



Mokey

Enabling Narrow Fixed-Point Inference for Out-of-the-Box Floating-Point Transformer Models

49th IEEE/ACM International Symposium on Computer Architecture
(ISCA '22)



Ali Hadi Zadeh^{1,2}, Mostafa Mahmoud¹, Ameer Abdelhadi¹, and Andreas Moshovos^{1,2}

¹ University of Toronto, ² Vector Institute



Transformers for Text Generation

InferKit DEMO 9210 / 10000 weekly free characters [Sign In](#)

Generate Options

Learn more in [the docs](#).

Length to generate ?

200

Start at beginning ?

[Advanced Settings »](#)

My name is Ali. I am a PhD candidate at UofT. Today I am presenting my paper at the 49th IEEE/ACM International Symposium on Computer Architecture (ISCA '22).

[Generate Text](#) ✕

<https://app.inferkit.com/demo>



Transformers for Text Generation

InferKit DEMO 9210 / 10000 weekly free characters [Sign In](#)

Generate Options

Learn more in [the docs](#).

Length to generate ?

200

Start at beginning ?

[Advanced Settings »](#)

My name is Ali. I am a PhD candidate at UofT. Today I am presenting my paper at the 49th IEEE/ACM International Symposium on Computer Architecture (ISCA '22). **To tell you the truth, I'm nervous, but excited to show off my ideas!**

[Generate Text](#) ✕

<https://app.inferkit.com/demo>

Challenges

Weights



2018

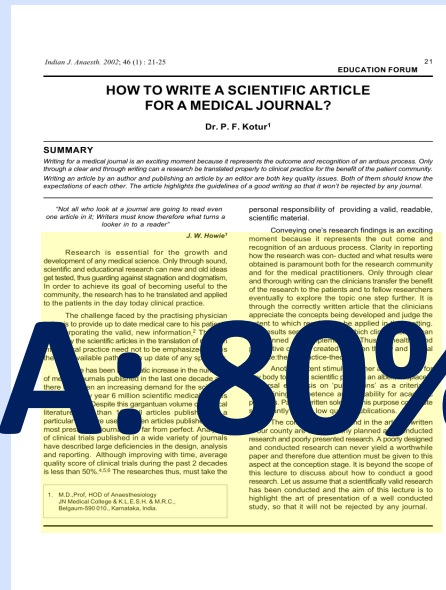
1.2GB



2021

2TB

Activations



Challenges

Weights

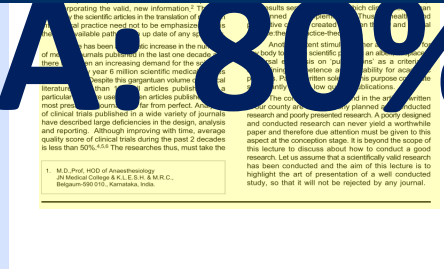
Activations

Memory: Performance & Energy Bottleneck



**2021
2TB**

A: 80%



Challenges

Weights



2018

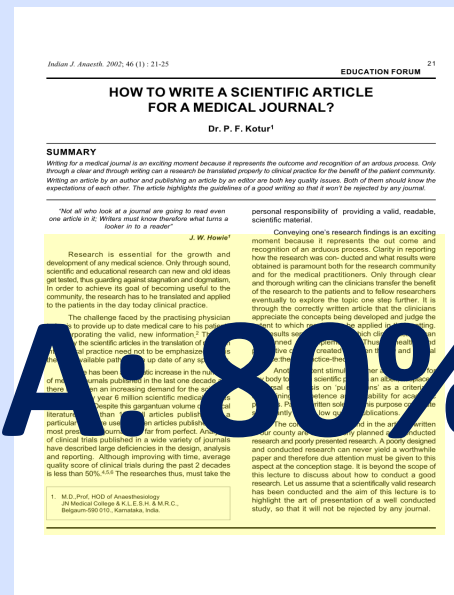
1.2GB



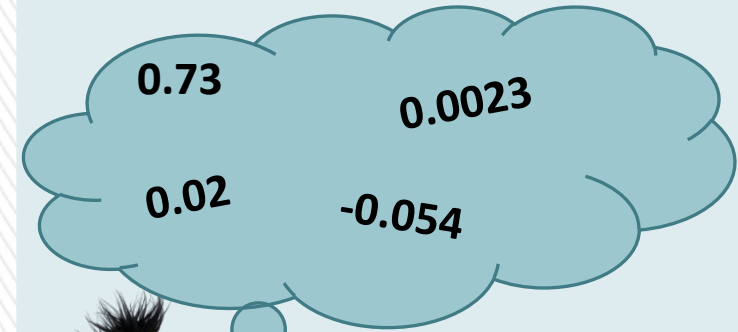
2021

2TB

Activations



FP Compute



~100T FP MACs

Mokey: BERT's Better Self

0.02 0.73
-0.054 0.0023

Floating point



0001 0101
1010 1100



4b Int index

8x vs FP32

4x vs FP16

*Not your typical 4b quantization ;)



Mokey



4-bit Quantization: W+A

$$W, A = f(idx)$$

Index	Value
...	...
...	...



$+ = A \times W. \rightarrow$ Count *idx*

Post-training



Fixed-point compute

Mokey HW Accelerator

Vs. Tensor Cores: **15x** Faster + **100x** Energy Efficient



Mokey Memory Compression

For Tensor Cores:

Off-chip Only: **4x** Faster + **8x** Energy Efficient

Off- and on-chip*: **10x** Faster + **50x** Energy Efficient

Roadmap



Mokey
Quantization



Computation
on indices



Models'
Accuracy

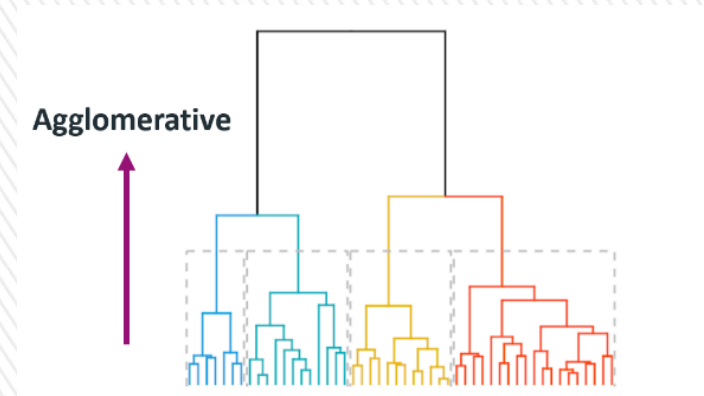
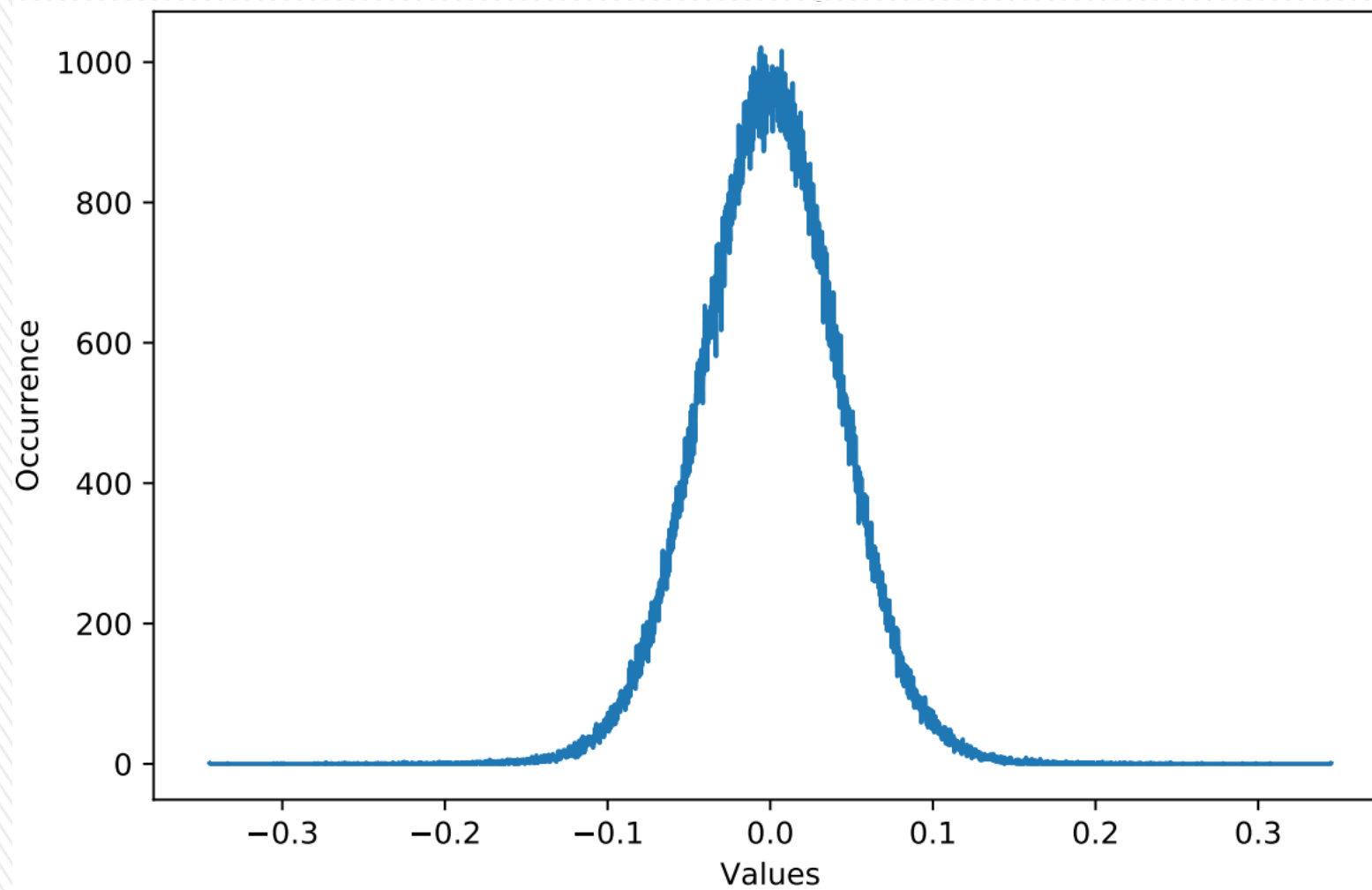


Hardware
Evaluation



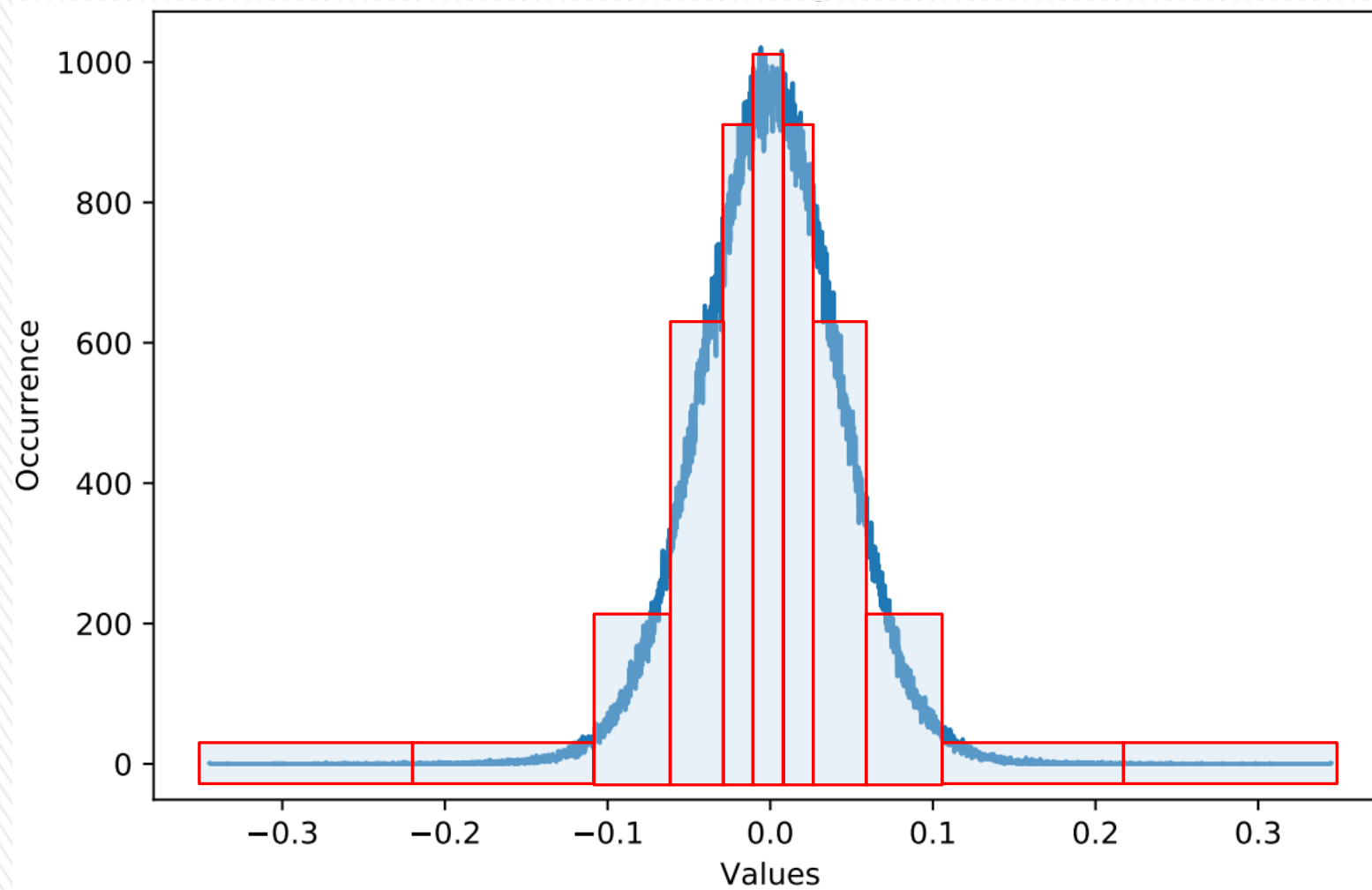
Conclusion

Dictionary-Based Quantization





Dictionary-Based Quantization



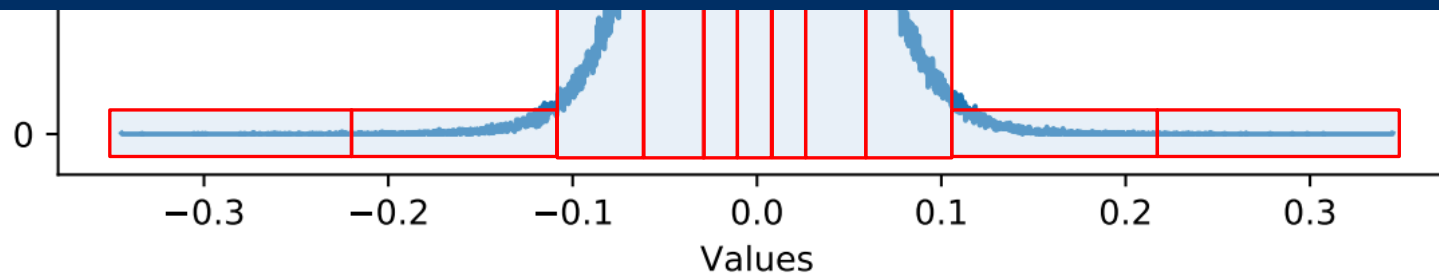
Quantization Dict.

Index	Value
I	0.02
II	0.07
III	0.12
IV	0.25
...	...

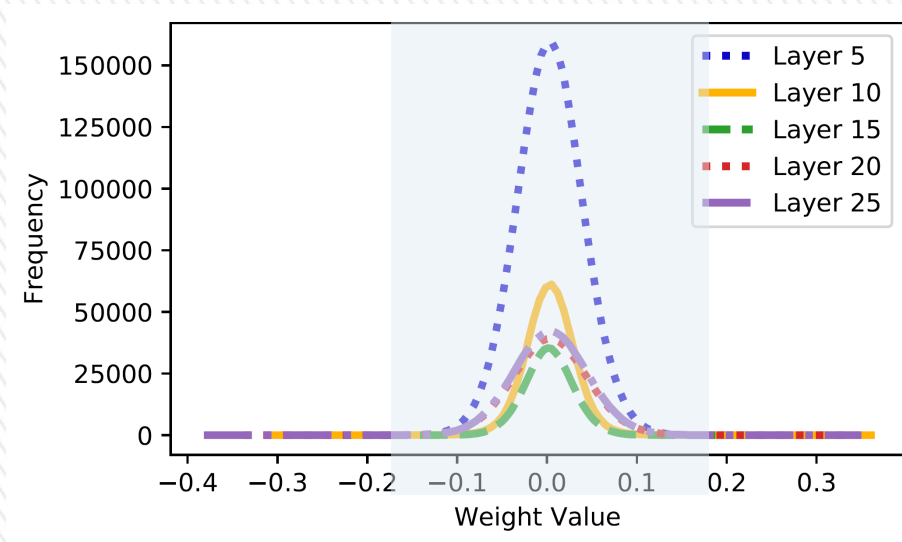
Dictionary-Based Quantization



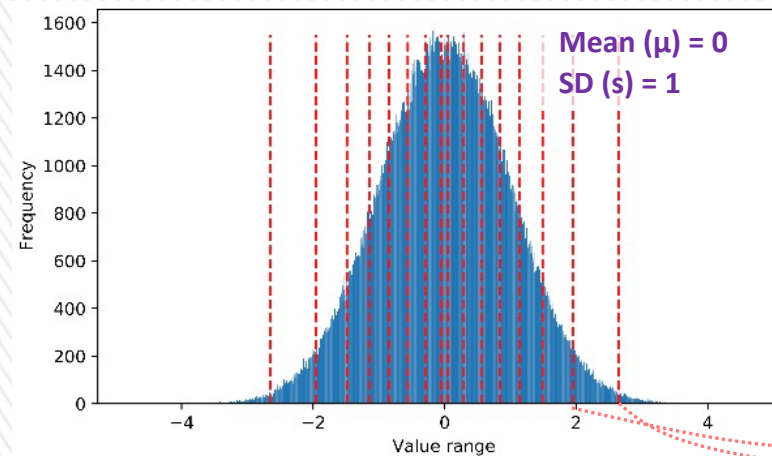
Clustering: Iterative ☹️
Not Feasible for Activations



A Dictionary for All Layers



Reference Distribution



Golden Dict. (GD)

Index	Value
I	-2.7
II	-1.98
...	...
VI	1.98
XVI	2.7

Scale and Shift is All You Need!



A Dictionary for All Layers

Reference Distribution

**Weights & Embeddings: Offline
Activation: Profiling**

Scale and Shift is All You Need!

Inference Computation

Original

$$A = 0.2 \quad W = 0.7$$

$$A \times W += 0.2 \times 0.7 = 0.14$$

Dictionary Quant.

$$A = I \quad W = II$$

Index	Value
I	0.2
II	0.7
III	1.1
IV	1.4
...	...

$$A \times W += I \times II$$

$$A \times W += 0.2 \times 0.7 = 0.14$$

Mokey Quant.

$$A = I \quad W = II$$



$$A \times W += I \times II = 0.14$$



How to Use Indices for Computation

Original

Dictionary Quant.

Mokey Quant.

Let's See in Practice!

$$A \times W += I \times II$$

$$A \times W += 0.2 \times 0.7 = 0.14$$

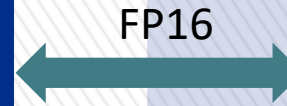
FP16 Baseline

Original

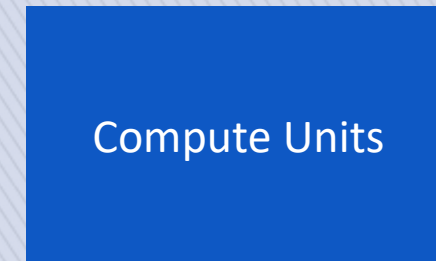
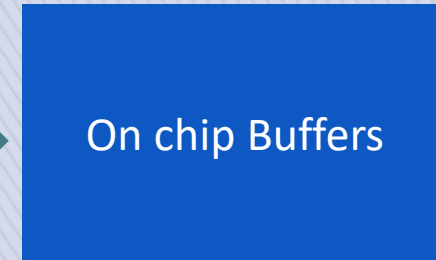
$$A = 0.2$$

$$W = 0.7$$

$$A \times W = 0.2 \times 0.7 = 0.14$$



Chip



Dictionary Quantization

Dictionary Quant.

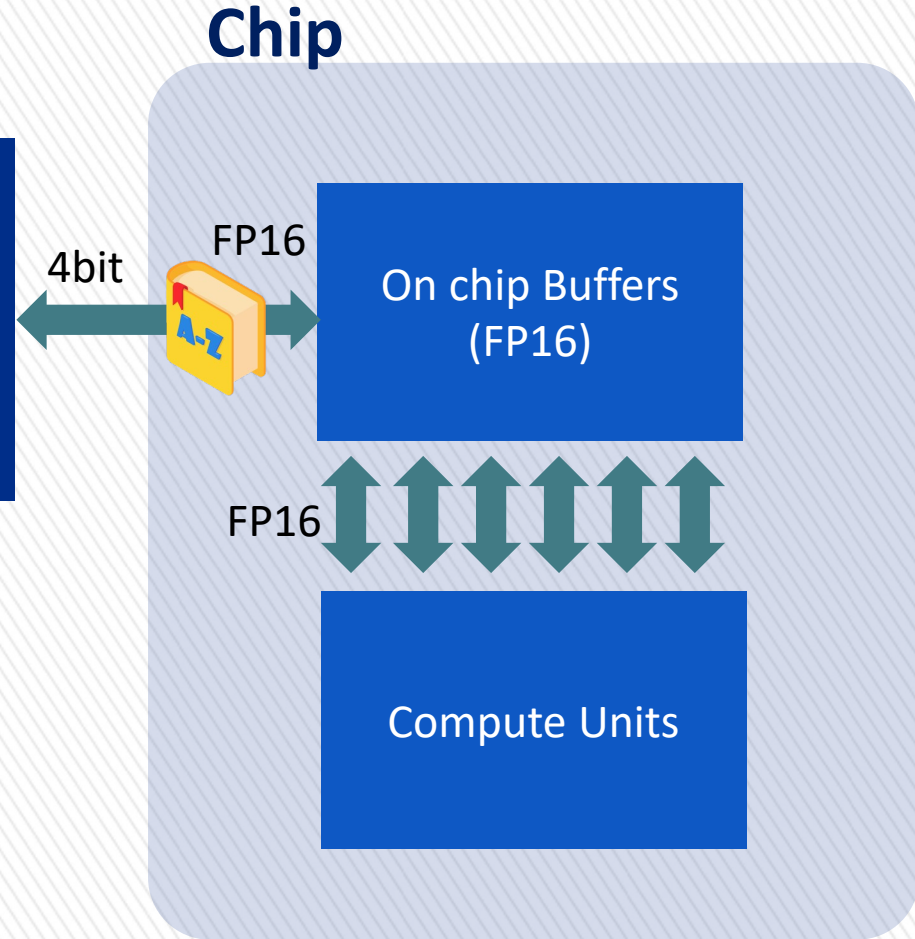
$A = I$

$W = II$

Index	Value
I	0.2
II	0.7
III	1.1
IV	1.4
...	...

$$A \times W += I \times II$$

$$A \times W += 0.2 \times 0.7 = 0.14$$





Dictionary Quantization

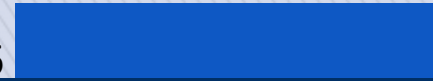
Dictionary Quant.

$A = I$ $W = II$



Chip

FP16



No on chip mem comp
Limited performance/energy gain

$$A \times W += I \times II$$

$$A \times W += 0.2 \times 0.7 = 0.14$$

Dictionary Quantization

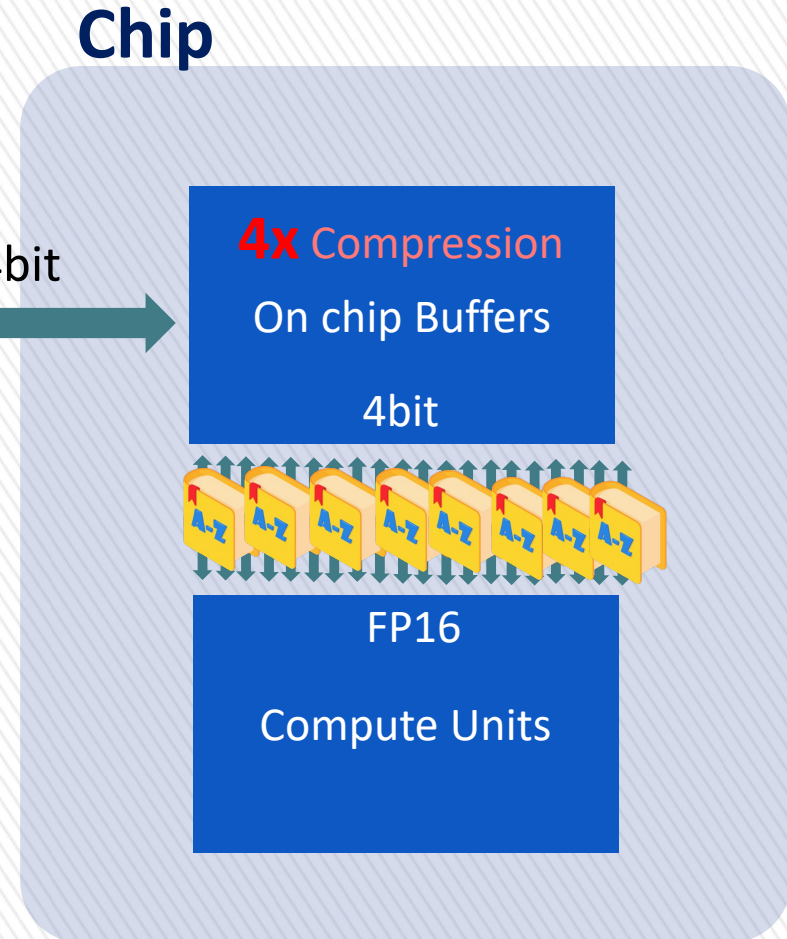
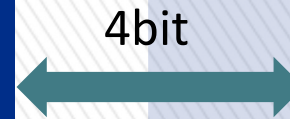
Dictionary Quant.

$A = I$ $W = II$

Index	Value
I	0.2
II	0.7
III	1.1
IV	1.4
...	...

$$A \times W += I \times II$$

$$A \times W += 0.2 \times 0.7 = 0.14$$





Dictionary Quantization

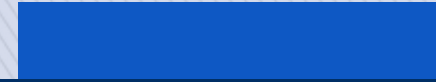
Dictionary Quant.

$A = I$ $W = II$



4bit

Chip



Many instances of LUT
Waste of Area & Energy ☹️

$$A \times W \approx I \times II$$

$$A \times W \approx 0.2 \times 0.7 = 0.14$$

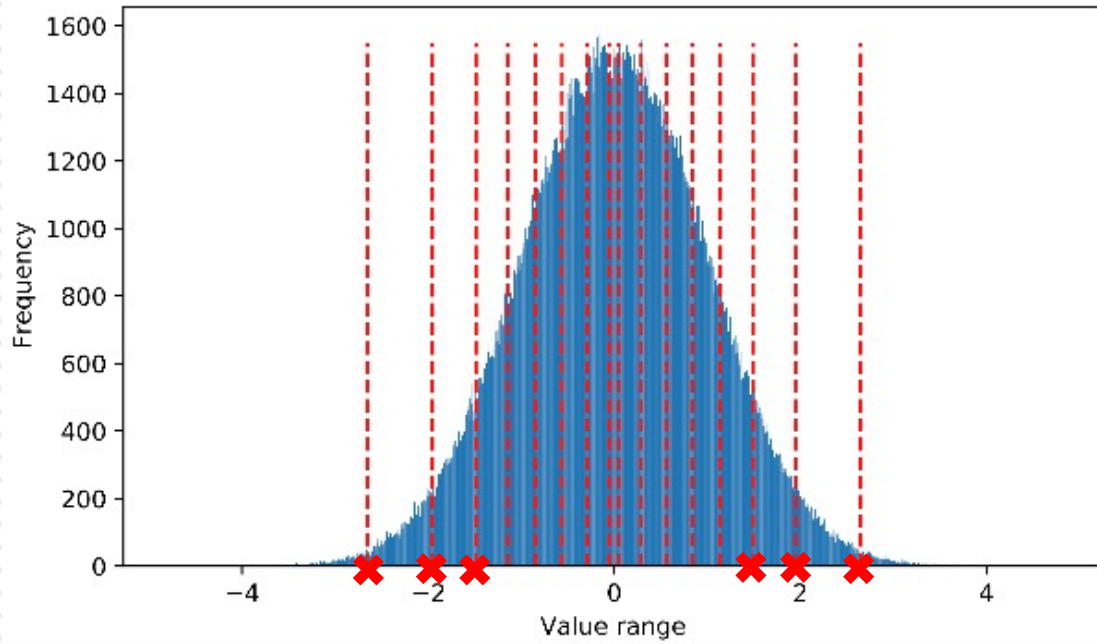
Mokey Quantization



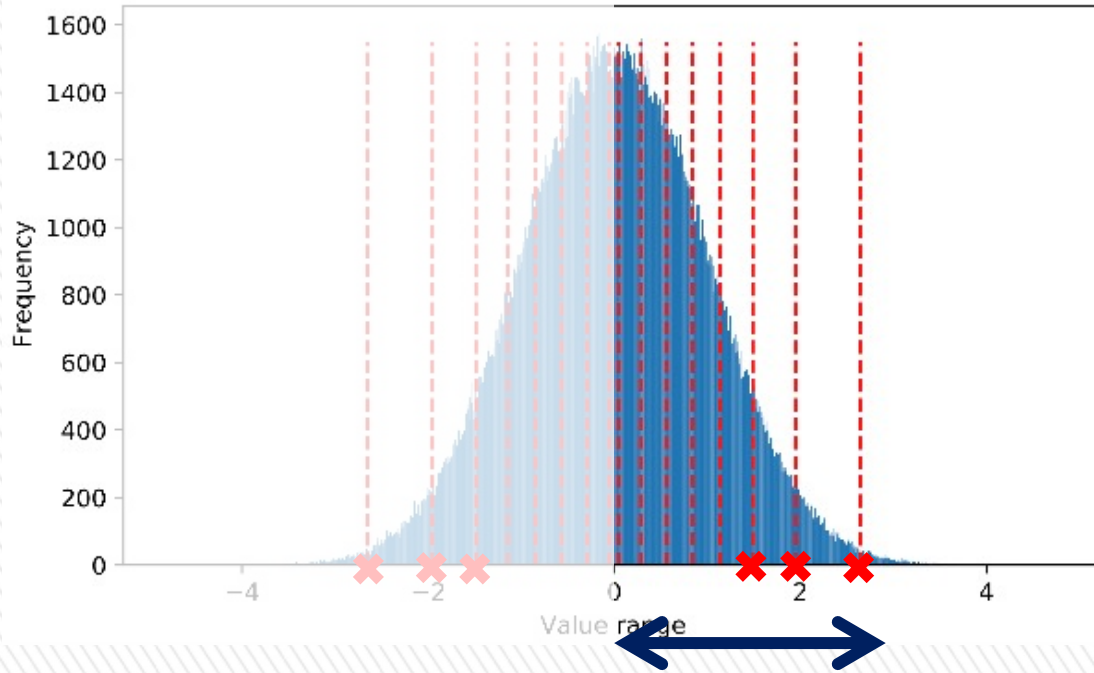
“Simple” Relationship between Index and Value



Mokey Quantization



Symmetrical Dictionary

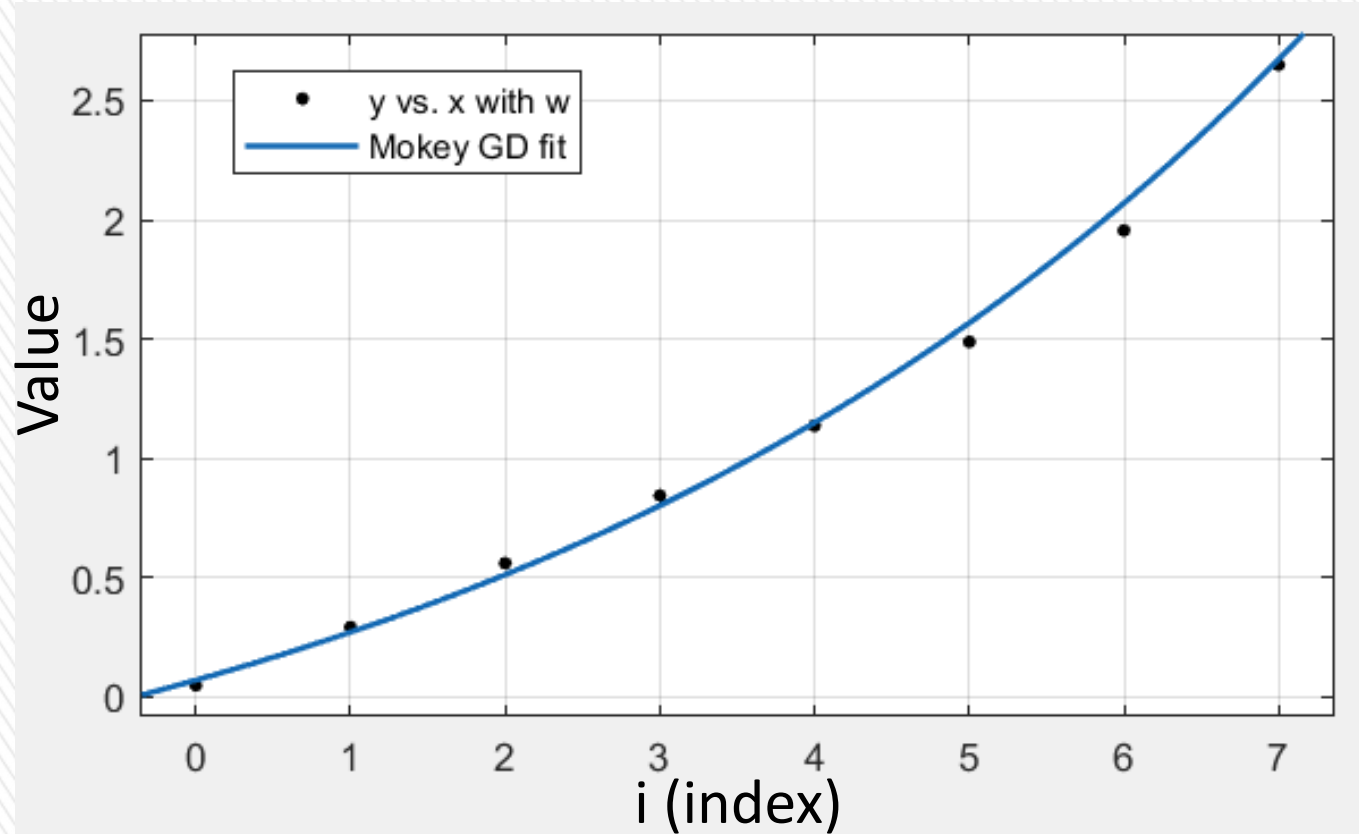


Index	Value
I	0.05
II	0.35
...	...
VI	1.97
VII	2.61

Golden Dict. (GD)

Exponential Function

Index	Value
I	0.05
II	0.35
...	...
VI	1.97
VII	2.6



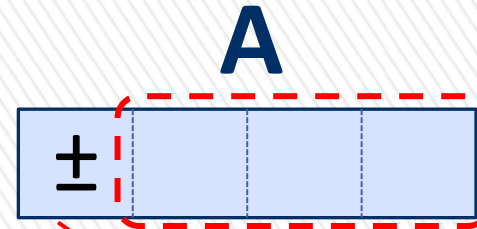
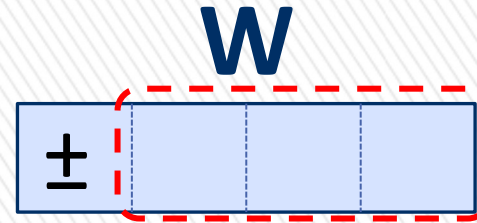
Golden Dict. (GD)

$$GD = a^i + b$$

Values Format

Index	Value
I	0.05
II	0.35
...	...
VI	1.97
VII	2.6

Golden Dict. (GD)



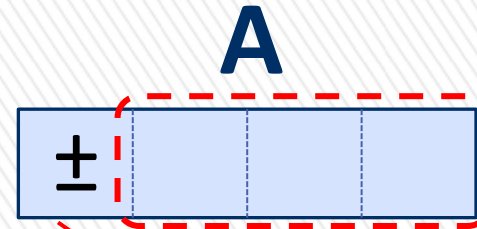
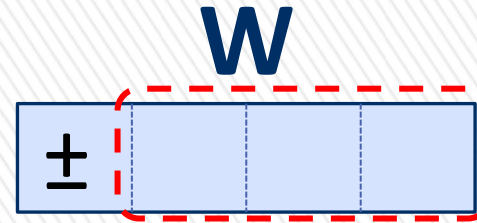
$$\text{Values} = \pm (a^i + b)$$

per value Fixed

Values Format

Index	Value
I	0.05
II	0.35
...	...
VI	1.97
VII	2.6

Golden Dict. (GD)



$$\text{Values} = \pm (a^i + b) \times s + \mu$$

per value Fixed Per layer



Revisiting Computation

$$\begin{array}{ll} A_0 = a^1 + b & W_0 = a^2 + b \\ A_1 = a^4 + b & W_1 = a^5 + b \\ \dots & \dots \end{array}$$



Revisiting Computation

$$\begin{array}{ll} A_0 = a^1 + b & W_0 = a^2 + b \\ A_1 = a^4 + b & W_1 = a^5 + b \\ \dots & \dots \end{array}$$

$$\begin{aligned} \sum_N AW &= A_0W_0 + A_1W_1 + \dots \\ &= a^{1+2} + ba^1 + ba^2 + b^2 + \\ &\quad a^{4+5} + ba^4 + ba^5 + b^2 + \dots \end{aligned}$$



Revisiting Computation

$$\begin{array}{ll}
 A_0 = a^1 + b & W_0 = a^2 + b \\
 A_1 = a^4 + b & W_1 = a^5 + b \\
 \dots & \dots
 \end{array}$$

$$\begin{aligned}
 \sum_N AW &= A_0W_0 + A_1W_1 + \dots \\
 &= a^{1+2} + ba^1 + ba^2 + b^2 + \\
 &\quad a^{4+5} + ba^4 + ba^5 + b^2 + \\
 &\quad \dots
 \end{aligned}$$

Range [0,7]

$$= (a^3 + a^9 + \dots) + b(a^1 + a^4 + \dots) + \underbrace{b(a^2 + a^5 + \dots)}_{\text{Pre-computed}} + Nb^2$$



Revisiting Computation

$$\begin{array}{ll}
 A_0 = a^1 + b & W_0 = a^2 + b \\
 A_1 = a^4 + b & W_1 = a^5 + b \\
 \dots & \dots
 \end{array}$$

$$\begin{aligned}
 \sum_N AW &= A_0W_0 + A_1W_1 + \dots \\
 &= a^{1+2} + ba^1 + ba^2 + b^2 + \\
 &\quad a^{4+5} + ba^4 + ba^5 + b^2 + \\
 &\quad \dots \\
 &= (a^3 + a^9 + \dots) + \underbrace{b(a^1 + a^4 + \dots)}_{\text{Computed during last layer's quantization}} + \underbrace{b(a^2 + a^5 + \dots)}_{\text{Pre-computed}} + Nb^2
 \end{aligned}$$

Range [0,7]



Revisiting Computation

$$\begin{array}{ll}
 A_0 = a^1 + b & W_0 = a^2 + b \\
 A_1 = a^4 + b & W_1 = a^5 + b \\
 \dots & \dots
 \end{array}$$

$$\begin{aligned}
 \sum_N AW &= A_0W_0 + A_1W_1 + \dots \\
 &= a^{1+2} + ba^1 + ba^2 + b^2 + \\
 &\quad a^{4+5} + ba^4 + ba^5 + b^2 +
 \end{aligned}$$

Range [0,7]+[0,7]=[0,14] ...

Range [0,7]

$$= \underbrace{(a^3 + a^9 + \dots)}_{(1) \text{ Histogram}} + b \underbrace{(a^1 + a^4 + \dots)}_{\text{Computed during last layer's quantization}} + b \underbrace{(a^2 + a^5 + \dots)}_{\text{Pre-computed}} + Nb^2$$

(1) Histogram
(2) Weighted reduction

Computed during last layer's quantization

Pre-computed



Revisiting Computation

Histogram: Most of the computation => 3-bit INT Add

Reduction: Post processing => 16-bit Fixed-point

Range $[0,7]+[0,7]=[0,14]$...

$$= \underbrace{(a^3+a^9+\dots)}_{(1) \text{ Histogram}} + b \underbrace{(a^1+a^4+\dots)}_{\text{Computed during last layer's quantization}} + b \underbrace{(a^2+a^5+\dots)}_{\text{Pre-computed}} + Nb^2$$

(1) Histogram
(2) Weighted reduction

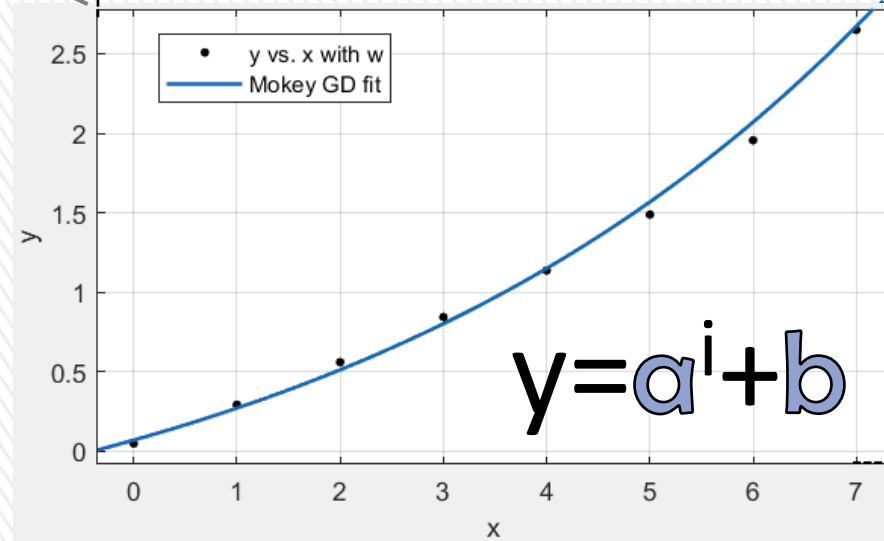
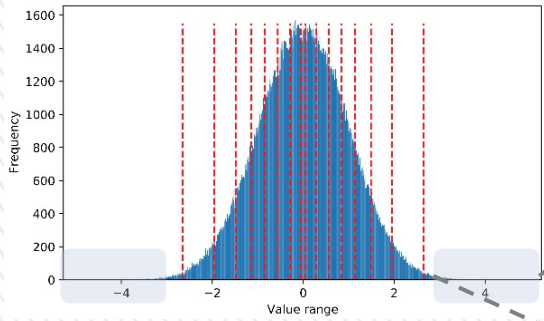
Computed during last layer's quantization

Pre-computed



Approach

One ~~Size~~ Fits All



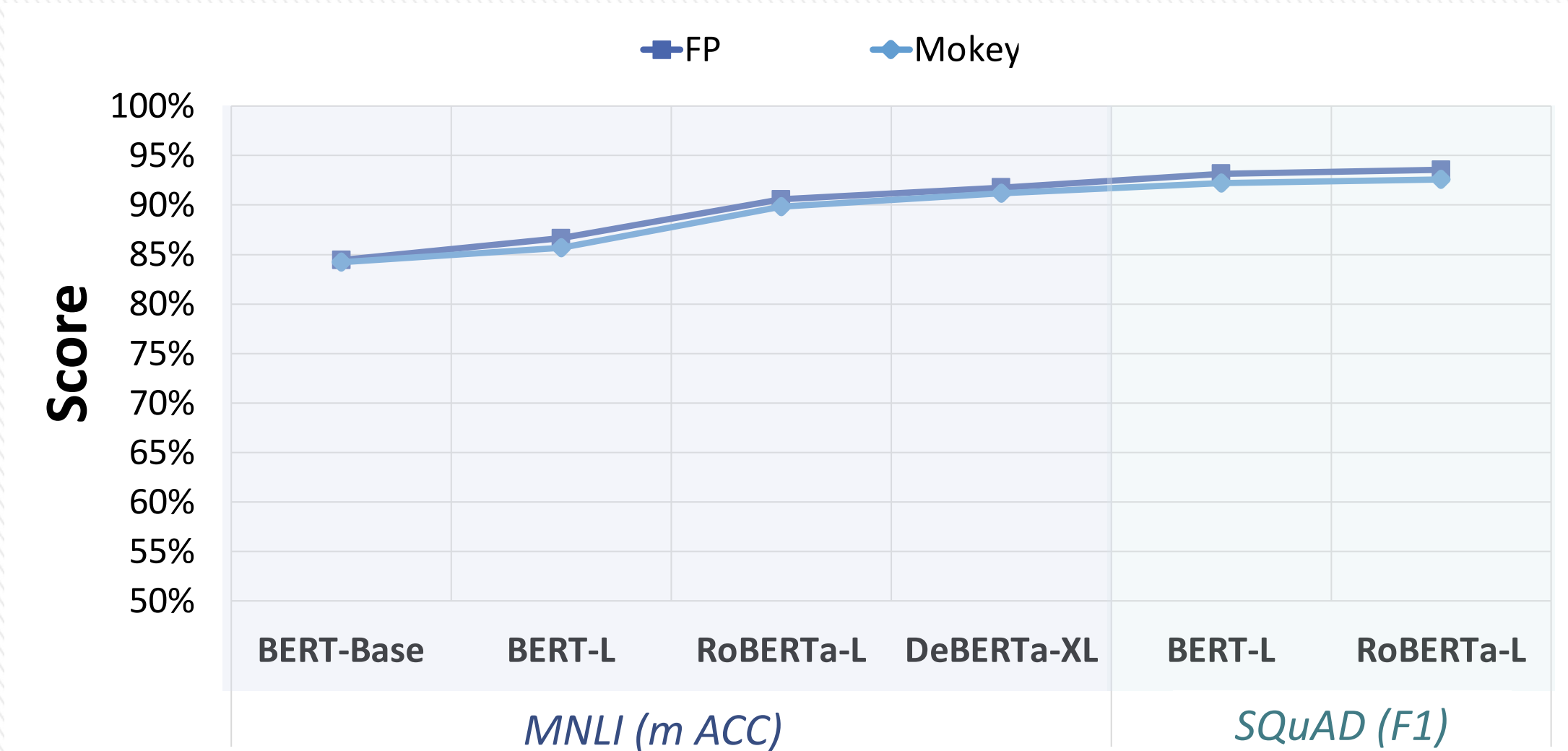


Evaluation

- FP16 Tensor Cores baseline
- Wide range of on-chip buffers
- 110M - 750M parameter models
- Custom cycle accurate simulator.
 - DRAMsim3: Dual Channel DDR4-3200
- On-chip Memory: CACTI
- Synthesis: Synopsis Design Compiler
 - 65nm TSMC – 1Ghz
- Layout: Cadence Innovus
- Signal Activity: Modelsim
- Power Estimation: Cadence Innovus



Quantization Accuracy

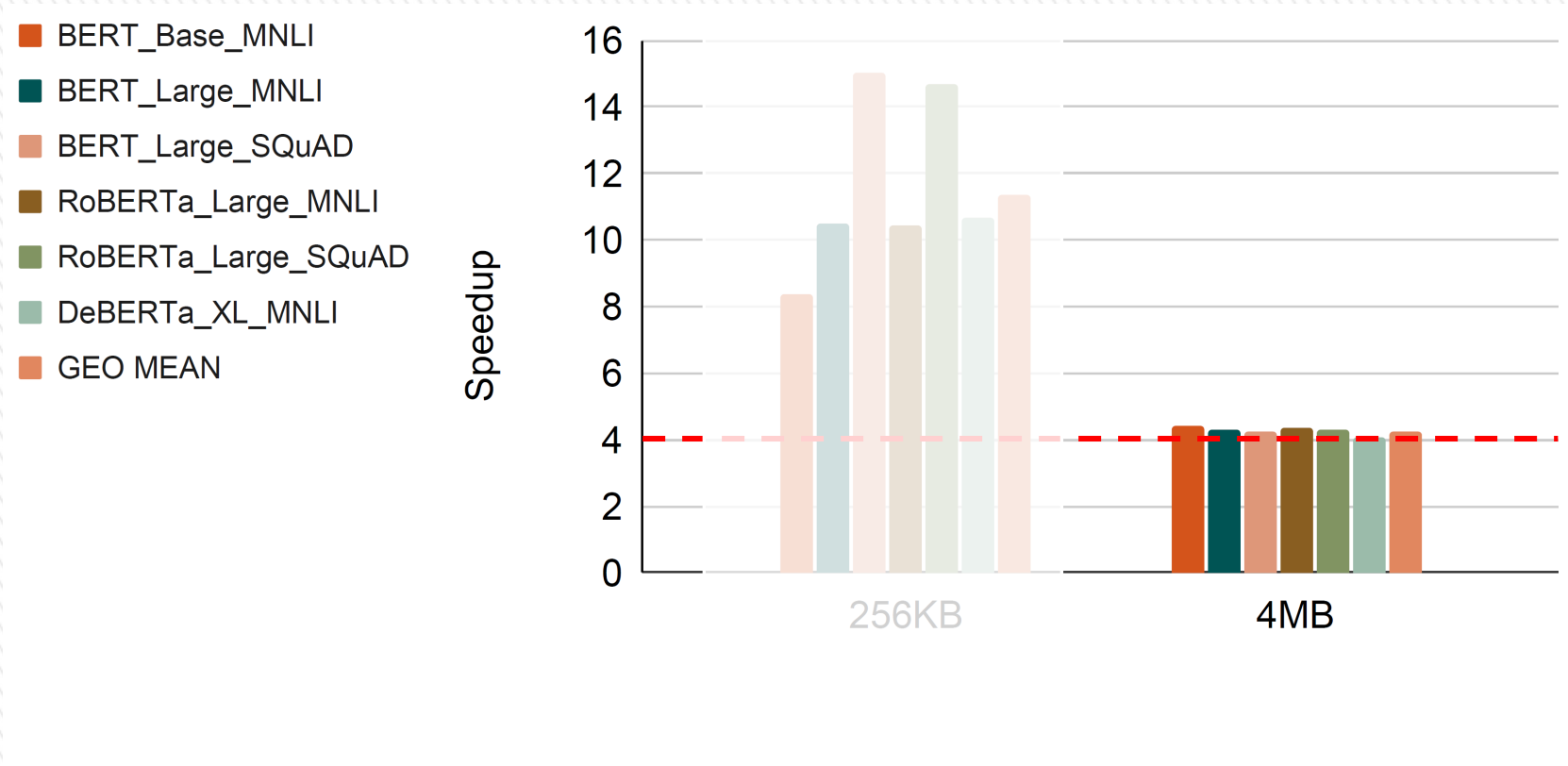


Models/Tasks



Accelerator Performance

Compression: 16-bit to 4-bit => 4x

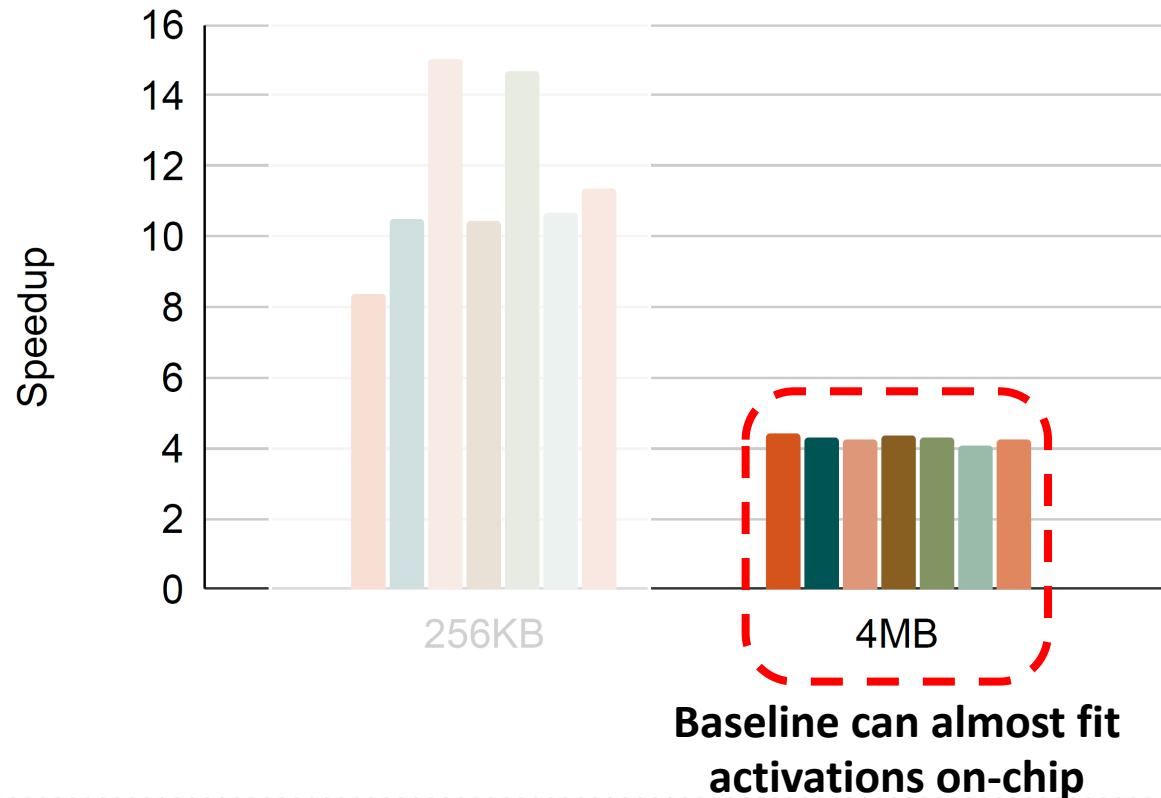




Accelerator Performance

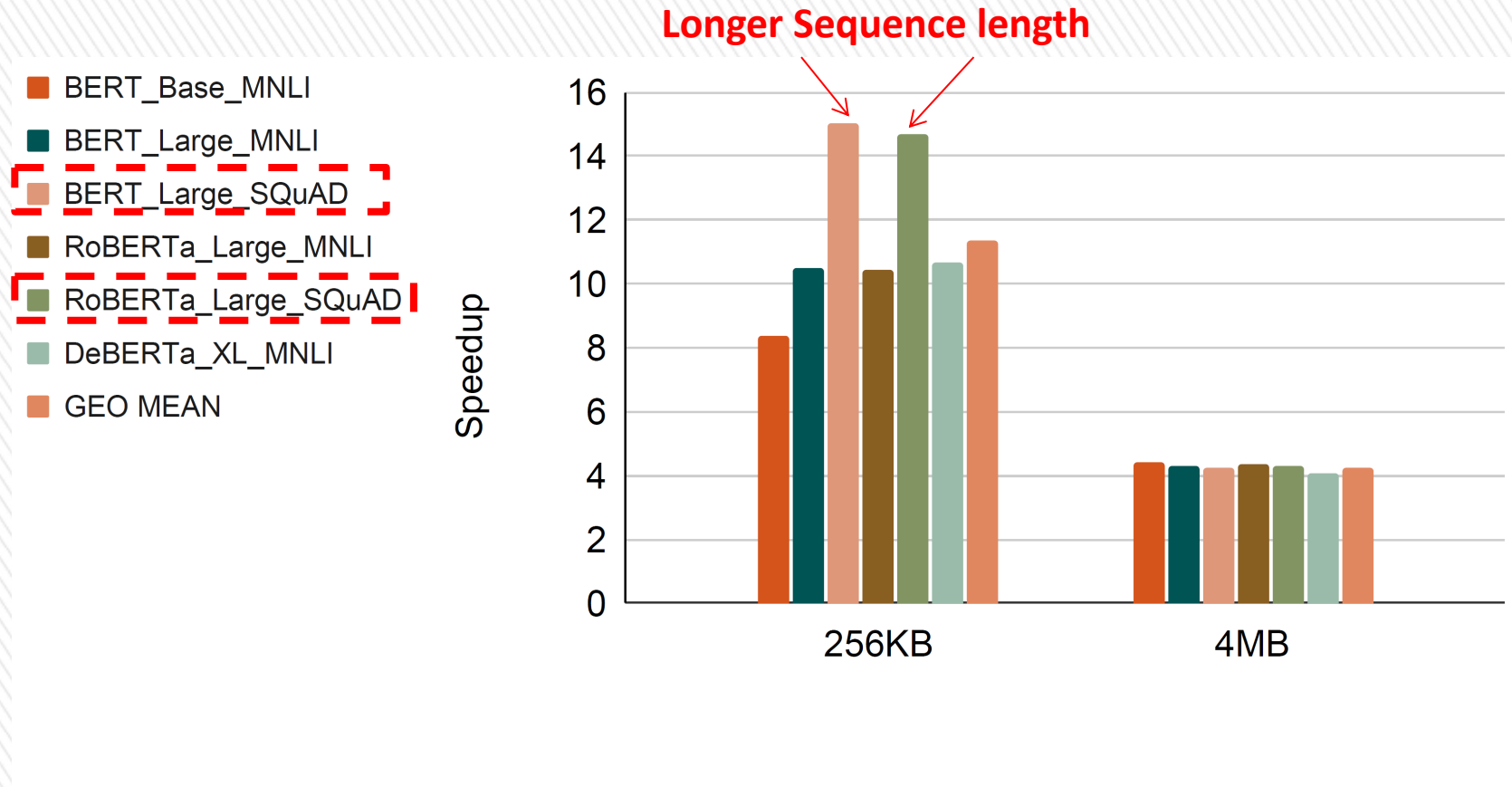
Compression: 16-bit to 4-bit => 4x

- BERT_Base_MNLI
- BERT_Large_MNLI
- BERT_Large_SQuAD
- RoBERTa_Large_MNLI
- RoBERTa_Large_SQuAD
- DeBERTa_XL_MNLI
- GEO MEAN



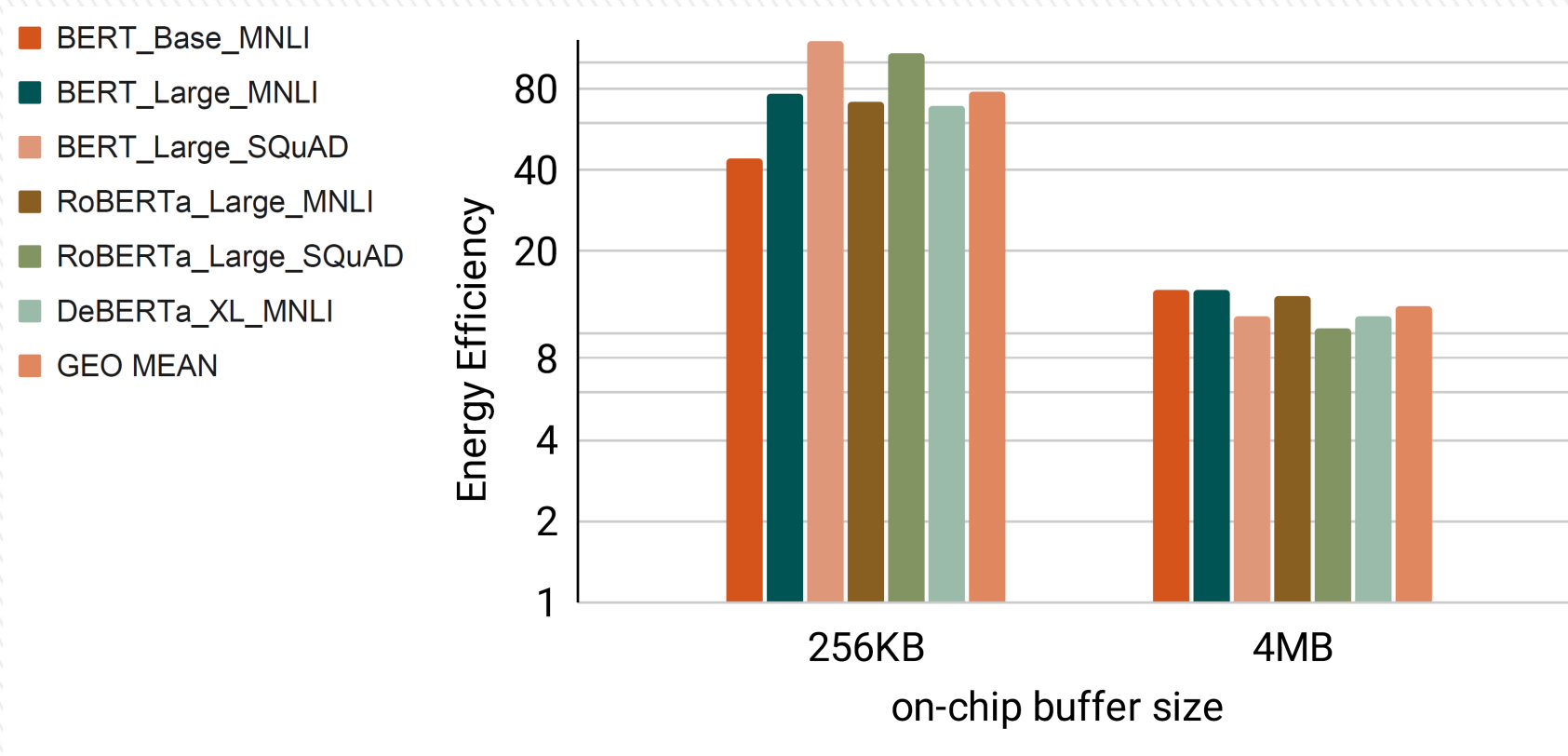
Accelerator Performance

Compression: 16-bit to 4-bit => 4x



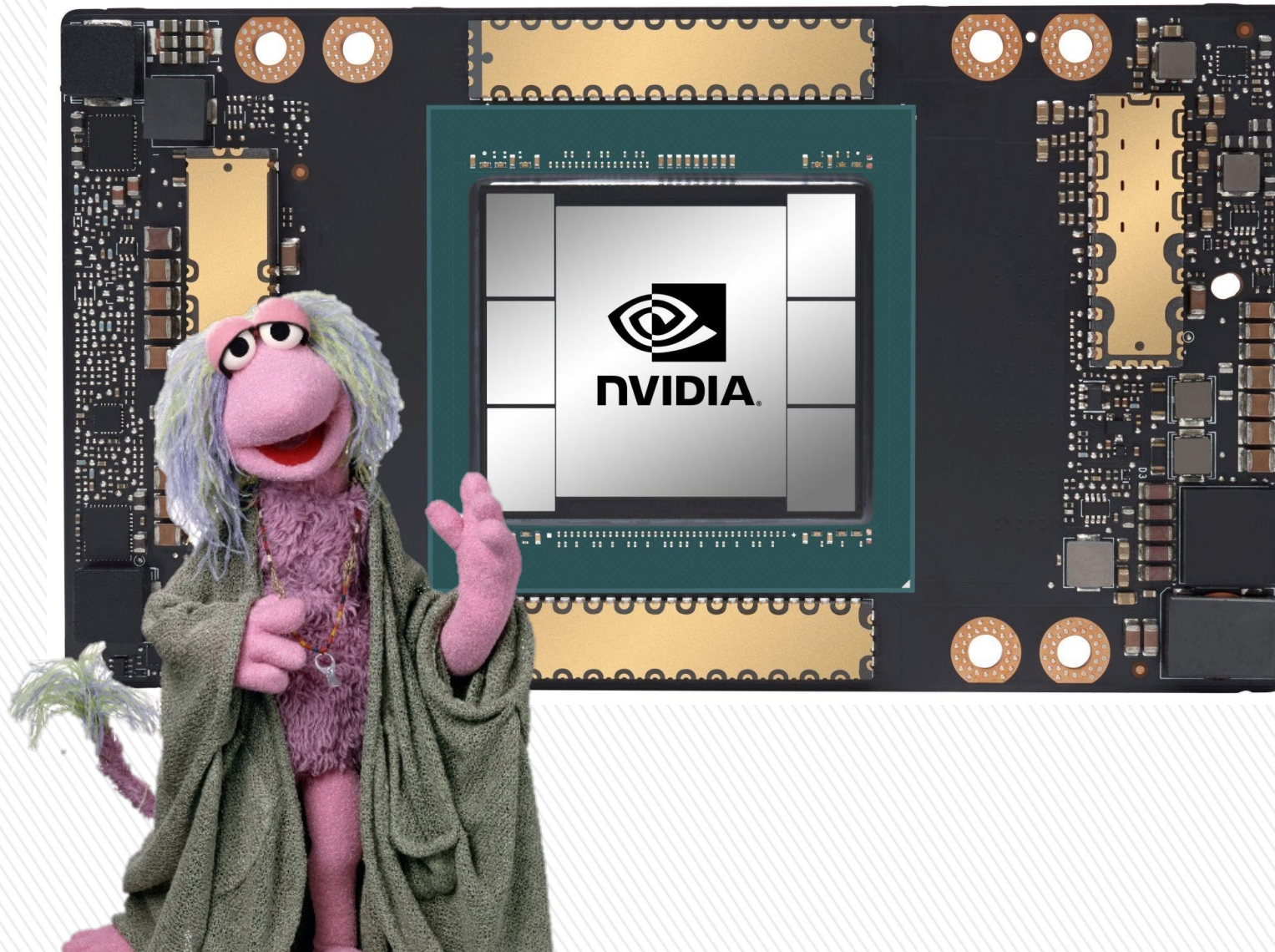


Accelerator Energy Efficiency



Memory Compression and more in paper 😊

Memory Compression



Conclusion

➤ Mokey Quantization:

- 4-bit Dictionary-based
- Focus on a subspace => closed-form representation
- Fixed-point compute
- No fine-tuning

➤ Mokey HW Accelerator:

- Compute directly on indices
- 1.6x Smaller tiles vs. Tensor Cores
- 15x Faster and 100x Energy efficient
- Can be adapted to other accelerators

