# Optimizing Large Language Model Training Using FP4 Quantization

2025.05.12

Efficient ML

**Hyeonseok Shin, SeungHyeon Kim, Yeeun Kim**



*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*

# Introduction

- Background
  - Why do we need low bit training?
- Related work
  - Low bit training
- Introduction
  - 4bit quantization
  - Issues in LLM quantization
    - Straight through estimator (STE)
    - Handling outliers

# Background

- Why do we need low bit training?
  - Time, energy, resources


*https://endplan.ai*

- Example for LLaMA 405B (16K H100 GPUs)

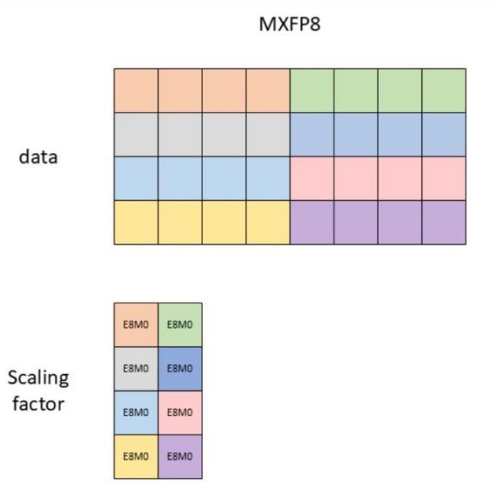
| Component | Category | Interruption Count | % of Interruptions |
|---|---|---|---|
| Faulty GPU | GPU | 148 | 30.1% |
| GPU HBM3 Memory | GPU | 72 | 17.2% |
| Software Bug | Dependency | 54 | 12.9% |
| Network Switch/Cable | Network | 35 | 8.4% |
| Host Maintenance | Unplanned Maintenance | 32 | 7.6% |
| GPU SRAM Memory | GPU | 19 | 4.5% |
| GPU System Processor | GPU | 17 | 4.1% |
| NIC | Host | 7 | 1.7% |
| NCCL Watchdog Timeouts | Unknown | 7 | 1.7% |
| Silent Data Corruption | GPU | 6 | 1.4% |
| GPU Thermal Interface + Sensor | GPU | 6 | 1.4% |
| SSD | Host | 3 | 0.7% |
| Power Supply | Host | 3 | 0.7% |
| Server Chassis | Host | 2 | 0.5% |
| IO Expansion Board | Host | 2 | 0.5% |
| Dependency | Dependency | 2 | 0.5% |
| CPU | Host | 2 | 0.5% |
| System Memory | Host | 2 | 0.5% |

**Table 5 Root-cause categorization of unexpected interruptions during a 54-day period of Llama 3 405B pre-training.** About 78% of unexpected interruptions were attributed to confirmed or suspected hardware issues.

*Grattafiori, et al. arXiv (2024).*

# Related work – Low bit training

- NVIDIA's Transformer Engine (2022)
  - MXFP8
    - Block scaling
    - E8M0 scaling factor



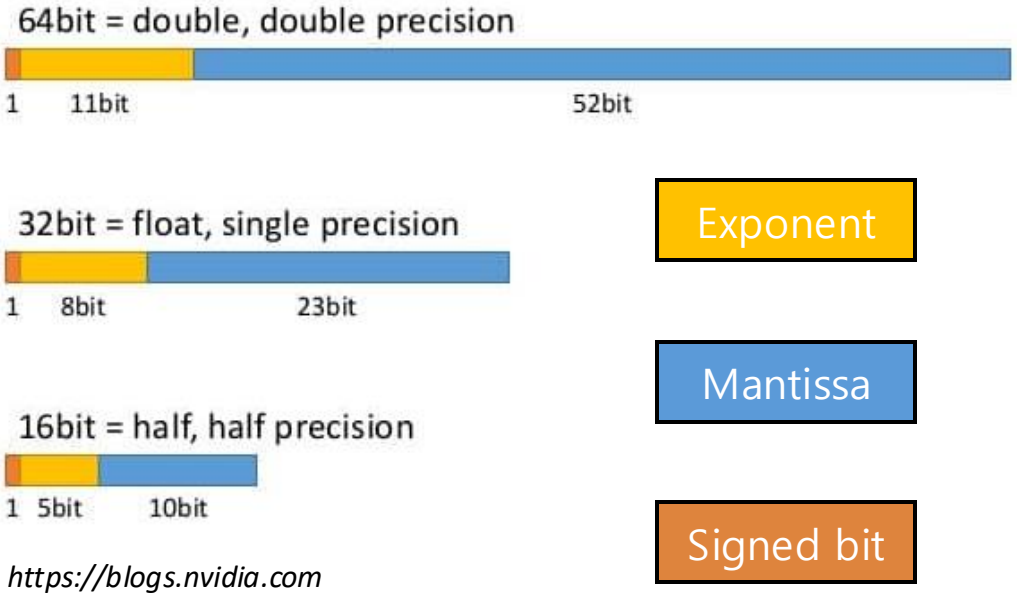

*Rouhani, et al. NVIDIA (2022).*

- FP8-LM: Training FP8 Large Language Model (2023)
  - Precision decoupling
  - Automatic scaling

*Peng, et al. Microsoft (2023).*

# 4bit quantization

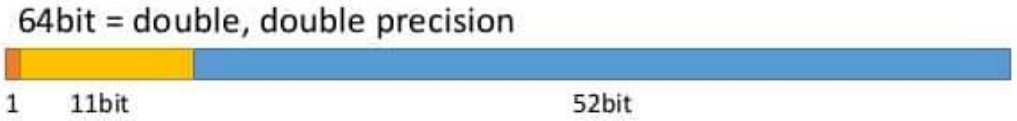
- Floating point
  - Sign / Exponent / Mantissa
  - 64, 32, 16, 8. 4bit




*https://blogs.nvidia.com*

# 4bit quantization

- Floating point
  - Sign / Exponent / Mantissa
  - 64, 32, 16, 8. 4bit





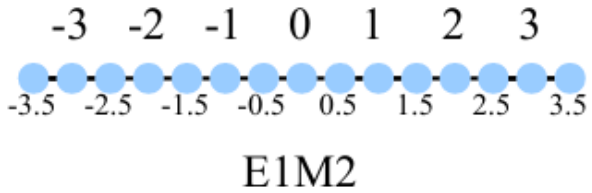

*https://blogs.nvidia.com*

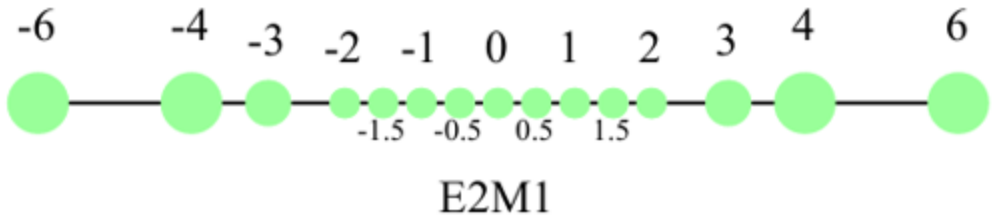
Exponent

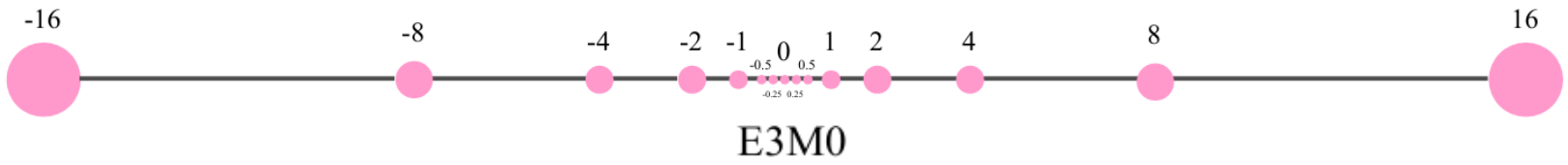
Mantissa

Signed bit

- Format (Ex. FP4)
  - E1M2 / E2M1 / E3M0



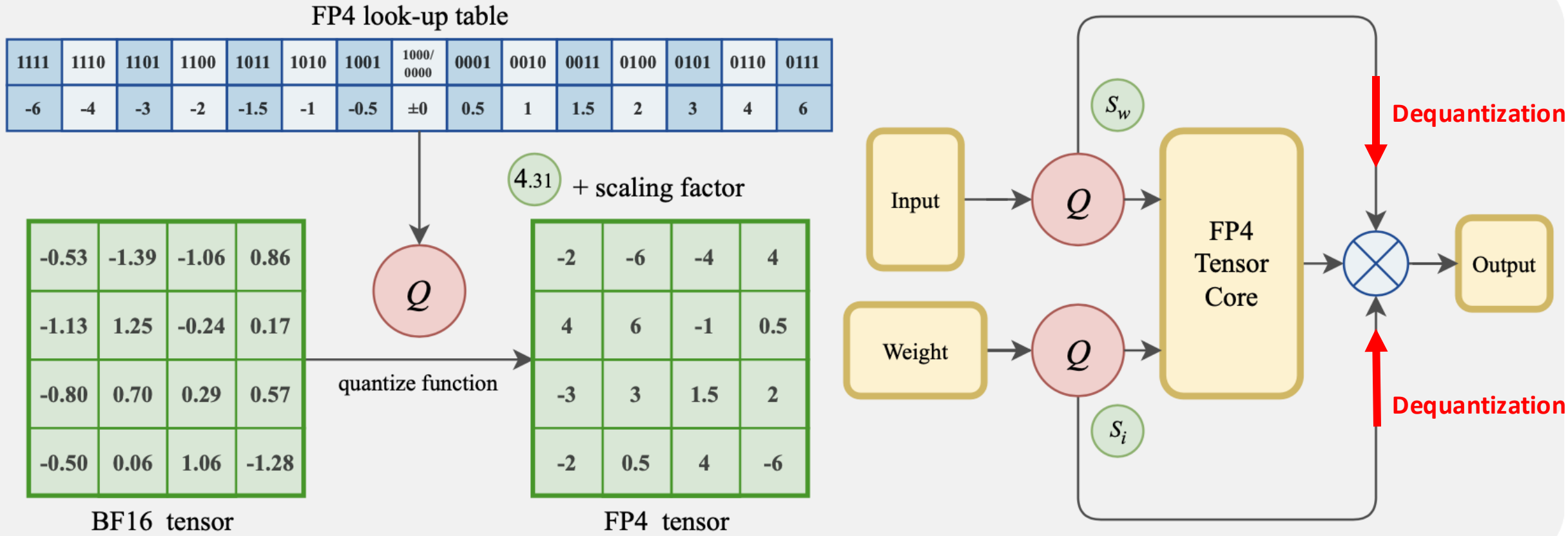





WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)

# 4bit quantization

- FP4 quantization

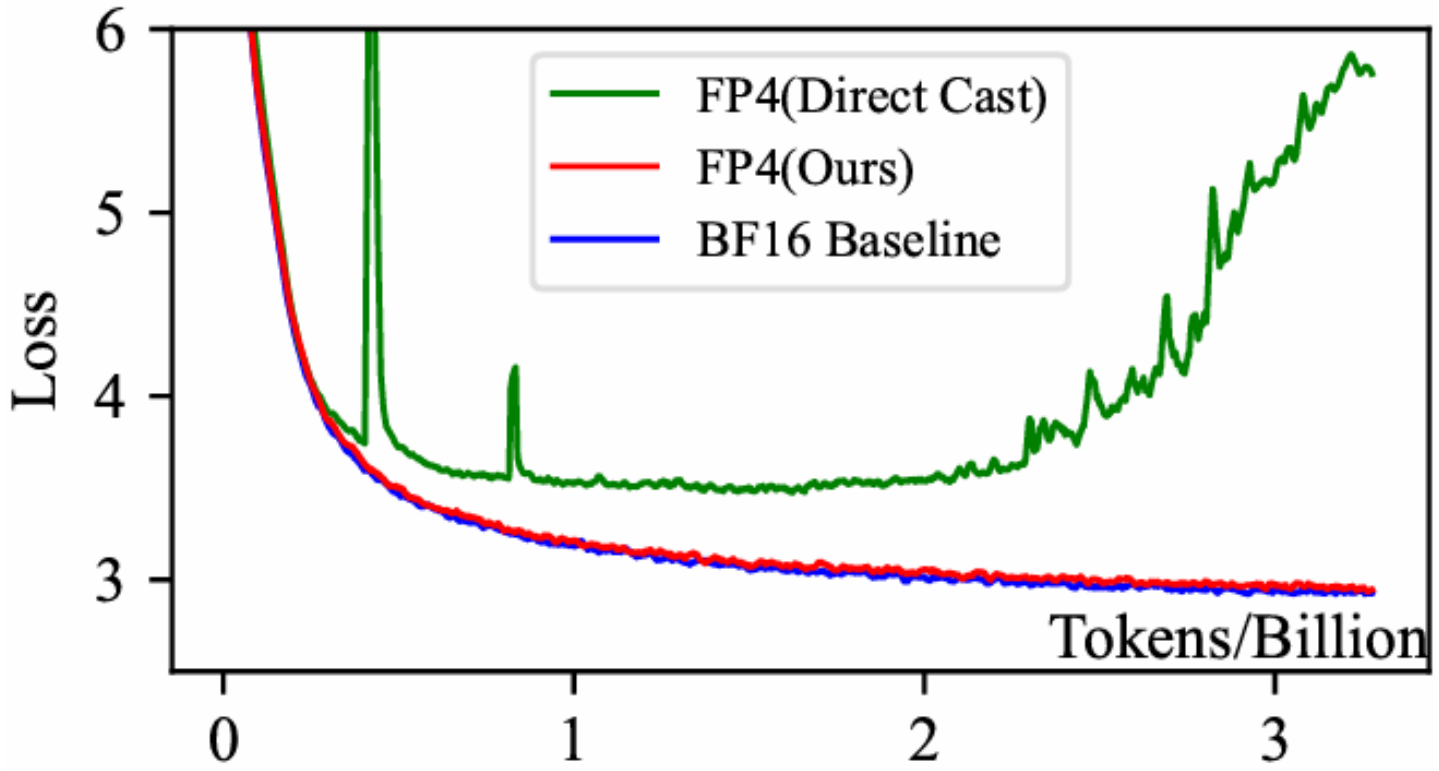




WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)

# 4bit quantization

- FP4 quantization

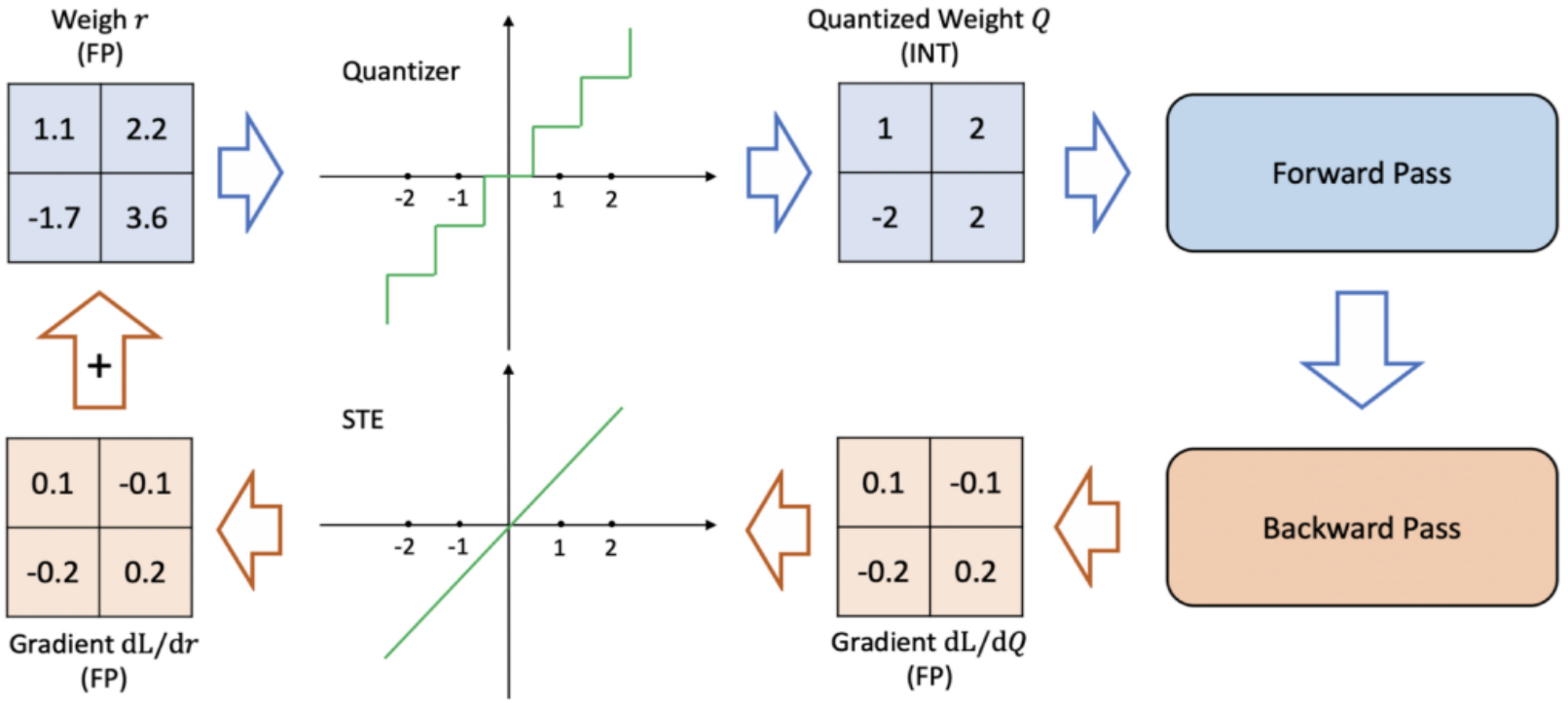



- **Differentiable gradient estimator (DGE)**
- **Outlier clamping and compensation (OCC)**


*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*

# Issues in LLM quantization – Straight through estimator

- Straight through estimator



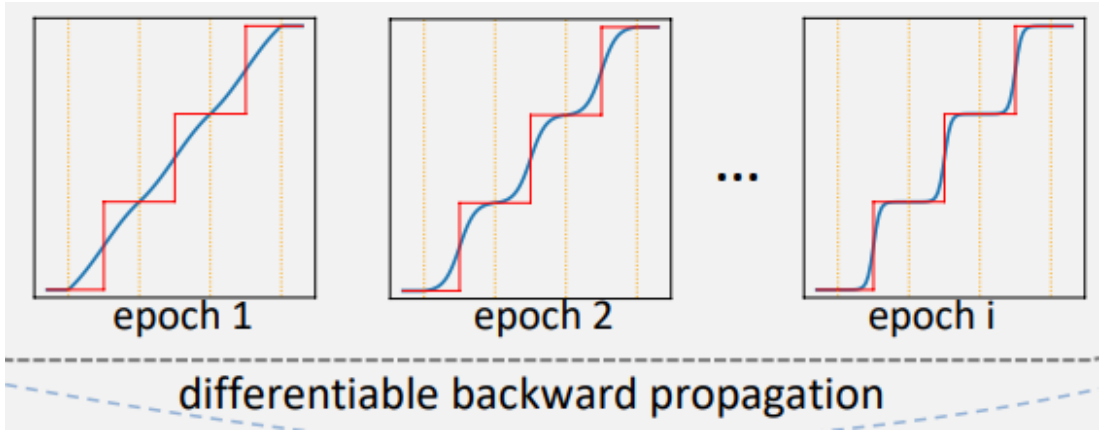

$$2(q(\mathbf{w})^\top \mathbf{x} - y) \cdot \mathbf{x}^\top \nabla_{\mathbf{w}} q(\mathbf{w})$$

$$2(q(\mathbf{w})^\top \mathbf{x} - y) \cdot \mathbf{x}^\top \nabla_{\mathbf{w}} (\mathbf{w})$$

*Bengio, et al. arXiv (2013).*

# Issues in LLM quantization – Straight through estimator

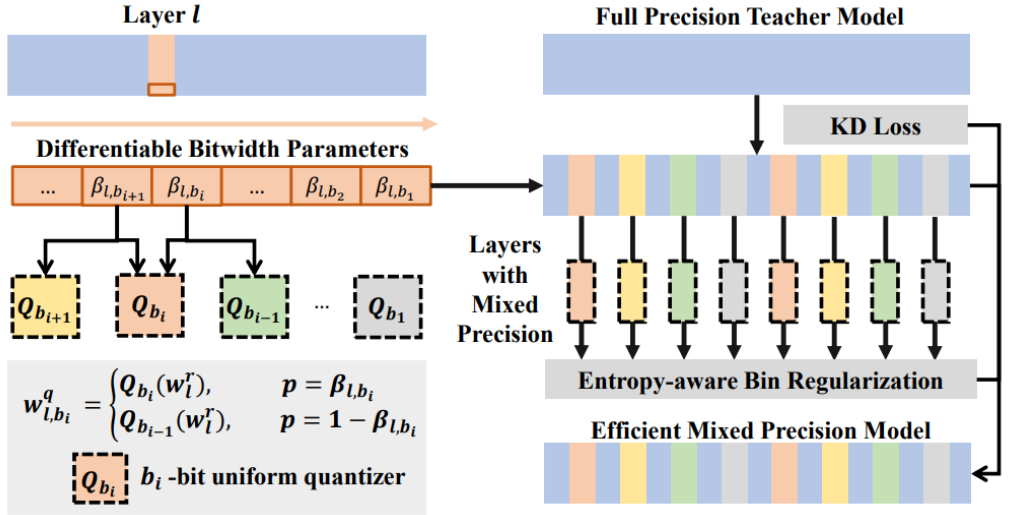
- Differentiable soft quantization (DSQ, 2019)




  - **Hyperbolic tangent function**
  - Closer to rounding function as training progresses

*Gong, Ruihao,et al. ICCV (2019).*

- Stochastic differentiable quantization (SDQ, 2022)
  - Differentiable bit-width parameter (DBP)
  - **Gumbel-softmax**

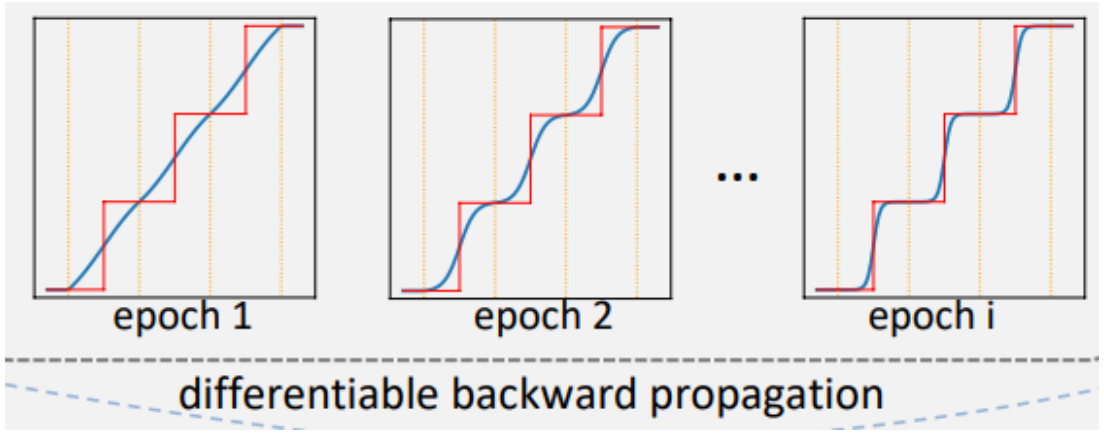



(a) Proposed Stochastic Differentiable Quantization (SDQ)

*Huang, Xijie, et al. ICML (2022).*

# Issues in LLM quantization – Straight through estimator

- Differentiable soft quantization (DSQ, 2019)




- **Hyperbolic tangent function**
- Closer to rounding function as training progresses

*Gong, Ruihao,et al. ICCV (2019).*

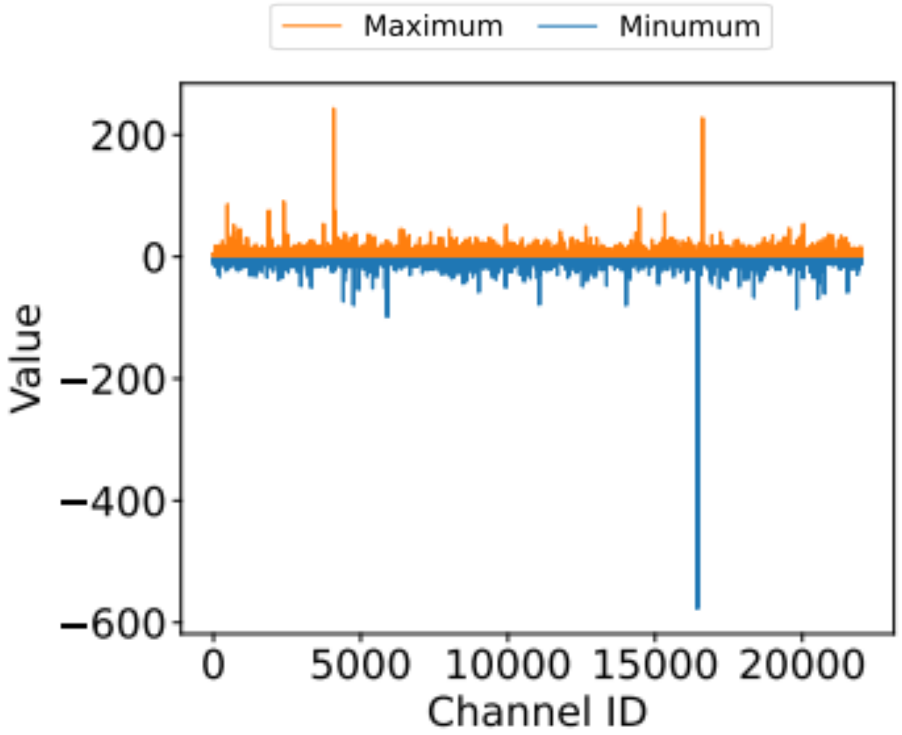
- Stochastic differentiable quantization (SDQ, 2022)
  - Differentiable bit-width parameter (DBP)
  - **Gumbel-softmax**

**Strongly dependent on learnable quantization parameters**

✓ **Proposed Solution: Differentiable gradient estimator (DGE)**

# Issues in LLM quantization – Handling outliers

- Outlier in activation for quantization
  - Input activations of a linear layer in LLaMa-65B

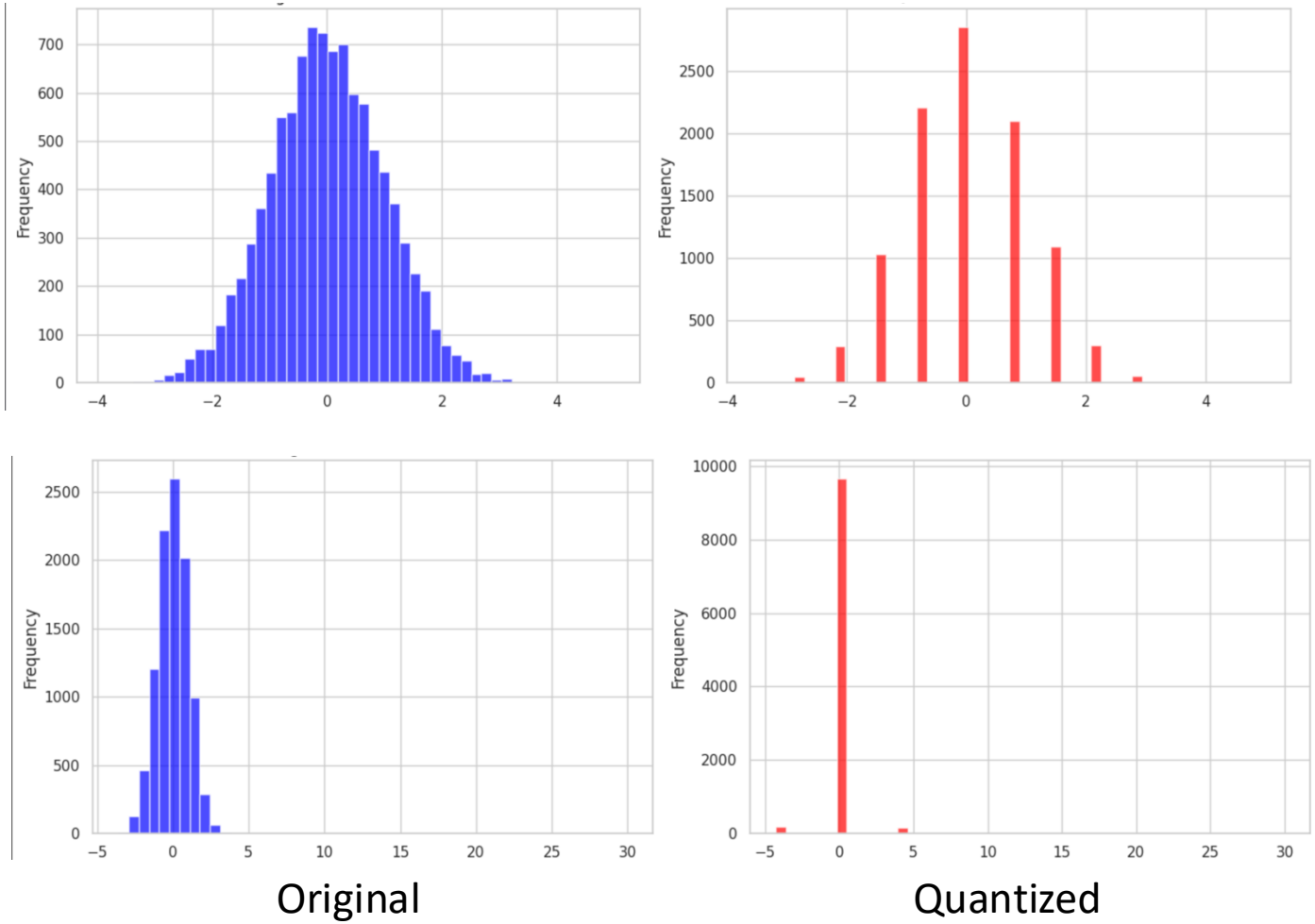



(a) Pre-trained model

*Liu, et al. ICML (2024).*

# Issues in LLM quantization – Handling outliers

- Outlier in activation for quantization
  - Input activations of a linear layer in LLaMa-65B




**e.g.) [-4,4] tensor distribution**

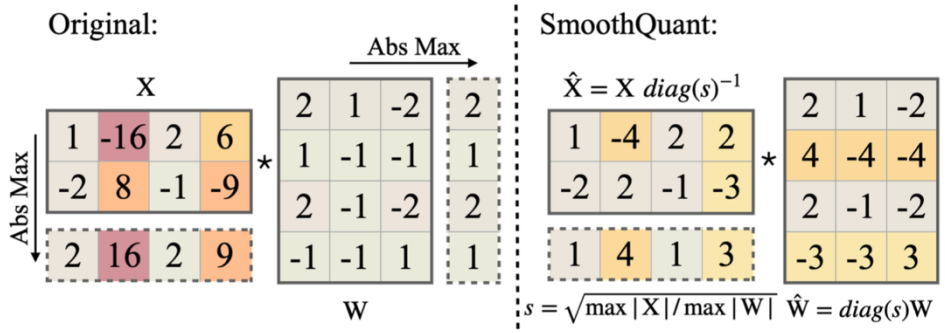
[w/o outlier]
Max magnitude: 5
MSE: 0.04

[w/ outlier]
Max magnitude: 30
MSE: 3.18

**→ Mapping most elements to zero**
**→ Increased MSE**

# Issues in LLM quantization – Handling outliers

- SmoothQuant

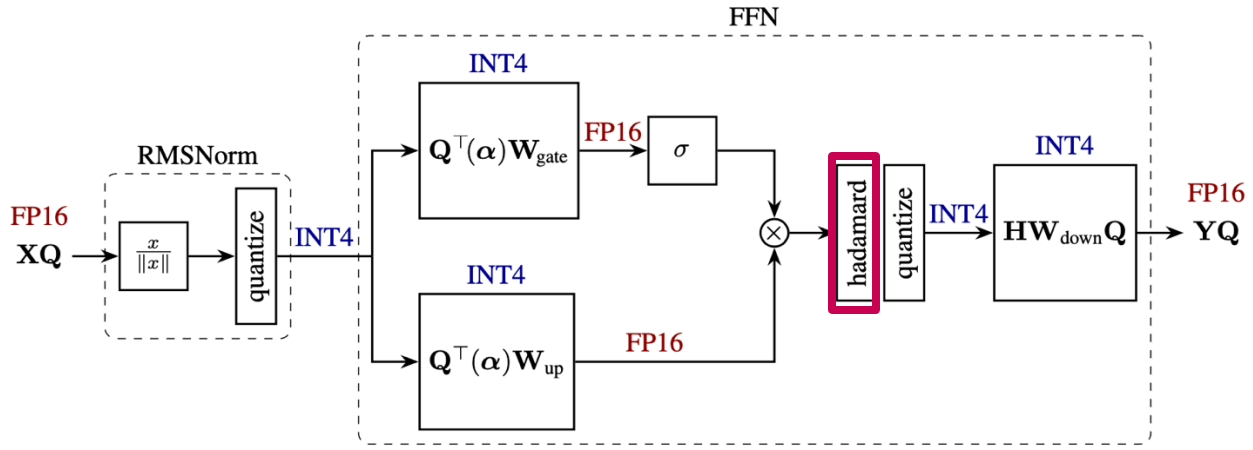



*Xiao, et al. ICML (2023).*

- Per-channel smoothing factor **s**

→ **Conflict with the computation structure**

- QuaRot




*Ashkboos, et al. NeurIPS (2024).*

- Hadamard transformation

→ **Rely on offline pre-processing**

→ **Incompatible with pretraining task**

# Methodology

- Differentiable gradient estimator (DGE)
- Outlier clamping and compensation (OCC)


*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*

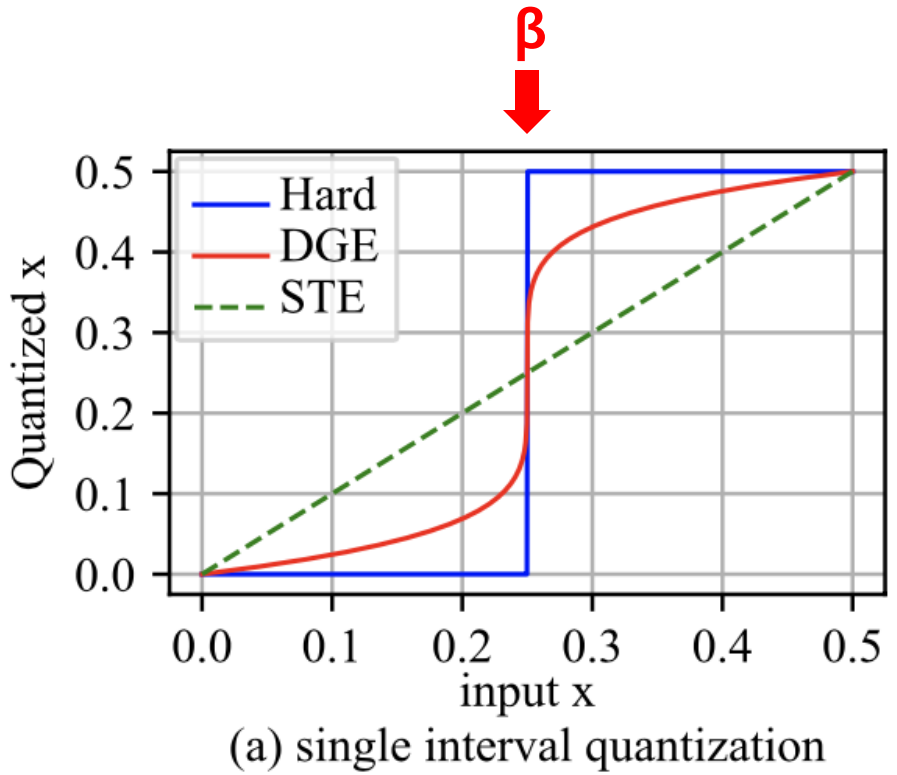
# Methodology – Differentiable gradient estimator (DGE)

- Concept

A gradient correction term derived from a differentiable approximation of the quantization function.

- **Hard**
  $x = \beta$: Non-differentiable
  Others: f'=0

- **STE**
  For all x: f'=1



(a) single interval quantization


WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)

# Methodology – Differentiable gradient estimator (DGE)

- Concept

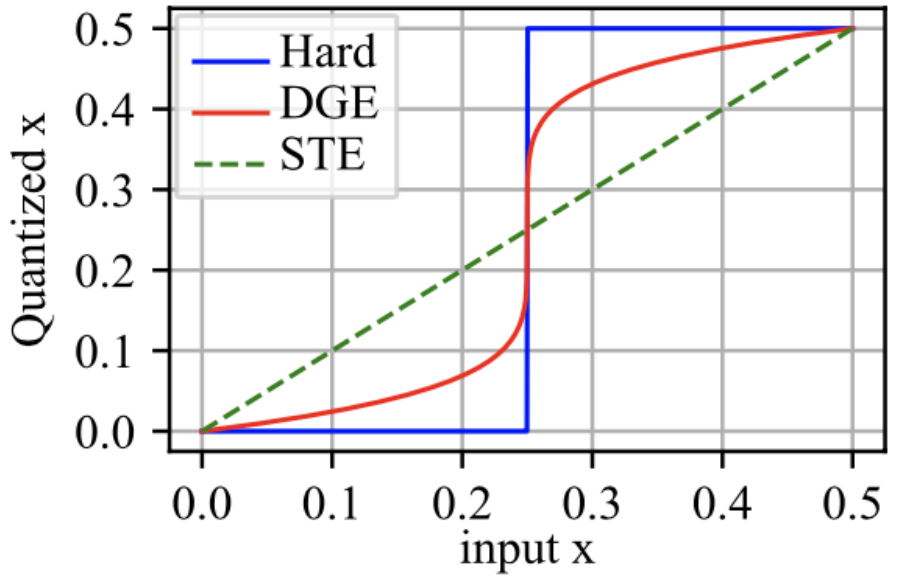
A gradient correction term derived from a differentiable approximation of the quantization function.

- **DGE**

$$f(x) = \delta \cdot \left(1 + \text{sign}(x - \frac{\delta}{2}) \cdot |x - \frac{\delta}{2}|^{\frac{1}{k}}\right)$$

δ: Quantization interval
k: Degree of approximation

k ↑ : Close to hard quantization function



(a) single interval quantization


WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)

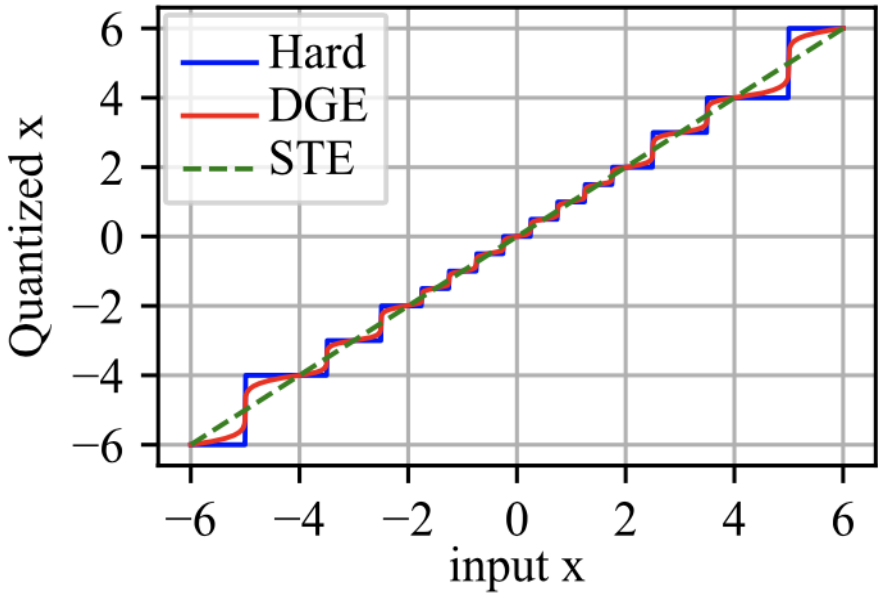
# Methodology – Differentiable gradient estimator (DGE)

- Concept

  A gradient correction term derived from a differentiable approximation of the quantization function.

  - **Applying to E2M1 quantizaition**

    E2M1 range: [-6, 6]






*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*

# Methodology – Differentiable gradient estimator (DGE)

- Concept

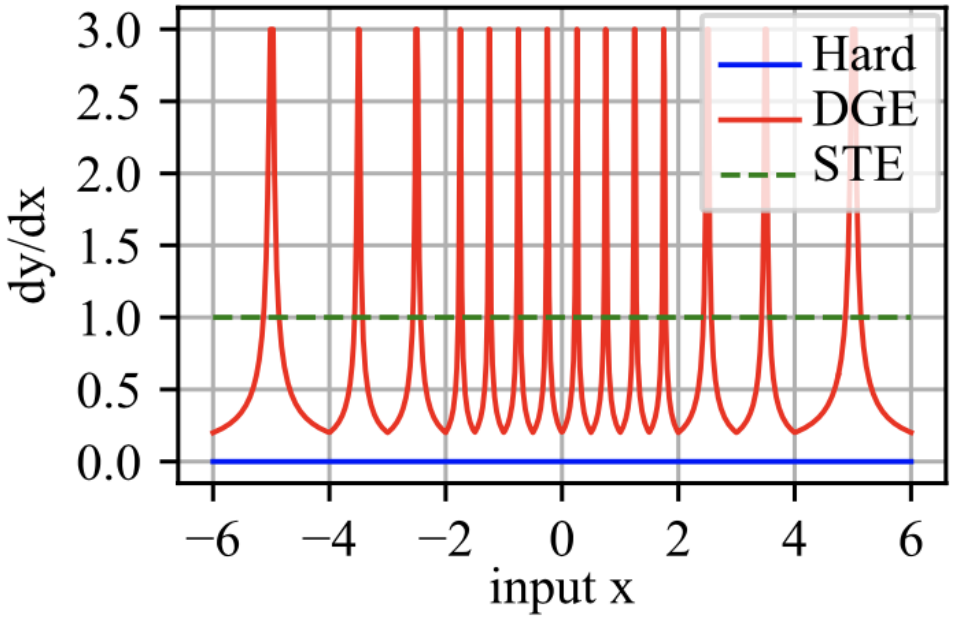
A gradient correction term derived from a differentiable approximation of the quantization function.

- **E2M1 quantizaition derivative**

$$f'(x) = \frac{1}{k} \cdot |x - \frac{\delta}{2}|^{\frac{1}{k}-1}$$

Preventing excessively large gradient values

→ Clipping magnitude of f' at 3.0






WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)

# Methodology – Differentiable gradient estimator (DGE)

- Concept

A gradient correction term derived from a differentiable approximation of the quantization function.

- **E2M1 quantizaition derivative**

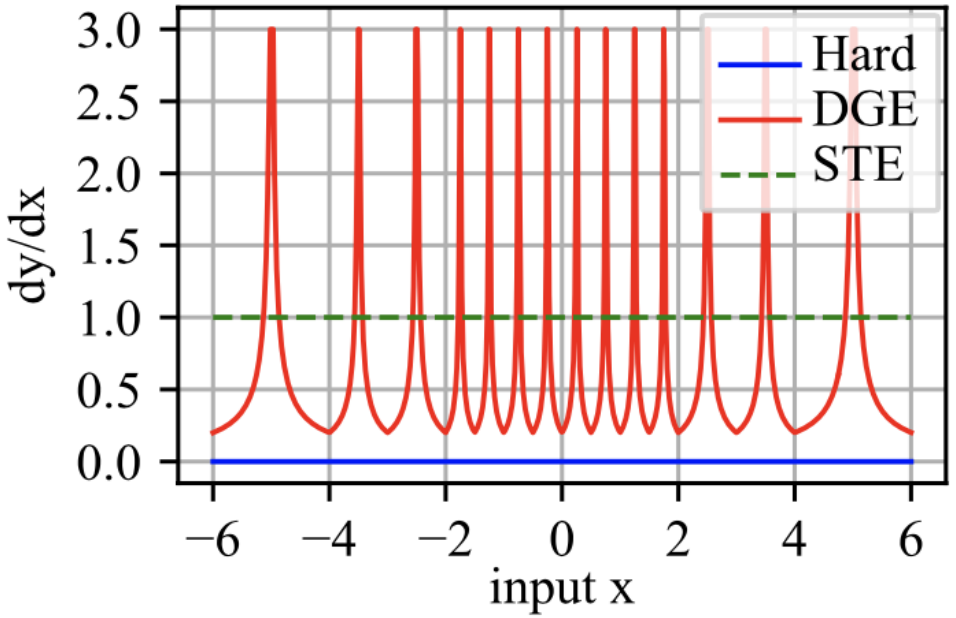
$$f'(x) = \frac{1}{k} \cdot |x - \frac{\delta}{2}|^{\frac{1}{k}-1}$$

Preventing excessively large gradient values

→ Clipping magnitude of f' at 3.0

Forward ➔ Hard quantization function

Backward ➔ DGE quantization function





WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)

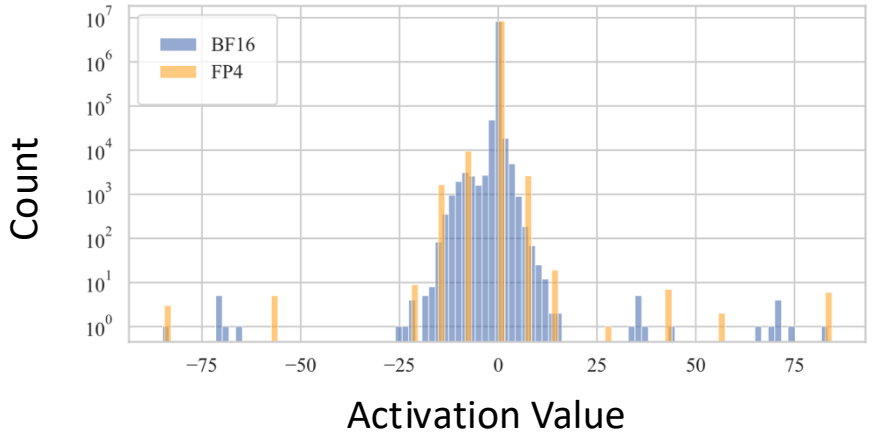
# Methodology – Outlier clamping and compensation (OCC)

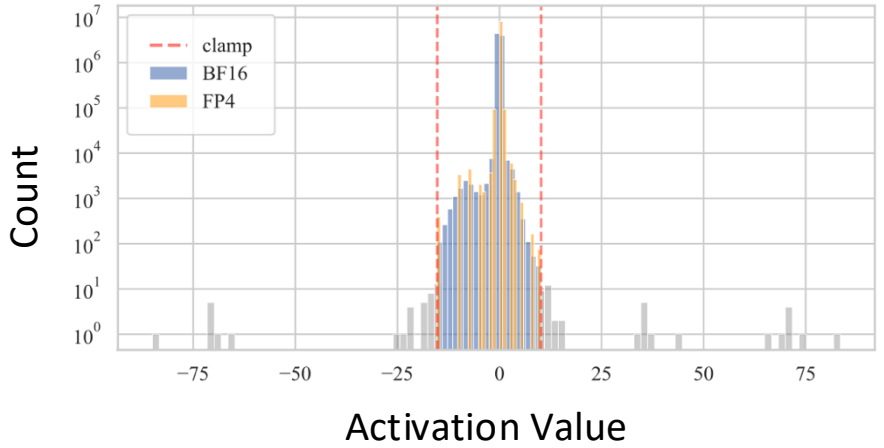
- Concept

Select the top k% largest elements by magnitude and clamp them.

$$Y_c = \text{clamp}(Y, \max = \alpha, \min = 1 - \alpha)$$

α: Quantile (Pre-defined)




LLaMA 1.3B
First transformer layer's output


WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)

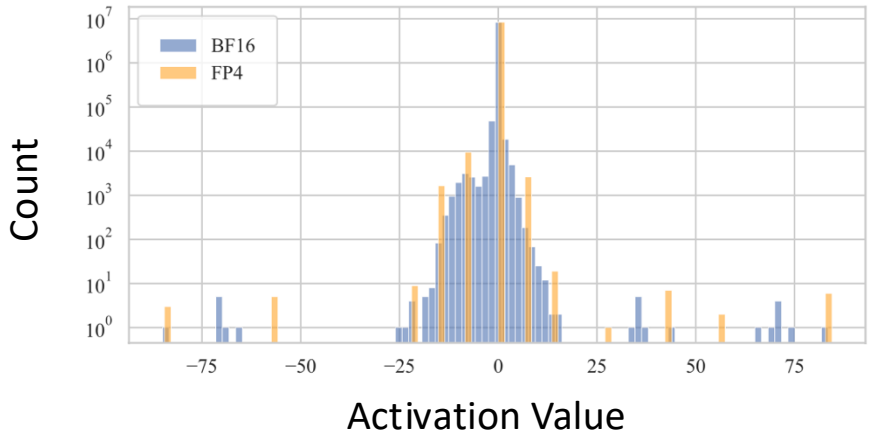
# Methodology – Outlier clamping and compensation (OCC)

- Concept

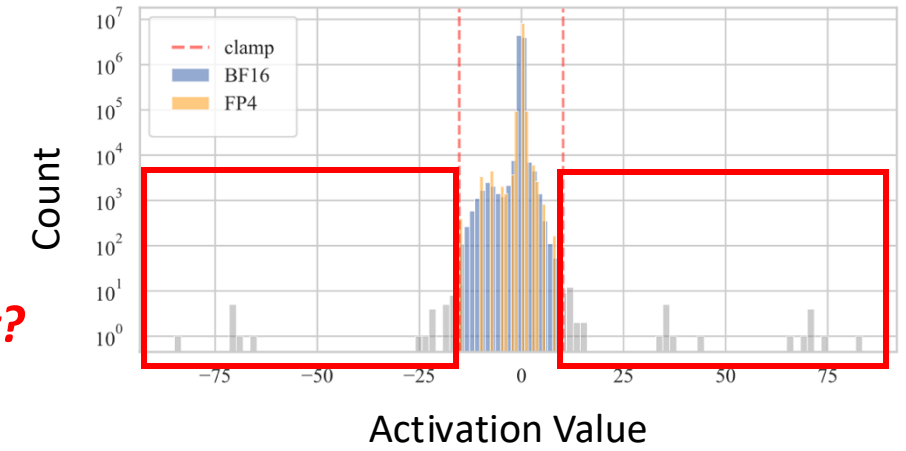
Select the top k% largest elements by magnitude and clamp them.

$$Y_c = \text{clamp}(Y, \max = \alpha, \min = 1 - \alpha)$$

α: Quantile (Pre-defined)




***How to compensate it?***





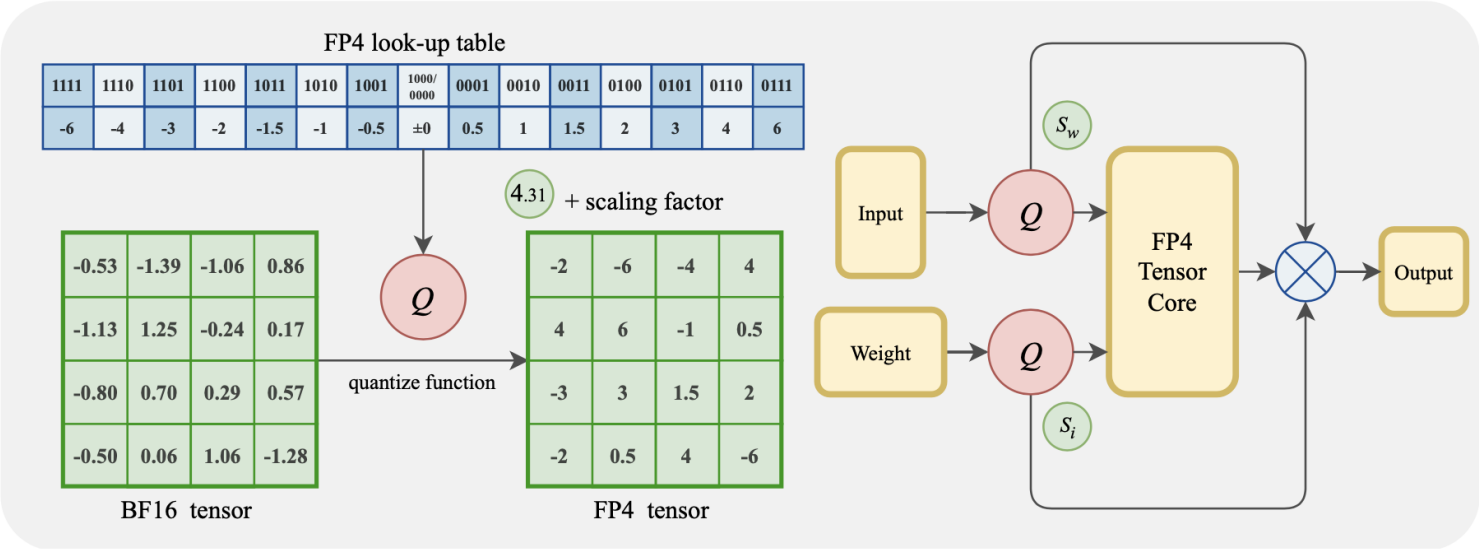

*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*

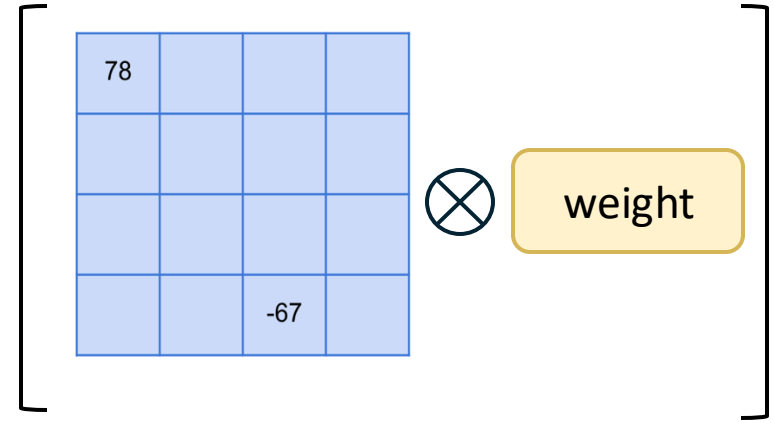
# Methodology – Outlier clamping and compensation (OCC)

- Concept

Select the top k% largest elements by magnitude and clamp them.

- Compensation for clamped outliers
  - Add a sparse outlier matrix after FP4 MM





WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)

# Methodology – Outlier clamping and compensation (OCC)

- Concept

Select the top k% largest elements by magnitude and clamp them.

LLaMA 1.3B, 30,000 training iter.
**SIM, MSE, SNR** between high-precison MM activation tensors and FP4 MM activation tensors

| CLAMP | COMP | QUANTILE | SIM ↑ | MSE ↓ | SNR ↑ |
|---|---|---|---|---|---|
| × | — | — | 92.19% | 0.1055 | 8.31 |
| √ | × | 99.9 | 98.83% | 0.0366 | 14.25 |
| √ | √ | 99.9 | 99.61% | 0.0245 | 15.31 |
| √ | √ | 99 | 100% | 0.0099 | 18.38 |
| √ | √ | 97 | 100% | 0.0068 | 20.88 |

High quantile → Low cost, High quantization error
Low quantile → High cost, Low quantization error


WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)

# Methodology – Outlier clamping and compensation (OCC)

- Concept

Select the top k% largest elements by magnitude and clamp them.

LLaMA 1.3B, 30,000 training iter.
**SIM, MSE, SNR** between high-precison MM activation tensors and FP4 MM activation tensors

| CLAMP | COMP | QUANTILE | SIM ↑ | MSE ↓ | SNR ↑ |
|---|---|---|---|---|---|
| × | — | — | 92.19% | 0.1055 | 8.31 |
| √ | × | 99.9 | 98.83% | 0.0366 | 14.25 |
| √ | √ | 99.9 | 99.61% | 0.0245 | 15.31 |
| √ | √ | 99 | 100% | 0.0099 | 18.38 |
| √ | √ | 97 | 100% | 0.0068 | 20.88 |

▲ Clamping → ☺ **Better SIM, MSE, SNR**


*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*

# Methodology – Outlier clamping and compensation (OCC)

- Concept

Select the top k% largest elements by magnitude and clamp them.

LLaMA 1.3B, 30,000 training iter.
**SIM, MSE, SNR** between high-precison MM activation tensors and FP4 MM activation tensors

| CLAMP | COMP | QUANTILE | SIM ↑ | MSE ↓ | SNR ↑ |
|---|---|---|---|---|---|
| × | — | — | 92.19% | 0.1055 | 8.31 |
| √ | × | 99.9 | 98.83% | 0.0366 | 14.25 |
| √ | √ | 99.9 | 99.61% | 0.0245 | 15.31 |
| √ | √ | 99 | 100% | 0.0099 | 18.38 |
| √ | √ | 97 | 100% | 0.0068 | 20.88 |

▲ Compensation → ☺ **Better SIM, MSE, SNR**


*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*

# Methodology – Outlier clamping and compensation (OCC)

- Concept

Select the top k% largest elements by magnitude and clamp them.

LLaMA 1.3B, 30,000 training iter.
**SIM, MSE, SNR** between high-precison MM activation tensors and FP4 MM activation tensors

| CLAMP | COMP | QUANTILE | SIM ↑ | MSE ↓ | SNR ↑ |
|---|---|---|---|---|---|
| × | — | — | 92.19% | 0.1055 | 8.31 |
| √ | × | 99.9 | 98.83% | 0.0366 | 14.25 |
| √ | √ | 99.9 | 99.61% | 0.0245 | 15.31 |
| √ | √ | 99 | 100% | 0.0099 | 18.38 |
| √ | √ | 97 | 100% | 0.0068 | 20.88 |

▲ Lower Quantile → ☺ **Better SIM, MSE, SNR**


*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*

# Methodology – Outlier clamping and compensation (OCC)

- Concept

Select the top k% largest elements by magnitude and clamp them.

LLaMA 1.3B, 30,000 training iter.
**SIM, MSE, SNR** between high-precison MM activation tensors and FP4 MM activation tensors

| CLAMP | COMP | QUANTILE | SIM ↑ | MSE ↓ | SNR ↑ |
|---|---|---|---|---|---|
| × | — | — | 92.19% | 0.1055 | 8.31 |
| √ | × | 99.9 | 98.83% | 0.0366 | 14.25 |
| √ | √ | 99.9 | 99.61% | 0.0245 | 15.31 |
| √ | √ | 99 | 100% | 0.0099 | 18.38 |
| √ | √ | 97 | 100% | 0.0068 | 20.88 |

▲ **Note: Trade-off** between computational efficiency and numerical accuracy


*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*

# Methodology – Outlier clamping and compensation (OCC)

- Cost

Select top k% largest magnitude elements → **By sort algorithm!**

```
sorted_tensor = torch.sort(input, dim=0).values
lower_index = int((1 - clip_threshold) * sorted_tensor.size(0))
upper_index = int(clip_threshold * sorted_tensor.size(0))

lower_bound = sorted_tensor[lower_index:lower_index+1, :]
upper_bound =
sorted_tensor[upper_index:upper_index+1, :]

output = torch.clamp(input, min=lower_bound,
max=upper_bound)
```

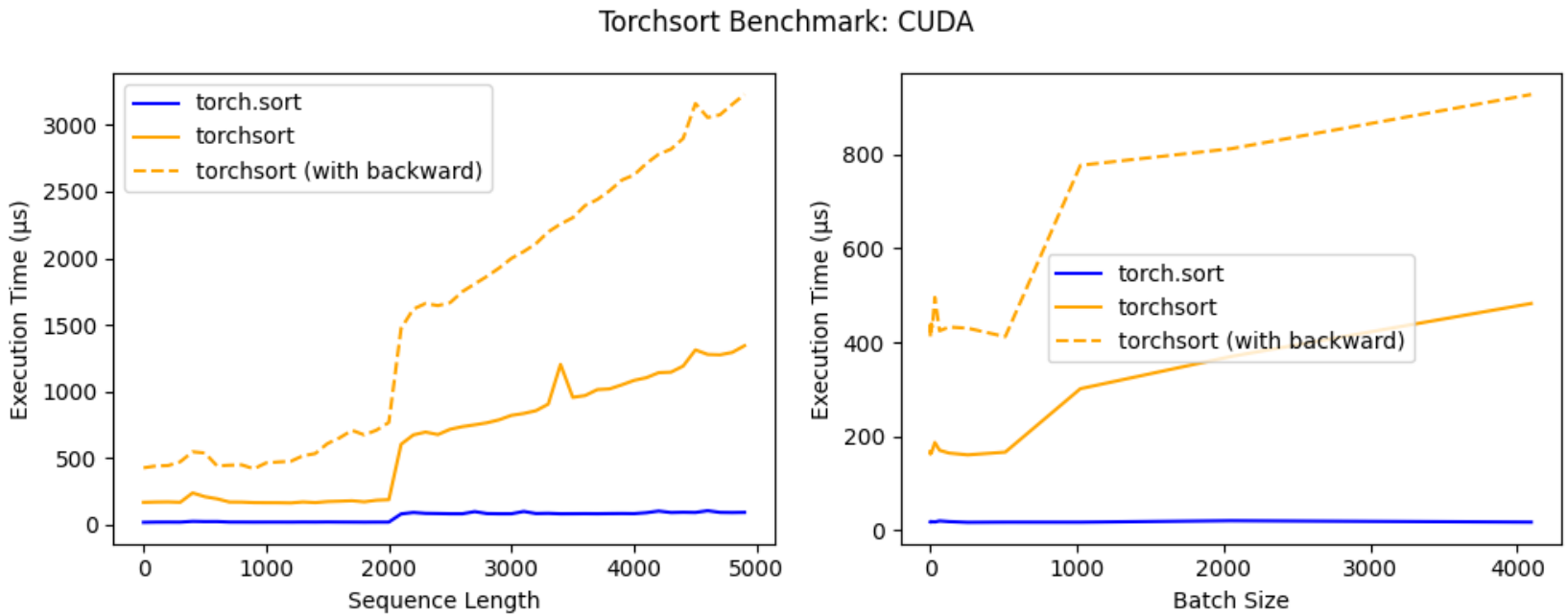


*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*

# Methodology – Outlier clamping and compensation (OCC)

- Cost

Select top k% largest magnitude elements → **By sort algorithm!**




*Blondel et al. ICML (2020).*

- Constant time complexity

☹ But percentage of computation per iteration **not mentioned in the paper**



*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*

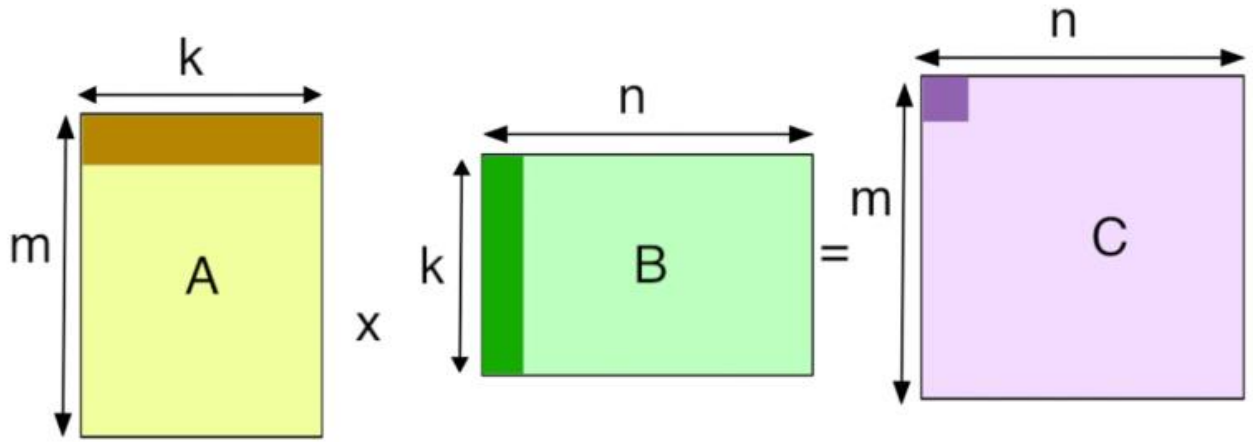
# Experiments and results

- Experiment setup
- Main results
- Ablation studies


WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)

# Experiments and results

- Experiment setup




▲ General Matrix Multiplication (GeMM)

$$Y = A \cdot W$$

A : Activation tensor (sequence length × input channels) ➔ **Token-wise quantization**

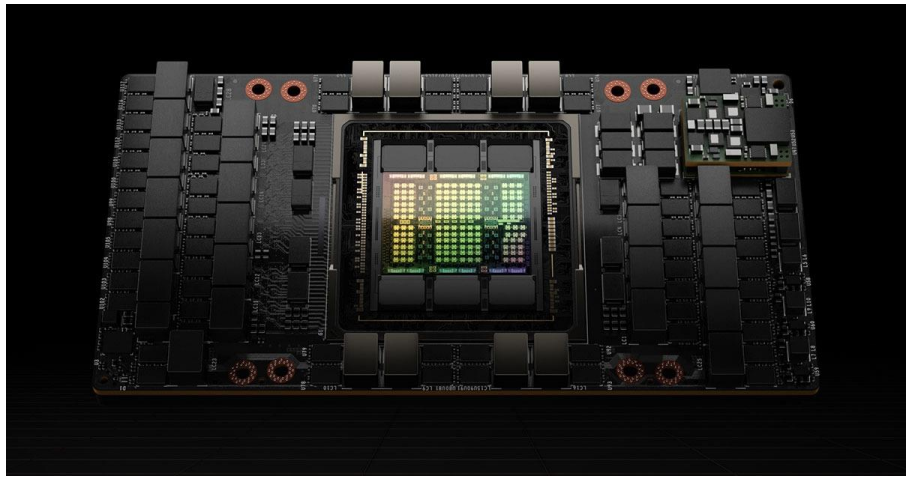
W : Weight tensor (input channels × output channels) ➔ **Channel-wise quantization**


*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*

# Experiments and results

- Experiment setup

  - GeMM operation



*Source : Nvidia*

Validation with Nvidia H-series **GPUs' FP8 Tensor Cores**

☺ Including the range of representation of FP4

  - Mixed-precision training

**1) Gradient communication** in FP8 format

☺ Reducing bandwidth usage

**2) Mixed-precision Adam optimizer**

☺ Conserving GPU memory

| Value (Parameter) | Stored format |
|---|---|
| Gradients first-order moments of Adam optimizer | FP8 |
| second-order moments of Adam optimizer | FP16 |
| Remaining operations (comprising a smaller computational portion) | FP16 or BF16 |



*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*

# Experiments and results

- Experiment setup

  Evaluate the proposed FP4 training framework across language models of various size

  - LLaMA 2 model (Primary model architecture)

    1) The training → From scratch using the **DCLM dataset**

    2) Parameter for proposed method → k = 5 (for **differentiable gradient estimator**)

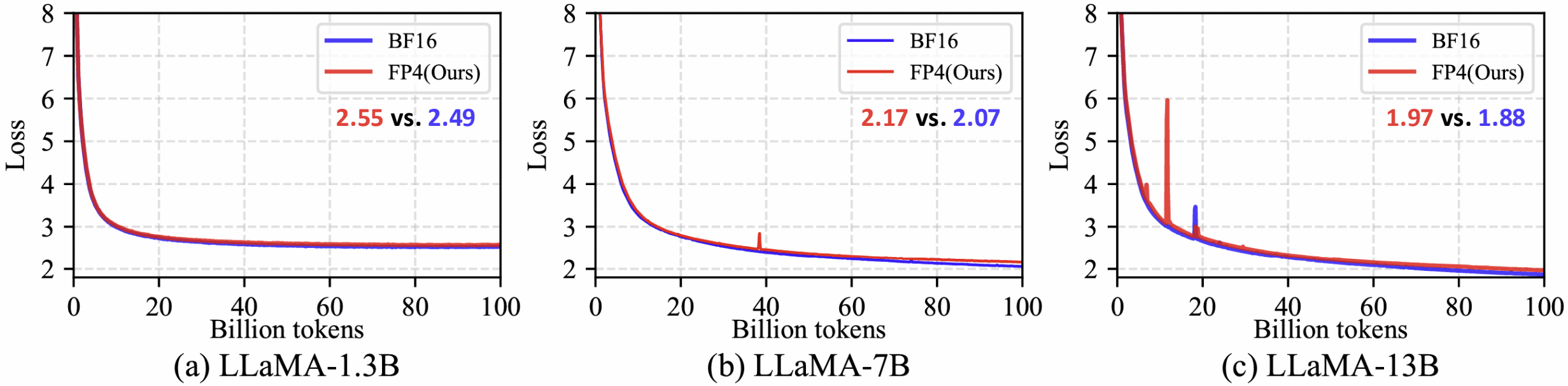
    → α = 0.99 (as the **activation clamp and compensation quantile**)

    3) Input sequences / batch size → 2048 tokens / 2048

# Experiments and results

- Main results

**LLaMA models** (1.3B, 7B, and 13B) trained with BF16 and FP4 precision




☺ As a result, **FP4 ≒ BF16 learning curves**


*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*

# Experiments and results

- Main results

**Zero-shot evaluation** for downstream tasks between BF16 models and FP4 models

| Model Size | Precision | **Average** | PiQA | Hellaswag | ObQA | Arc-C | Arc-E | BoolQ | LogiQA | SciQ | Lambada |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1.3B | BF16 | **53.23** | 71.11 | 50.80 | 36.60 | 36.69 | 68.60 | 57.83 | 30.26 | 83.30 | 43.84 |
| | FP4(Ours) | **53.13** | 70.89 | 50.82 | 36.20 | 36.86 | 67.47 | 58.23 | 29.49 | 83.90 | 44.30 |
| 7B | BF16 | **53.87** | 71.22 | 52.03 | 37.40 | 38.99 | 67.47 | 60.55 | 27.65 | 85.00 | 44.56 |
| | FP4(Ours) | **54.42** | 71.87 | 52.97 | 38.40 | 39.85 | 67.97 | 62.20 | 27.96 | 84.70 | 43.88 |
| 13B | BF16 | **54.44** | 72.80 | 53.56 | 38.60 | 38.82 | 67.97 | 57.40 | 29.65 | 86.30 | 44.87 |
| | FP4(Ours) | **54.95** | 73.78 | 54.12 | 39.60 | 39.68 | 67.89 | 55.90 | 30.88 | 85.80 | 46.89 |

☺ **zero-shot performance of FP4 ≒ zero-shot performance of BF16**

➔ **In-context learning ability preserved** , i.e., solve problems in context



*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*

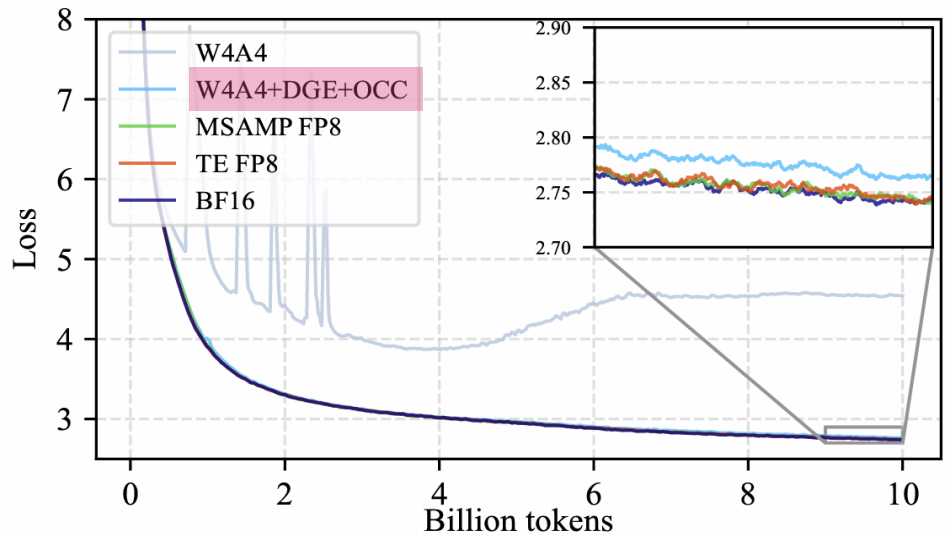
# Experiments and results

- Ablation study

**LLaMA 1.3B model**, trained with 10B tokens from a subset of the DCLM dataset (batch 256)

BF16(baseline), MS-AMP, Transformer-Engine FP8, directly-casted FP4, and **our FP4 method**

- Precision




✓ W4A4 : Quantizing both weight and activation to FP4

✓ W4A4 +DGE+OCC : Proposed FP4 quantization method

✓ MS-AMP FP8, Transformer-Engine FP8: Existing FP8 method

☺ Proposed FP4 → Showing **comparable training loss** curves to FP8 methods during pretraining



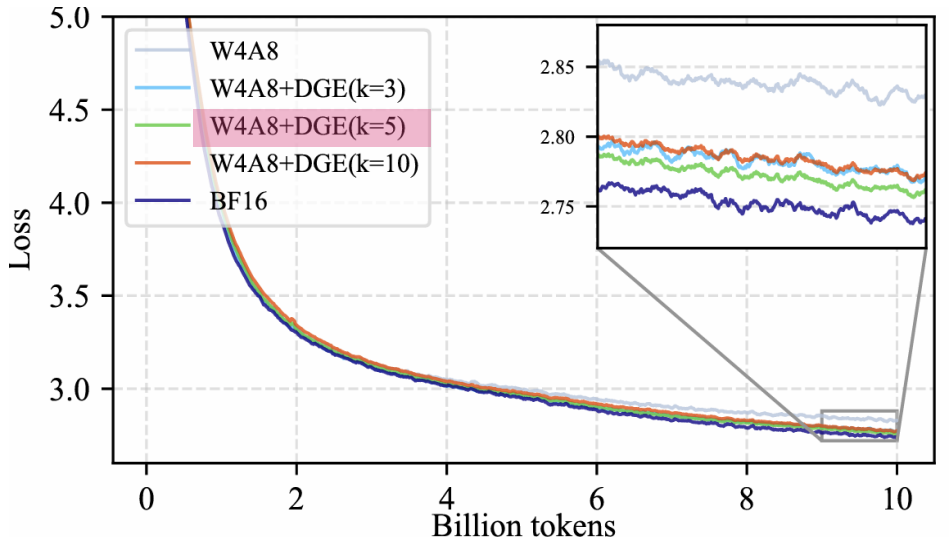
*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*

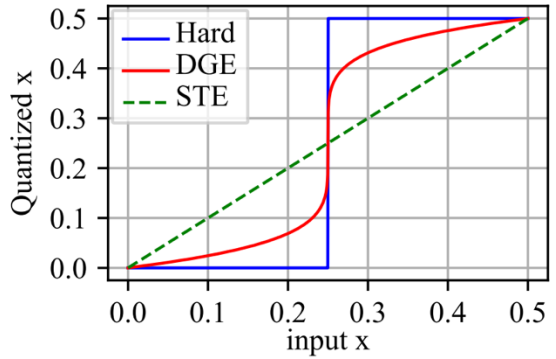
# Experiments and results

- Ablation study

**LLaMA 1.3B model**, trained with 10B tokens from a subset of the DCLM dataset (batch 256)

- Weights




✓ W4A8 : Weight-only 4-bit quantization

✓ W4A8 +DGE (k= __) → DGE method with hyperparameter k

(k : Parameter that controls the degree of approximation)

▲ Larger k → Better quantization-function modeling ☹ **Correction-term instability (Trade-off)**

☺ **k = 5** → Optimal final performance



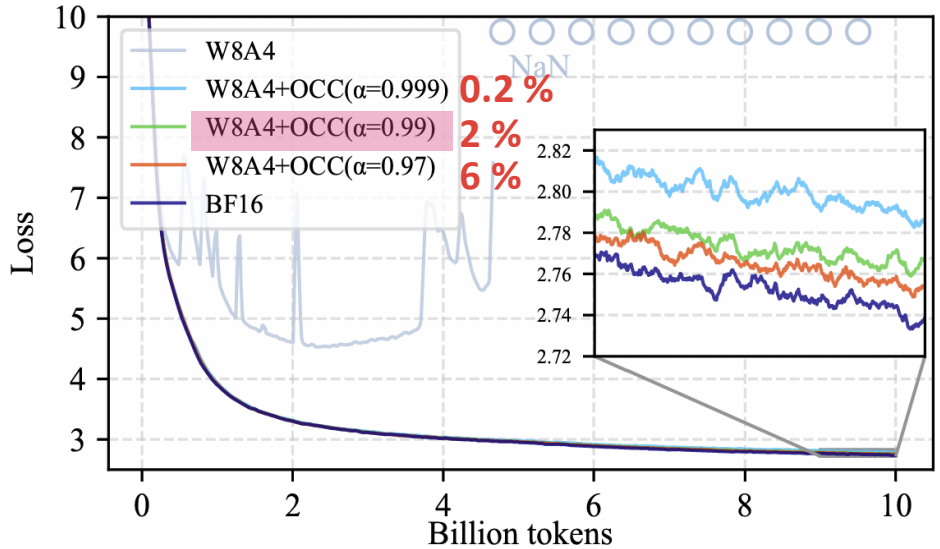
*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*

# Experiments and results

- Ablation study

**LLaMA 1.3B model**, trained with 10B tokens from a subset of the DCLM dataset (batch 256)

- Activation




✓ W8A4 : Activation-only 4-bit quantization

✓ W8A4 +OCC (α= __ ) → OCC method with quantile α

(α : Predefined threshold for quantile searching and clamping)

▲ Direct FP4 activation quantization → Curve divergence

▲ **Higher α → Increased representation ability of model**

☺ **α = 0.99** → Comprehensive computational performance

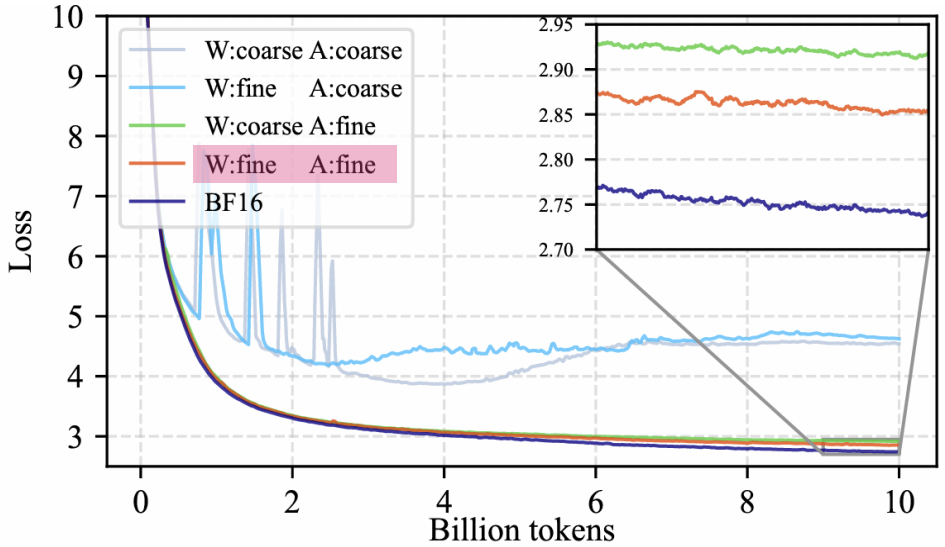

*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*

# Experiments and results

- Ablation study

**LLaMA 1.3B model**, trained with 10B tokens from a subset of the DCLM dataset (batch 256)

- Granularity




| Types of granularity |
|---|
| Tensor-wise |
| Channel-wise |
| Token-wise |
| Element-wise |

**Coarse-grained**

**Fine-grained**

| | Weight | Activation |
|---|---|---|
| | Tensor- | Tensor- |
| | Channel - | Tensor- |
| | Tensor- | Token- |
| | Channel- | Token- |

▲ **Token-wise activation** quantization · **Channel-wise weight** quantization

☹ **Coarse-grained** activation quantization → Propagation of quantization precision loss to the entire output


*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*

# Conclusion


WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)

# Conclusion

- FP4 pre-training framework for modern Large Language Models (LLMs)

  Overcoming the challenges of limited dynamic range and quantization precision in 4-bit formats

  - Differentiable gradient estimator (DGE)

    ☺ Differentially **feasible approximation method** to make gradients flow well for **weights**

  - Outlier clamping and compensation (OCC)

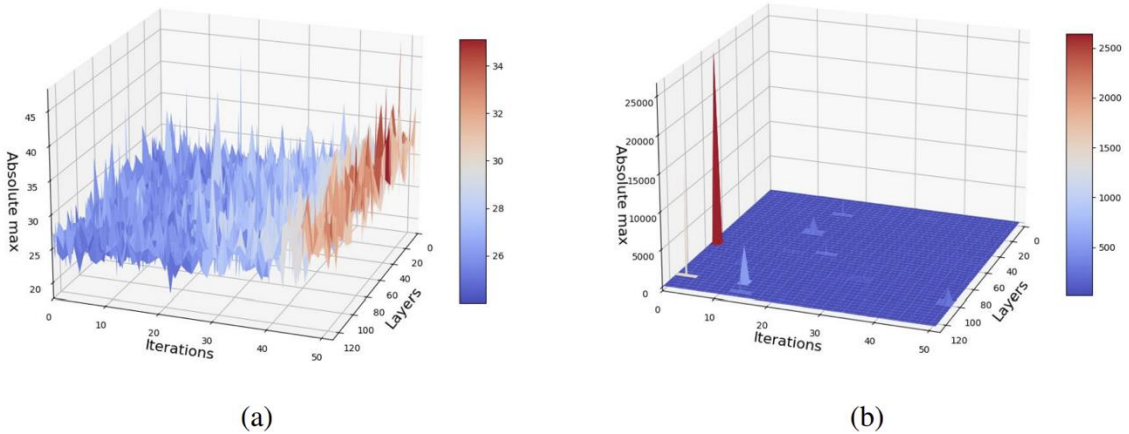
    ☺ Proposed method to **reduce losses** due to outliers in **activation** quantization

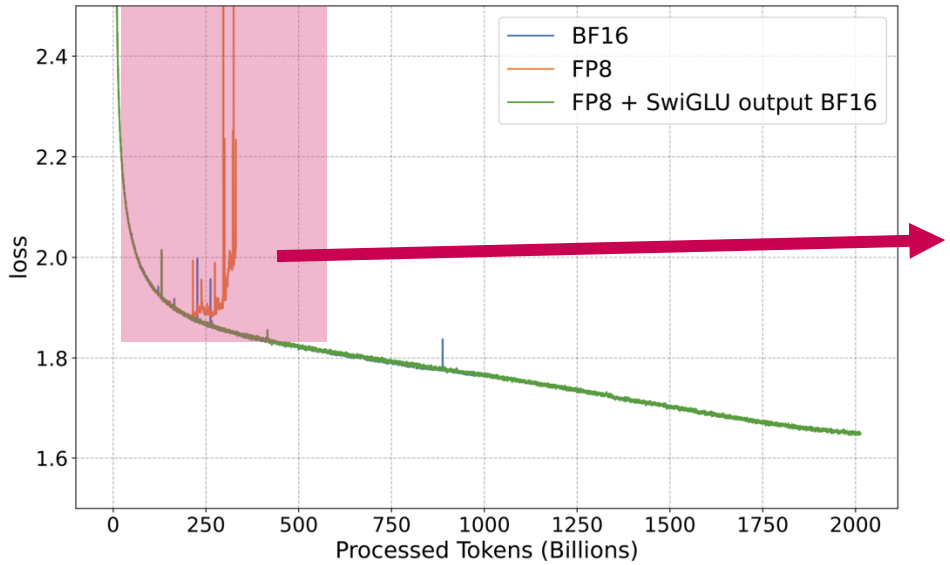
# Limitations


WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)

# Limitations

- Limited Training token (< 100B)




*FISHMAN, Maxim, et al. arXiv : 2409.12517 (2024)*

- Original LLaMA trainining token → 2 trillion tokens
- Large-token regime validation: required

☹ Scaling risks

→ **Model collapse· overfitting risk · generalization failure**

**FP8 training · SwiGLU** function-induced outlier spike **over 250B tokens**

→ **Divergence of training loss**

*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*

# Limitations

- Limited model scope
  - Experiments exclude Vision Transformers & outlier-prone LLMs
  - **Generality validation hindered**
- Unquantified sparse-path overhead in OCC
  - **Runtime · energy · memory costs** unmeasured, i.e. torch.sort + sparse GeMM
  - **Time-complexity** analysis absent
- Hyper-parameter (k, α) search cost
  - Empirical tuning across models → **High overhead**
  - **Sensitivity** analysis **lacking**
- No real-hardware evaluation
  - FP4 computation: **Simulation-only**
  - End-to-end cost reduction : **Quantification unavailable**


*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*

# Thank you


WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)

# Appendix

- Experiment setup
  - LLaMA 2 model (Primary model architecture)

Learning rate → **Warm-up** and **cosine decay** schedule

| Maximum learning rate: 0.0003, weight decay: 0.1 | |
|---|---|
| Warm-up | Increasing learning rate during the first 5% step |
| Cosine decay | The remaining 90% step is cosine decay |

** Hyper-parameters → **Consistent** across precision settings


*WANG, Ruizhe, et al. arXiv : 2501.17116 (2025)*