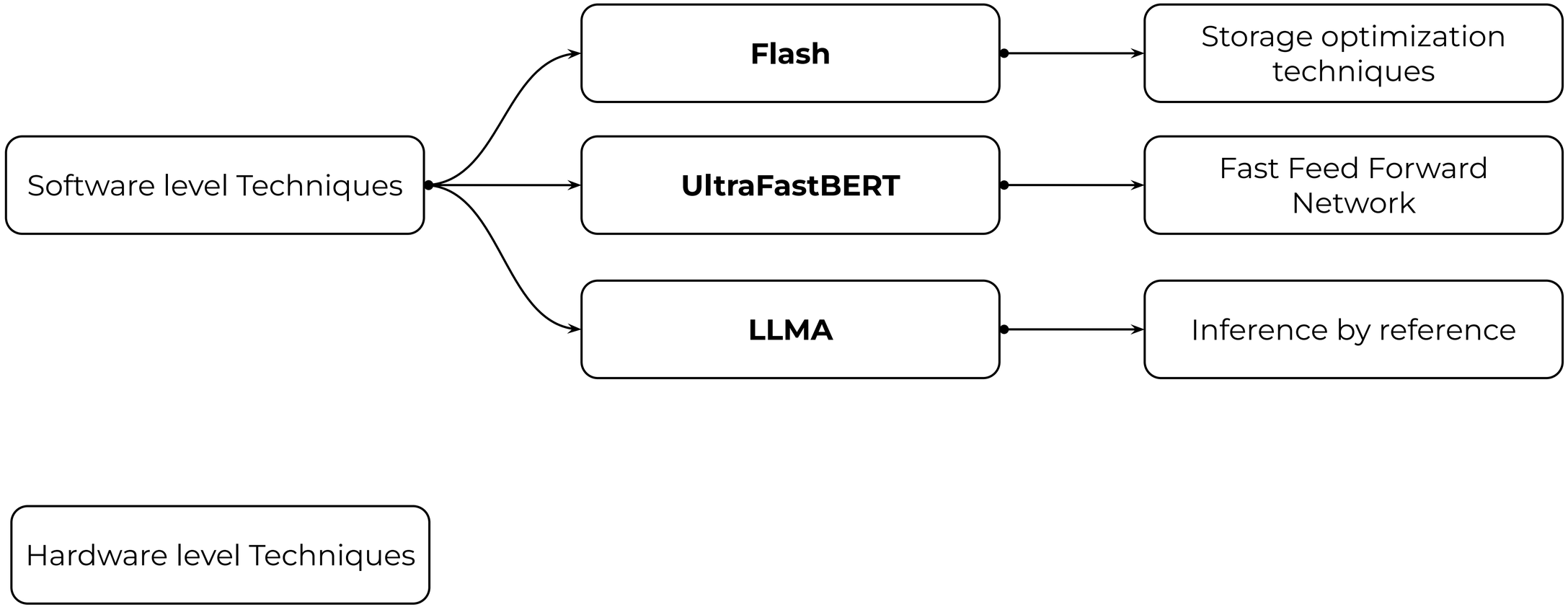
Hur, Suyeon, et al. A fast and flexible FPGA-based accelerator for natural language processing neural networks. ACM TACO 2023.

So far.

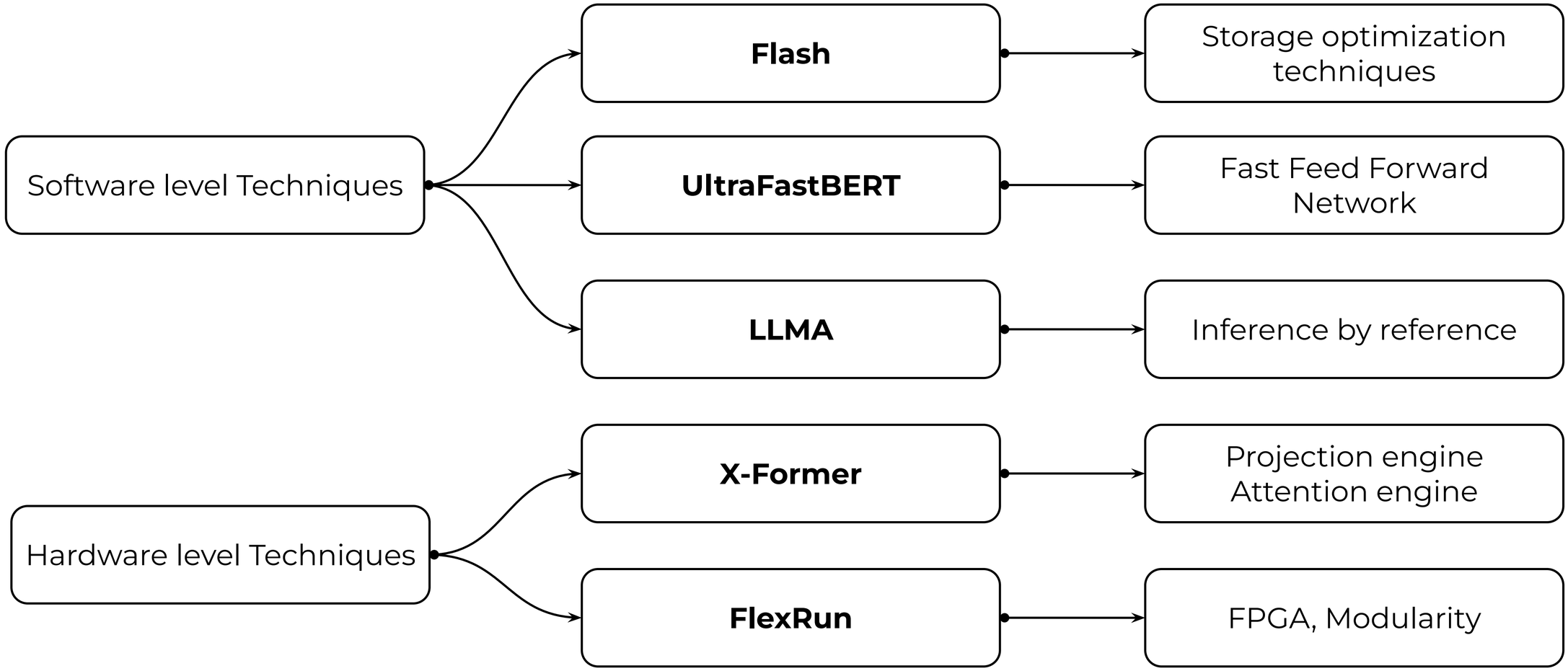
Software level Techniques

Hardware level Techniques

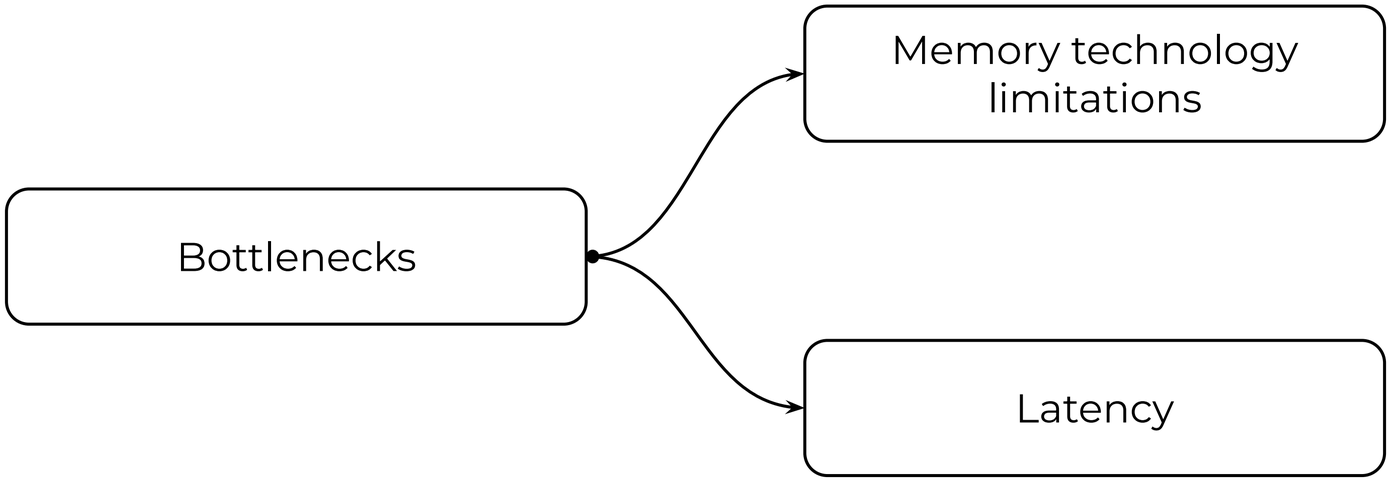
So far.


Software level Techniques
Flash
Storage optimization techniques
UltraFastBERT
Fast Feed Forward Network
LLMA
Inference by reference
Hardware level Techniques

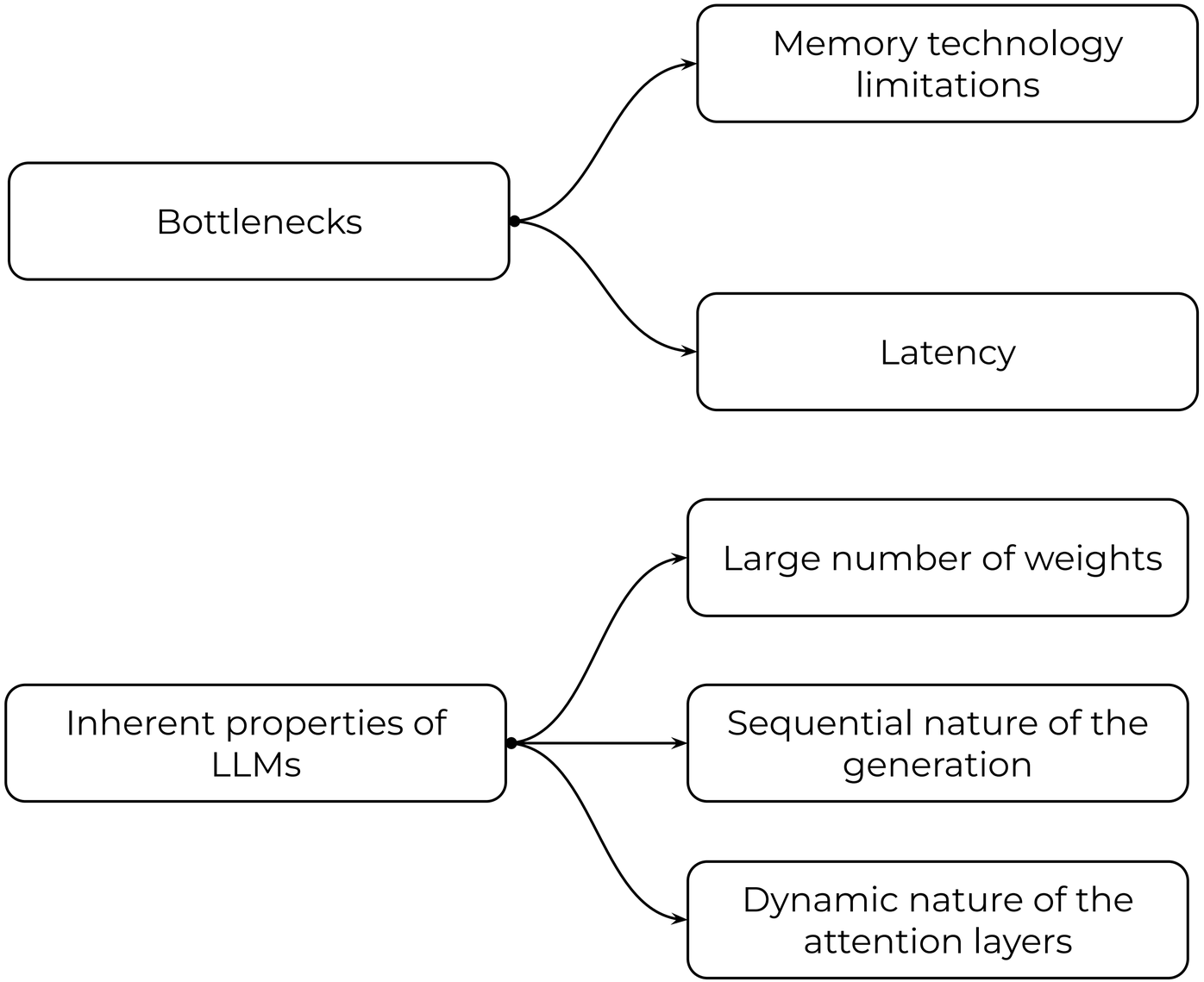
So far.


Software level Techniques
Flash
Storage optimization techniques
UltraFastBERT
Fast Feed Forward Network
LLMA
Inference by reference
Hardware level Techniques
X-Former
Projection engine
Attention engine
FlexRun
FPGA, Modularity

Comparative Analysis : Trends



Comparative Analysis : Trends



Comparative Analysis : Speedup

Year	Method	Optimization level	Technology	Baseline	Speedup	Energy efficiency
2023	Flash	Software	Flash	CPU/GPU	25x/5x	—
2023	UltraFastBERT	Software	CPU	CPU	78x	—
2023	LLMA	Software	GPU	GPU	2x	—
2023	X-Former	Hardware	In-memory	GPU	85x	7.5x
2023	FlexRun	Hardware	FPGA	GPU	2.7x	—

Comparative Analysis : Speedup

Year	Method	Optimization level	Technology	Baseline	Speedup	Energy efficiency
2023	Flash	Software	Flash	CPU/GPU	25x/5x	—
2023	UltraFastBERT	Software	CPU	CPU	78x	—
2023	LLMA	Software	GPU	GPU	2x	—
2023	X-Former	Hardware	In-memory	GPU	85x	7.5x
2023	FlexRun	Hardware	FPGA	GPU	2.7x	—

Interesting result given its a relatively simple software tweak compared to other methods.

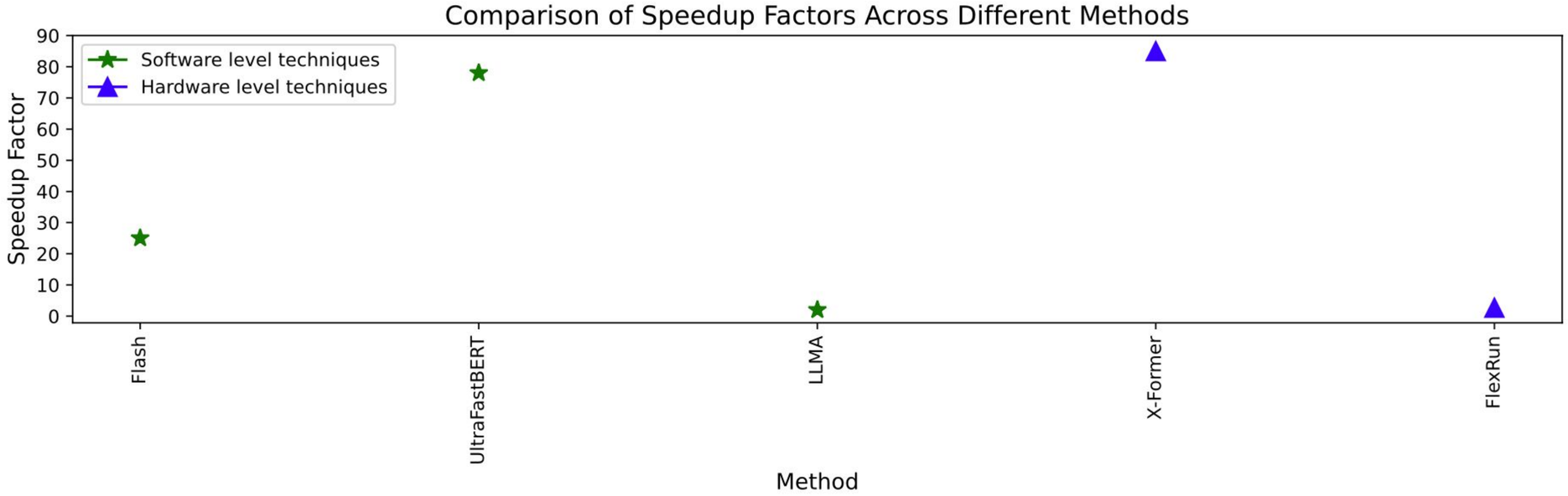
Comparative Analysis : Speedup

Year	Method	Optimization level	Technology	Baseline	Speedup	Energy efficiency
2023	Flash	Software	Flash	CPU/GPU	25x/5x	—
2023	UltraFastBERT	Software	CPU	CPU	78x	—
2023	LLMA	Software	GPU	GPU	2x	—
2023	X-Former	Hardware	In-memory	GPU	85x	7.5x
2023	FlexRun	Hardware	FPGA	GPU	2.7x	—

Comparative Analysis : Speedup

Year	Method	Optimization level	Technology	Baseline	Speedup	Energy efficiency
2023	Flash	Software	Flash	CPU/GPU	25x/5x	—
2023	UltraFastBERT	Software	CPU	CPU	78x	—
2023	LLMA	Software	GPU	GPU	2x	—
2023	X-Former	Hardware	In-memory	GPU	85x	7.5x
2023	FlexRun	Hardware	FPGA	GPU	2.7x	—

Despite higher initial costs, this reinforces the value of investing in specialized hardware solutions.



Comparative Analysis : Energy efficiency

Year	Method	Optimization level	Technology	Baseline	Speedup	Energy efficiency
2023	Flash	Software	Flash	CPU/GPU	25x/5x	—
2023	UltraFastBERT	Software	CPU	CPU	78x	—
2023	LLMA	Software	GPU	GPU	2x	—
2023	X-Former	Hardware	In-memory	GPU	85x	7.5x
2023	FlexRun	Hardware	FPGA	GPU	2.7x	—

1. Without concrete numerical data, we refrain from drawing definitive conclusions about the energy efficiency improvements of these methods.
2. We acknowledge this gap in the analysis and suggest that future research should aim to provide detailed energy consumption metrics to fully assess the impact of these optimization strategies.

Conclusion

1. Significant lack of synergy between software and hardware optimization methods.

Conclusion

1. Significant lack of synergy between software and hardware optimization methods.
2. Strategic hardware improvements can yield significant efficiency benefits even in the face of substantial initial investments.

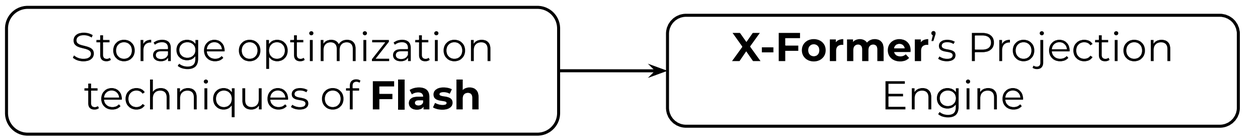
Conclusion

1. Significant lack of synergy between software and hardware optimization methods.
2. Strategic hardware improvements can yield significant efficiency benefits even in the face of substantial initial investments.
3. We acknowledge this gap in the analysis and suggest that future research should aim to provide detailed energy consumption metrics to fully assess the impact of these optimization strategies.

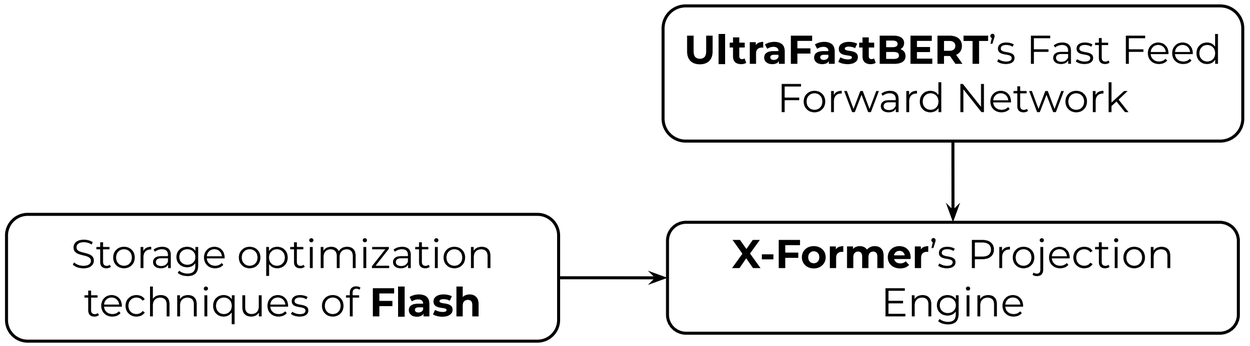
Future works that we thought of

X-Former’s Projection Engine

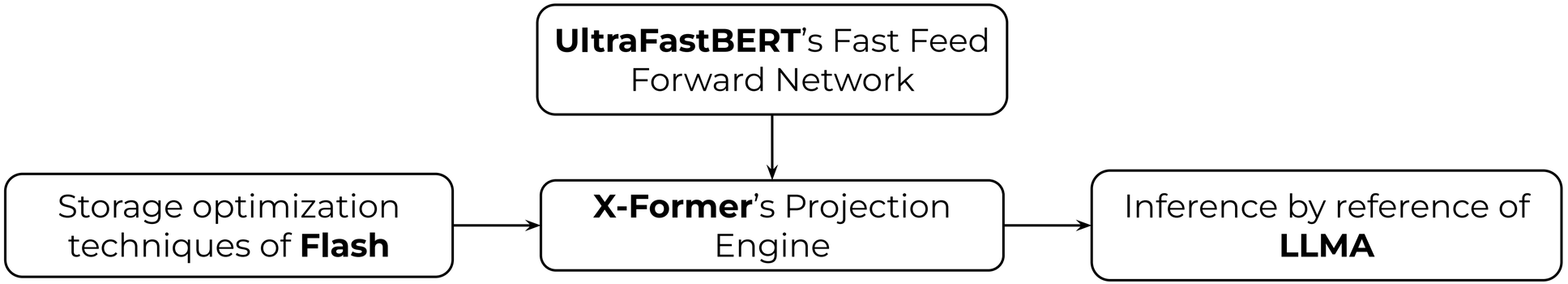
Future works that we thought of



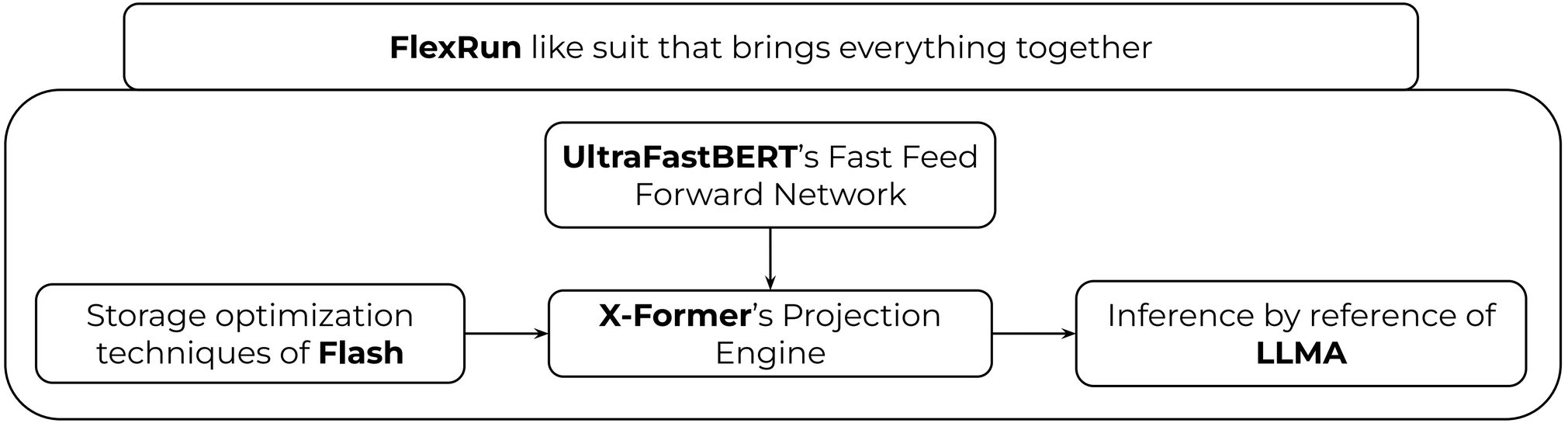
Future works that we thought of



Future works that we thought of



Future works that we thought of



Suggested direction for the community

- Potential for a holistic integration of optimization techniques appears to be a glaring omission in current research efforts and represents a promising direction for future exploration.

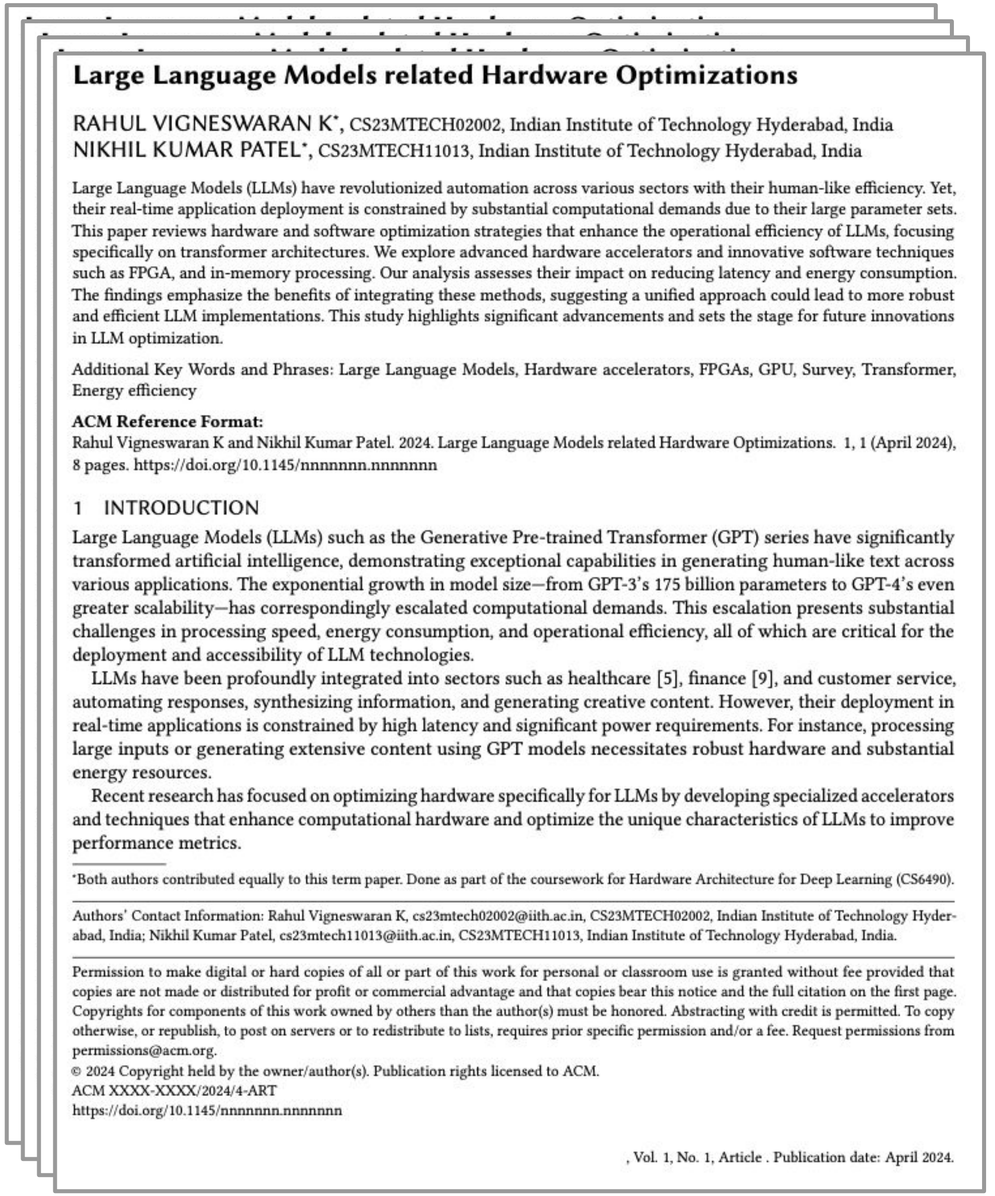
- Mandate to provide detailed energy consumption metrics.

Citations

- Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/
- Alizadeh-Vahid, Keivan et al. LLM in a flash: Efficient Large Language Model Inference with Limited Memory. ArXiv 2023.
- Mirzadeh, Iman, et al. Relu strikes back: Exploiting activation sparsity in large language models. arXiv 2023.
- Belcak, Peter, and Roger Wattenhofer. Fast feedforward networks. arXiv 2023.
- Belcak, Peter, and Roger Wattenhofer. Exponentially Faster Language Modelling. arXiv 2023.
- Yang, Nan, et al. Inference with reference: Lossless acceleration of large language models. arXiv 2023.
- Sridharan, Shrihari, et al. X-former: In-memory acceleration of transformers. IEEE T-VLSI 2023.
- Hur, Suyeon, et al. A fast and flexible FPGA-based accelerator for natural language processing neural networks. ACM T-ACO 2023.

Rahul Vigneswaran K

Nikhil Kumar Patel



Large Language Models related Hardware Optimizations

RAHUL VIGNESWARAN K*, CS23MTECH02002, Indian Institute of Technology Hyderabad, India
NIKHIL KUMAR PATEL*, CS23MTECH11013, Indian Institute of Technology Hyderabad, India

Large Language Models (LLMs) have revolutionized automation across various sectors with their human-like efficiency. Yet, their real-time application deployment is constrained by substantial computational demands due to their large parameter sets. This paper reviews hardware and software optimization strategies that enhance the operational efficiency of LLMs, focusing specifically on transformer architectures. We explore advanced hardware accelerators and innovative software techniques such as FPGA, and in-memory processing. Our analysis assesses their impact on reducing latency and energy consumption. The findings emphasize the benefits of integrating these methods, suggesting a unified approach could lead to more robust and efficient LLM implementations. This study highlights significant advancements and sets the stage for future innovations in LLM optimization.

Additional Key Words and Phrases: Large Language Models, Hardware accelerators, FPGAs, GPU, Survey, Transformer, Energy efficiency

ACM Reference Format:
Rahul Vigneswaran K and Nikhil Kumar Patel. 2024. Large Language Models related Hardware Optimizations. 1, 1 (April 2024), 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Large Language Models (LLMs) such as the Generative Pre-trained Transformer (GPT) series have significantly transformed artificial intelligence, demonstrating exceptional capabilities in generating human-like text across various applications. The exponential growth in model size—from GPT-3's 175 billion parameters to GPT-4's even greater scalability—has correspondingly escalated computational demands. This escalation presents substantial challenges in processing speed, energy consumption, and operational efficiency, all of which are critical for the deployment and accessibility of LLM technologies.

LLMs have been profoundly integrated into sectors such as healthcare [5], finance [9], and customer service, automating responses, synthesizing information, and generating creative content. However, their deployment in real-time applications is constrained by high latency and significant power requirements. For instance, processing large inputs or generating extensive content using GPT models necessitates robust hardware and substantial energy resources.

Recent research has focused on optimizing hardware specifically for LLMs by developing specialized accelerators and techniques that enhance computational hardware and optimize the unique characteristics of LLMs to improve performance metrics.

*Both authors contributed equally to this term paper. Done as part of the coursework for Hardware Architecture for Deep Learning (CS6490).

Authors' Contact Information: Rahul Vigneswaran K, cs23mtech02002@iith.ac.in, CS23MTECH02002, Indian Institute of Technology Hyderabad, India; Nikhil Kumar Patel, cs23mtech11013@iith.ac.in, CS23MTECH11013, Indian Institute of Technology Hyderabad, India.

ACM XXXX-XXXX/2024/4-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn