# DuQuant: Distributing Outliers via Dual Transformation Makes Stronger Quantized LLMs

NeurIPS 2024 Oral


Haokun Lin[*1,3,4], Haobo Xu[*2], Yichen Wu[*4], Jingzhi Cui[2], Yingtao Zhang[2], Linzhan Mou[5], Linqi Song[4], Zhenan Sun[^1,3], Ying Wei[^5]

[*]Equal Contribution [^]Corresponding Authors

[1]School of Artificial Intelligence, University of Chinese Academy of Sciences

[2]Tsinghua University [3]NLPR & MAIS, Institute of Automation, Chinese Academy of Sciences

[4]City University of Hong Kong [5]Zhejiang University

Haokun Lin

haokun.lin@cripac.ia.ac.cn

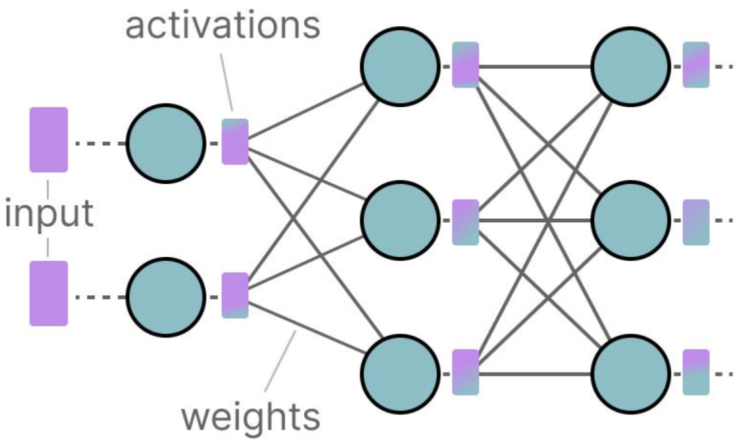

Project: https://duquant.github.io/

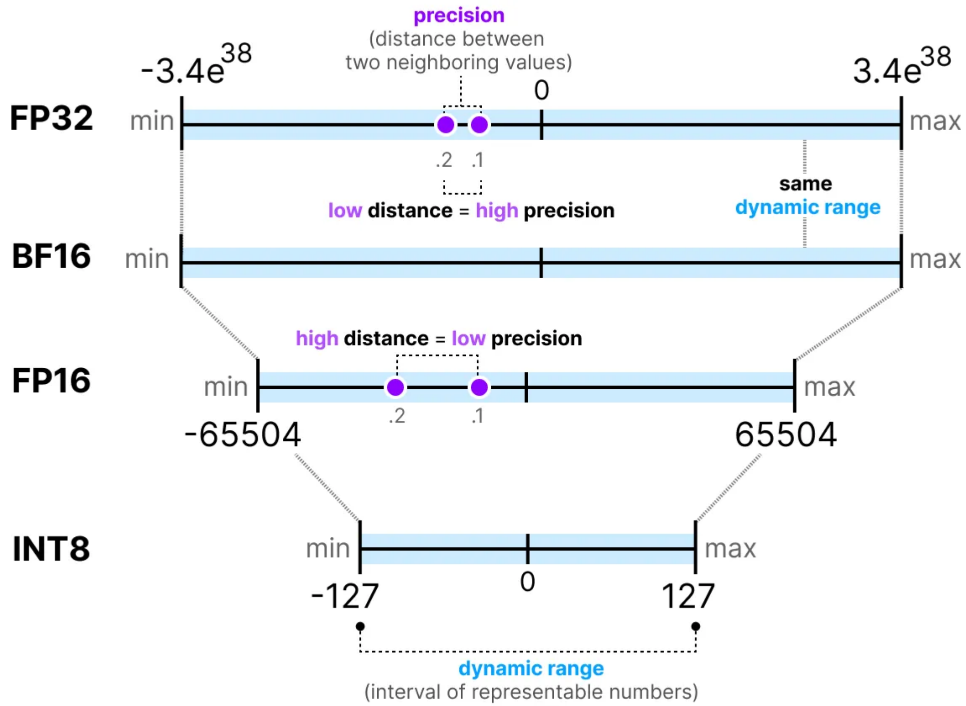
# Outline

# Network Quantization

- Network Quantization
  - Reduce redundancy in network representation
  - FP16 --- Low bits storage

- What to quantize?
  - Weights W
  - Activations X
  - Gradients




- Quantization scale
  - Binarization: binarize to −1 or +1.
  - m-bit quantization: int4、int8

$$\text{memory} = \frac{\text{nr\_bits}}{8} \times \text{nr\_params}$$

Original model: $S$ FP32

Binary model: $\frac{S}{32}$

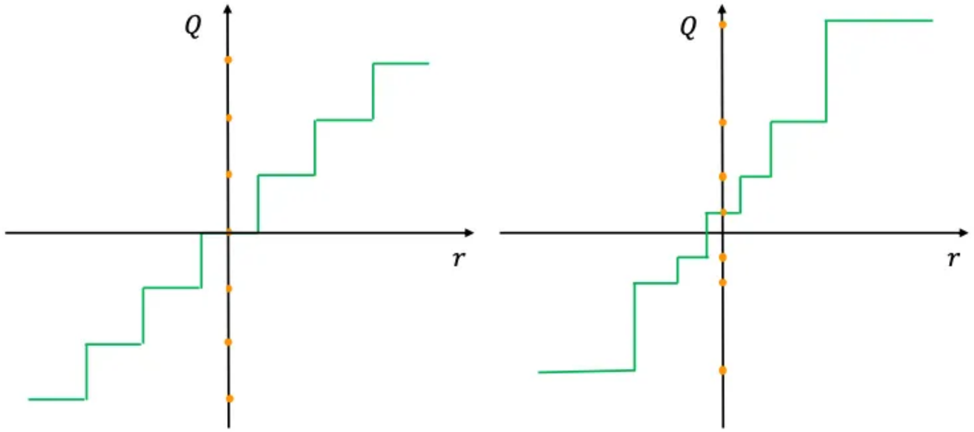
$m$-bit quantized model: $\frac{mS}{32}$

Smaller storage

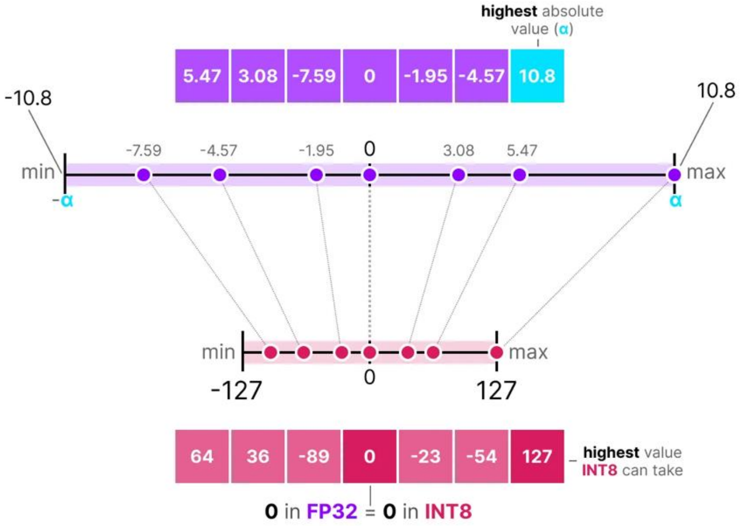
# Network Quantization

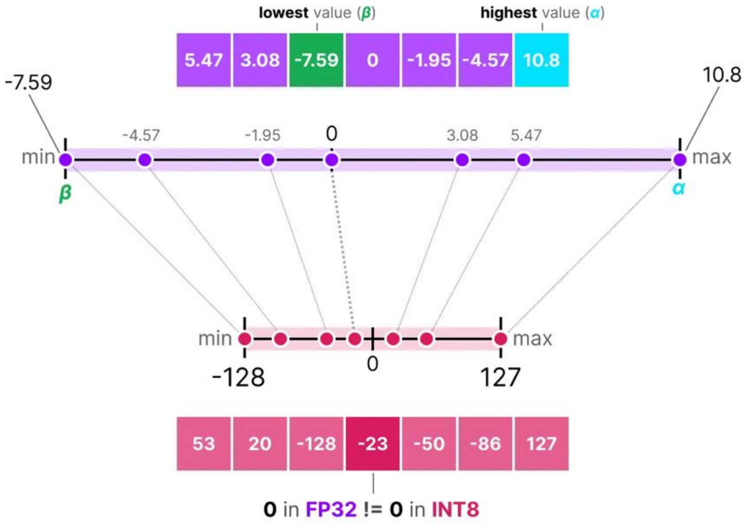
- PTQ
  - train a full-precision model
  - quantize with little or no data

- Uniform Quantization vs Non-Uniform Quantization




- Symmetric Quantization vs Asymmetric Quantization



# Network Quantization

- MinMax Quantization
  - $X$: Float format
  - $X_q$: Int format
  - $\Delta$: Scaling factor
  - $z$: Zero point

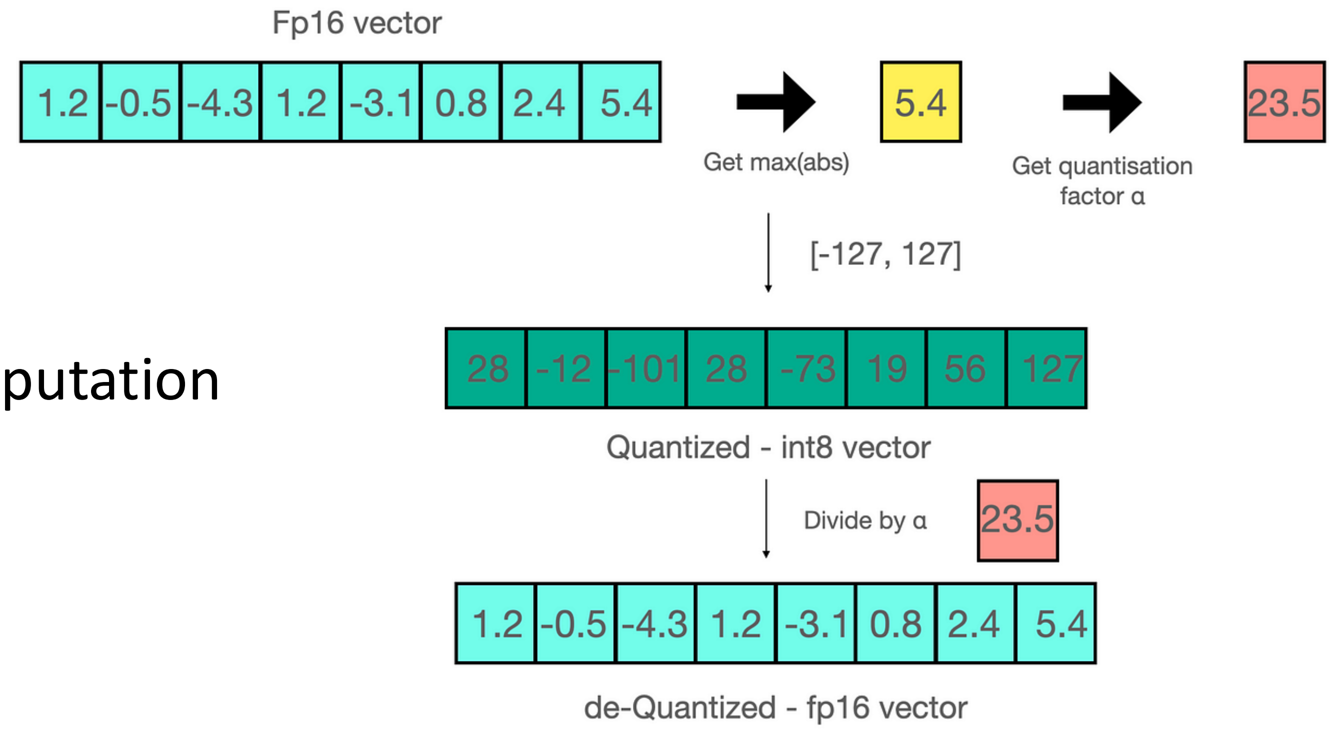
$$Quant: \mathbf{X}_q = \text{clamp}\left(\left\lfloor \frac{\mathbf{X}}{\Delta} \right\rceil + z, 0, 2^b - 1\right) \quad De\text{-}quant: \hat{\boldsymbol{X}} = s\left(\mathbf{X}_q - z\right) \approx \boldsymbol{X}$$

Rounding Function
Nearest Rounding

$$\Delta = \frac{\max(\mathbf{X}) - \min(\mathbf{X})}{2^b - 1}, z = -\left\lfloor \frac{\min(\mathbf{X})}{\Delta} \right\rceil$$

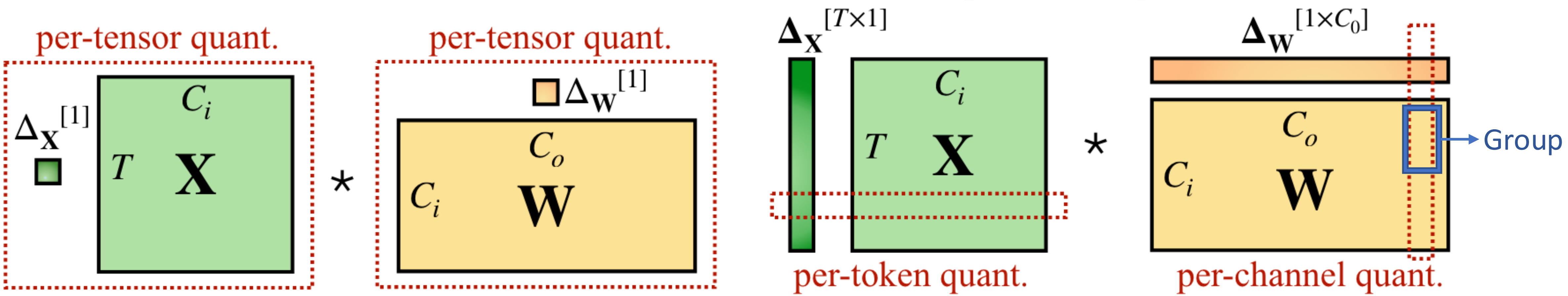
- INT8 symmetric quantization

- Fake Quantization
  - Dequantize to FP16 for computation



# Network Quantization

- Quantization scale
  - Per-tensor: one matrix has one zero-point and scaling factor
  - Per-channel: each output channel has one zero-point and scaling factor
  - Per-token: token-level for activation
  - Fine-grained group wise: divide the channel to small groups
- For DuQuant
  - Per-token for activation and per-channel for wight in WA setting
  - Quantize all activations including KV caches



(a) per-tensor quantization (b) per-token + per-channel quantization

# Outline

# SmoothQuant

## ➢ Normal Outliers

- Channels in the activation map whose magnitudes are obviously larger than other channels
- Outliers occur for almost all sequence dimensions (tokens) but are limited to specific feature/hidden dimensions.
- Outliers makes the quantization difficult

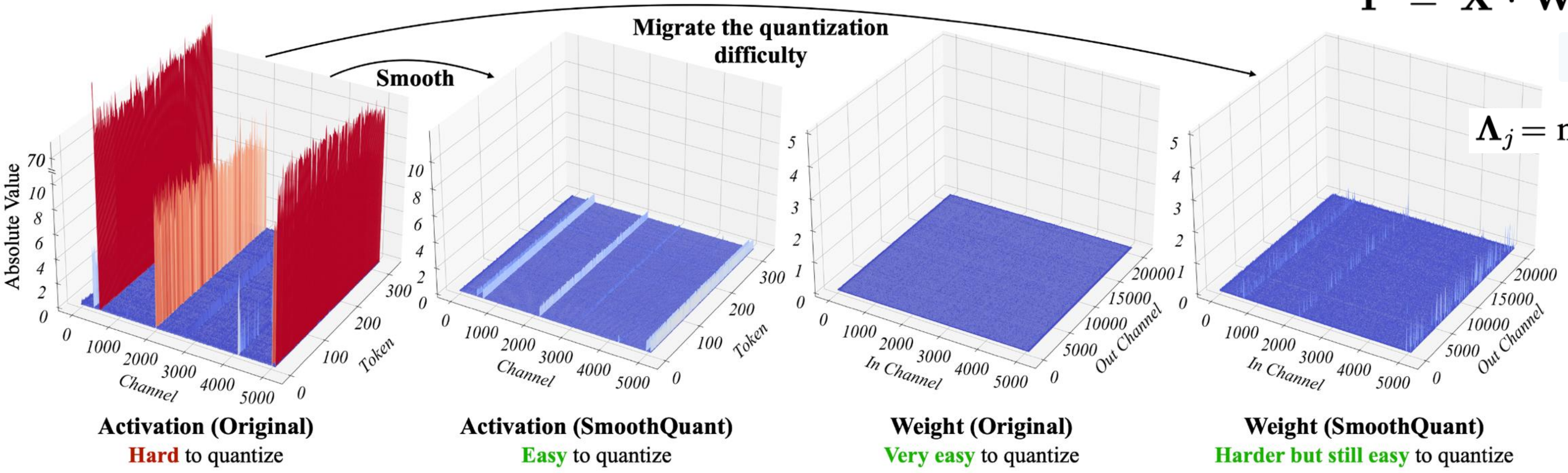
$$s = \frac{\max(\boldsymbol{X}) - \min(\boldsymbol{X})}{2^b - 1}, \quad z = \left\lceil -\frac{\min(\boldsymbol{X})}{s} \right\rfloor$$

- SmoothQuant propose to transfer the quantization difficulty from activations to model weights

$$\mathbf{Y} = \mathbf{X} \cdot \mathbf{W} = (\mathbf{X} \cdot \mathbf{\Lambda})(\mathbf{\Lambda}^{-1} \cdot \mathbf{W})$$

diagonal matrix

$$\mathbf{\Lambda}_j = \max(|\mathbf{X}_j|)^{\alpha} / \max(|\mathbf{W}_j|)^{1-\alpha}$$





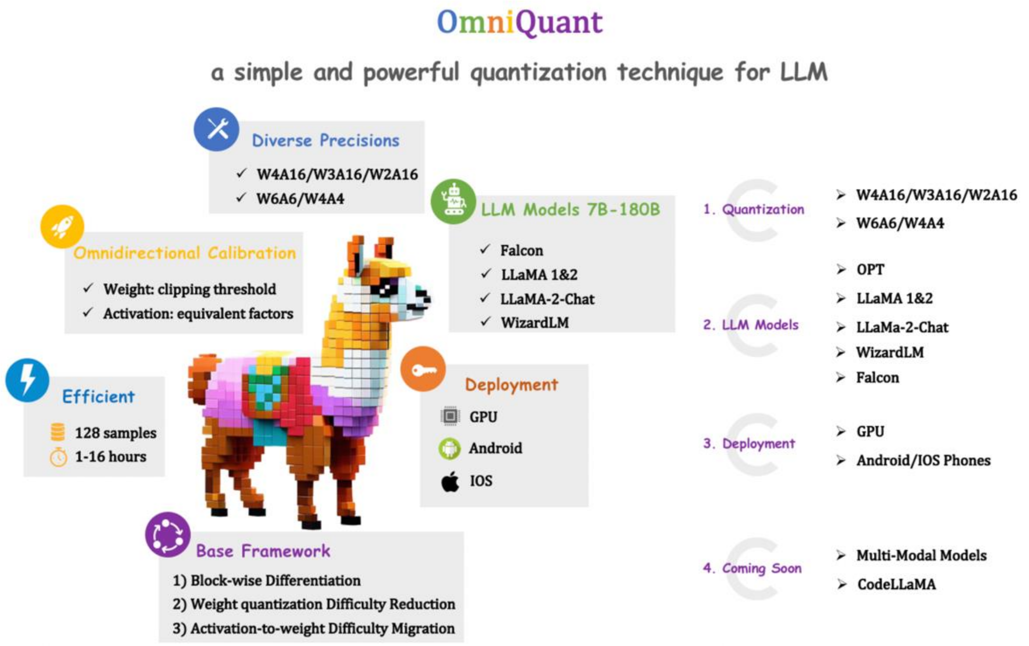

[1]. Xiao G, Lin J, Seznec M, et al. Smoothquant: Accurate and efficient post-training quantization for large language models[C] International Conference on Machine Learning. PMLR, 2023: 38087-38099.

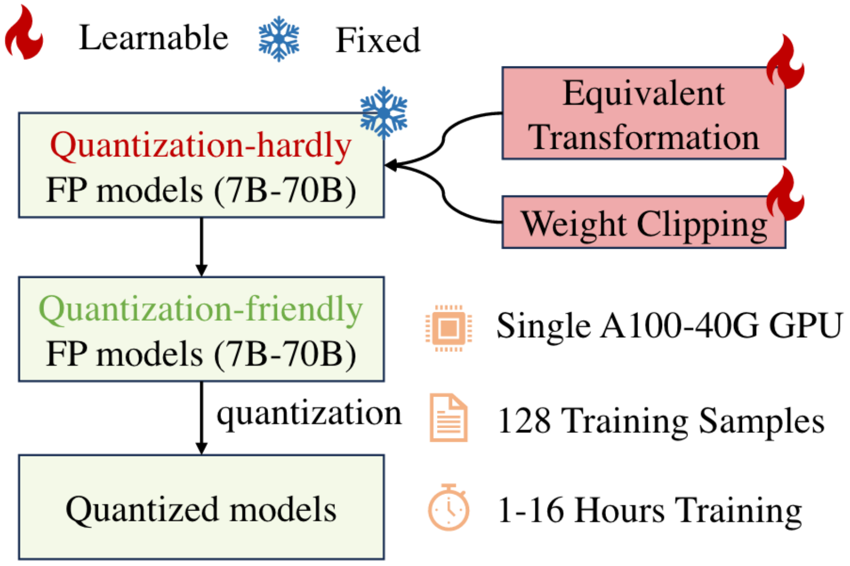
# OmniQuant

## ➢Two Learnable Modules

- Make the parameters of quantization learnable using a few samples under PTQ settings
- Learnable Weight Clipping (LWC) and Learnable Equivalent Transformation (LET).
- Formulate a block-wise quantization pipeline for LLM

$$\arg\min_{\Theta_1,\Theta_2} ||\mathcal{F}(\mathbf{W},\mathbf{X}) - \mathcal{F}\big(Q_w(\mathbf{W};\Theta_1,\Theta_2), Q_a(\mathbf{X},\Theta_2)\big)||$$




[1]. Shao, Wenqi, et al. "OmniQuant: Omnidirectionally Calibrated Quantization for Large Language Models." The Twelfth International Conference on Learning Representations.

# Learnable Weight Clipping

- Reduce the difficulty of quantizing weights in LLM
- N --- target bit, W --- full precision weight, $W_q$ --- quantized weight
- $h$ --- scaling factor, $z$ --- zero point
- Learnable clipping strengths for the upper and the lower bound of weights

$$\gamma \in [0,1] \text{ and } \beta \in [0,1] \quad \text{----->} \quad \Theta_1 = \{\gamma, \beta\}$$

$$\mathbf{W_q} = \mathrm{clamp}(\lfloor \frac{\mathbf{W}}{h} \rceil + z, 0, 2^N - 1), \text{where } h = \frac{\gamma \max(\mathbf{W}) - \beta \min(\mathbf{W})}{2^N - 1}, z = -\lfloor \frac{\beta \min(\mathbf{W})}{h} \rceil$$

$$\arg \min_{\Theta_1, \Theta_2} ||\mathcal{F}(\mathbf{W}, \mathbf{X}) - \mathcal{F}\big(Q_w(\mathbf{W}; \Theta_1, \Theta_2), Q_a(\mathbf{X}, \Theta_2)\big)||$$

$$\gamma = 1 \text{ and } \beta = 1$$ MinMax quantization

# Learnable equivalent transformation

- Migrate the difficulty of quantization from activations to weights with a mathematically equivalent transformation by channel-wise scaling and channel-wise shifting
- T --- token sequence length, $X$ --- input, $W$ --- weight, $B$ --- bias

$$\mathbf{X} \in \mathbb{R}^{T \times C_{in}} \qquad \mathbf{W} \in \mathbb{R}^{C_{in} \times C_{out}} \qquad \mathbf{B} \in \mathbb{R}^{1 \times C_{out}}$$

$$\mathbf{Y} = \mathbf{X}\mathbf{W} + \mathbf{B} = \underbrace{[(\mathbf{X} - \delta) \oslash s]}_{\tilde{\mathbf{X}}} \cdot \underbrace{[s \odot \mathbf{W}]}_{\tilde{\mathbf{W}}} + \underbrace{[\mathbf{B} + \delta\mathbf{W}]}_{\tilde{\mathbf{B}}}$$

- Y --- output

$$\mathbf{s} \in \mathbb{R}^{1 \times C_{in}} \text{ and } \delta \in \mathbb{R}^{1 \times C_{in}}$$

$\tilde{\mathbf{X}}, \tilde{\mathbf{W}}$ and $\tilde{\mathbf{B}}$ Equivalent activation, weight and bias

- Quantization pipeline

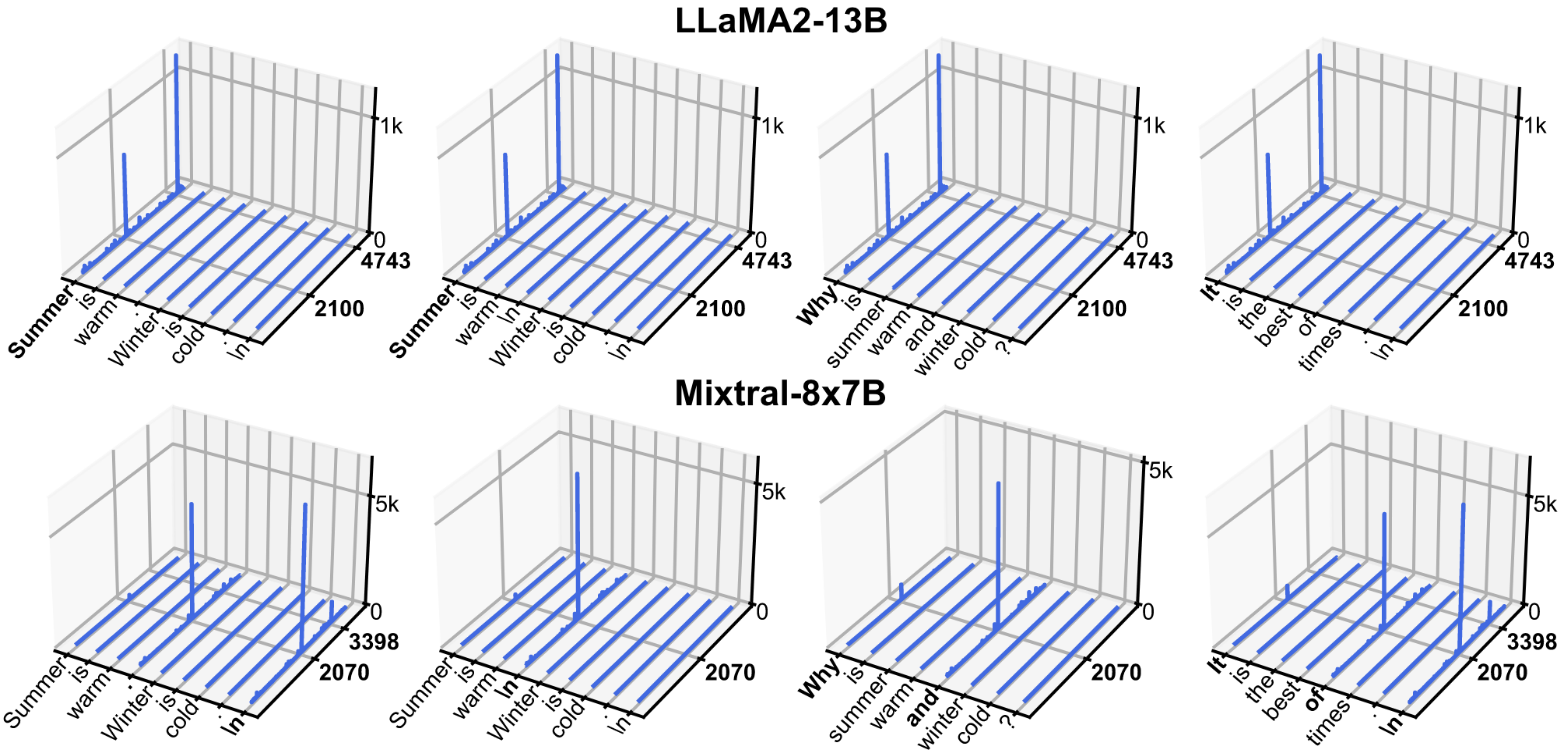
$$\mathbf{Y} = Q_a(\tilde{\mathbf{X}})Q_w(\tilde{\mathbf{W}}) + \widetilde{\mathbf{B}},$$

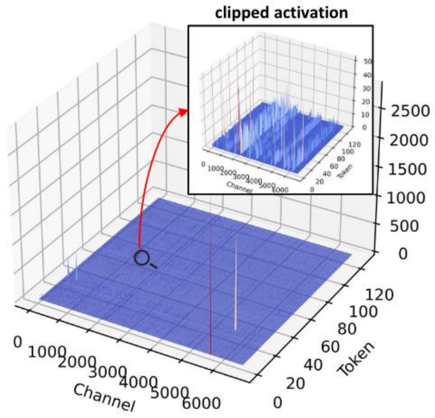
$Q_a$ --- MinMax quantization $\quad Q_w$ --- MinMax quantization + LWC

# Outlier

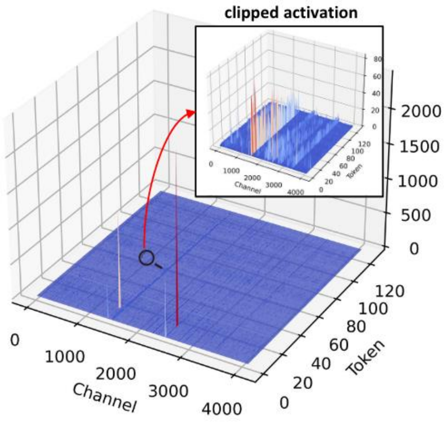
## ➢ Massive Outliers

- Very few activations exhibit significantly larger values than others (e.g., 100,000 times larger)
- Massive outliers are consistently present in very few fixed token dimensions
- Locations of Massive outliers (layer output): usually the starting tokens (e.g., [BOS] token)



(a) Output activations of LLaMA-30B Layer 24



(b) Output activations of LLaMA-2-7B Layer 24

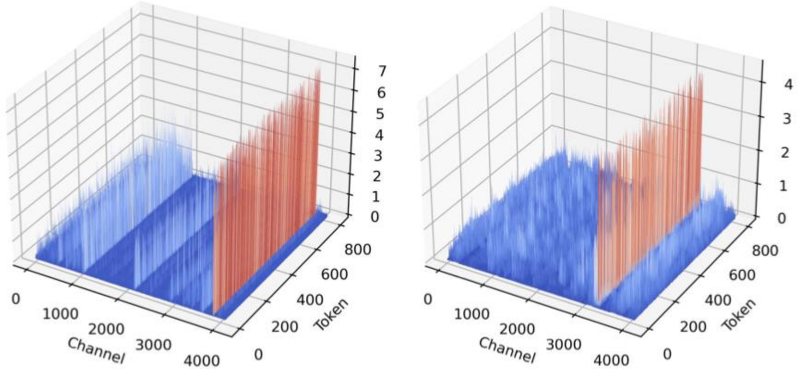
[1]. Sun M, Chen X, Kolter J Z, et al. Massive activations in large language models[J]. arXiv preprint arXiv:2402.17762, 2024.
[2]. Liu R, Bai H, Lin H, et al. IntactKV: Improving Large Language Model Quantization by Keeping Pivot Tokens Intact[J]. arXiv preprint arXiv:2403.01241, 2024.

# Outlier

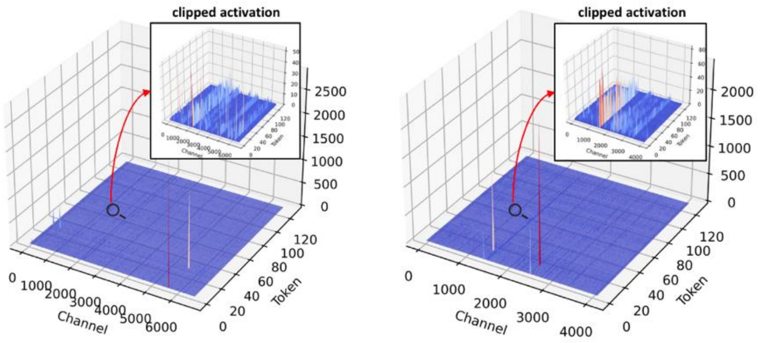
## ➢ Massive Outliers vs Normal Outliers

- Normal: large values across specific feature dimensions and present in all token sequences
- Massive: exceedingly high values and occur in a subset of tokens

Normal




(a) LLaMA7B_Layer2_Attn_K_Proj (b) LLaMA7B_Layer15_Attn_O_Proj




(a) Output activations of LLaMA-30B Layer 24 (b) Output activations of LLaMA-2-7B Layer 24

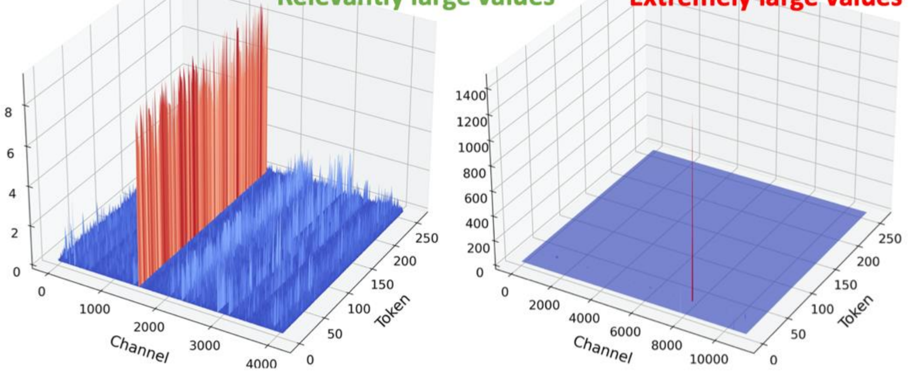
Massive

## ➢ Our Observations

- Massive Outliers Exist at the Second Linear Layer (Down Projection) of FFN Module
- We first discover this phenomenon, previous works only focus on layer output

All tokens,
Relevantly large values

Limited tokens,
Extremely large values



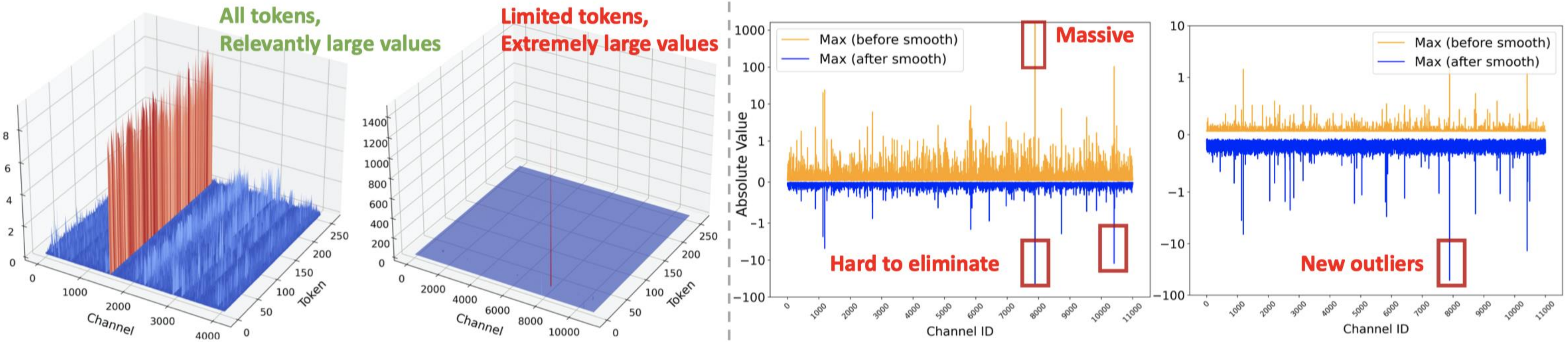

(a) LLaMA2_7B_Layer1_Attn_K_Proj (b) LLaMA2_7B_Layer1_FFN_Down_Proj

**Normal Outliers** **Massive Outliers**

# Outlier

## ➢ Our Observations

- Massive Outliers Exist at the Second Linear Layer (Down Proj) of FFN Module
- Traditional Methods fail to eliminate these massive outliers
  - SmoothQuant: cause the weights of the down-projection to display noticeable outliers
  - OmniQuant and AffineQuant: optimization-based methods to encounter problems with loss explosion



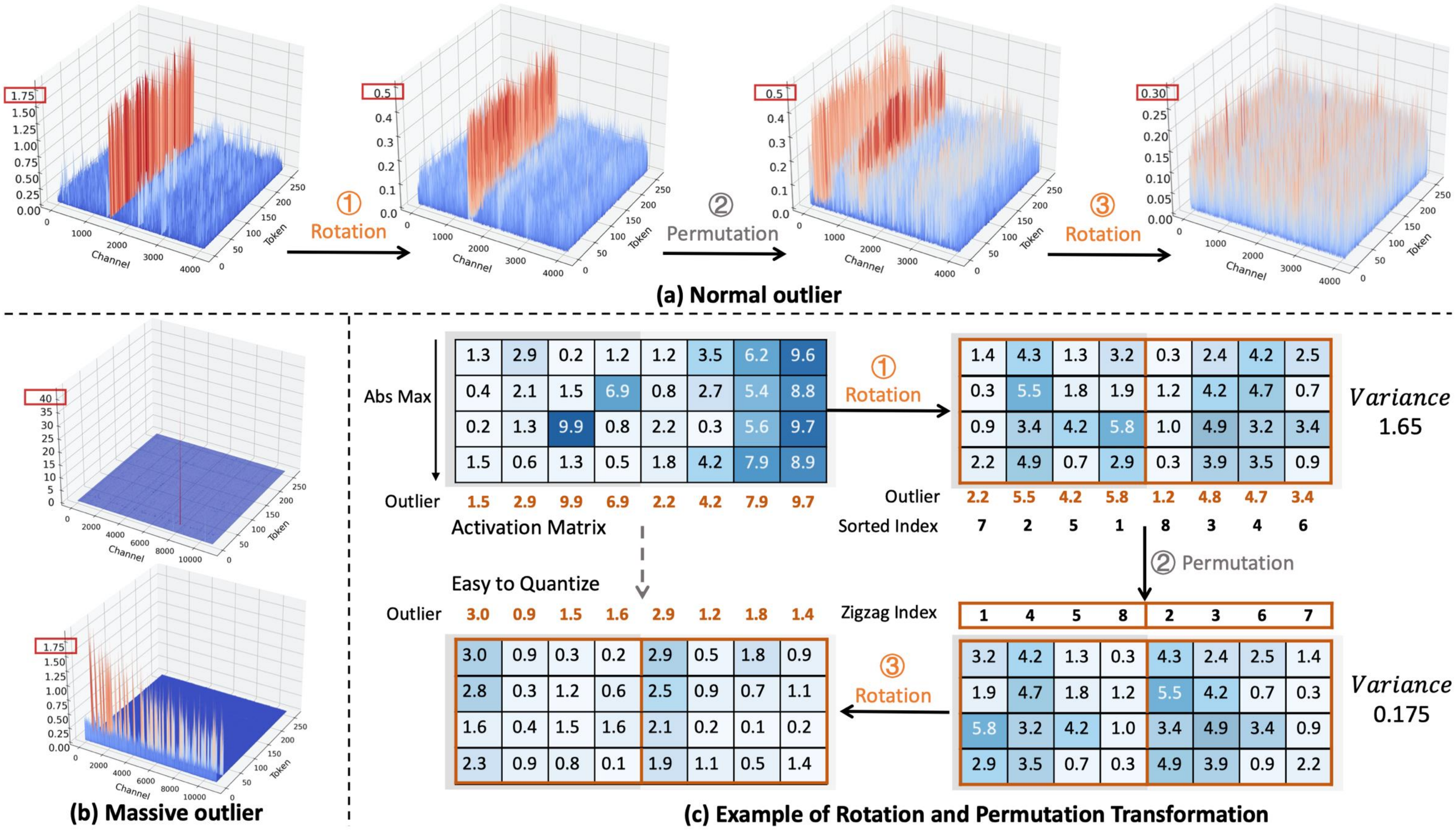

(a) LLaMA2_7B_Layer1_Attn_K_Proj (b) LLaMA2_7B_Layer1_FFN_Down_Proj (c) Activation Change with SmoothQuant (d) Weight Change with SmoothQuant

**Normal Outliers** **Massive Outliers** **SmoothQuant Fails to Eliminate Massive Outliers**

**How to eliminate both Normal and Massive outliers?**

# Outline

# DuQuant



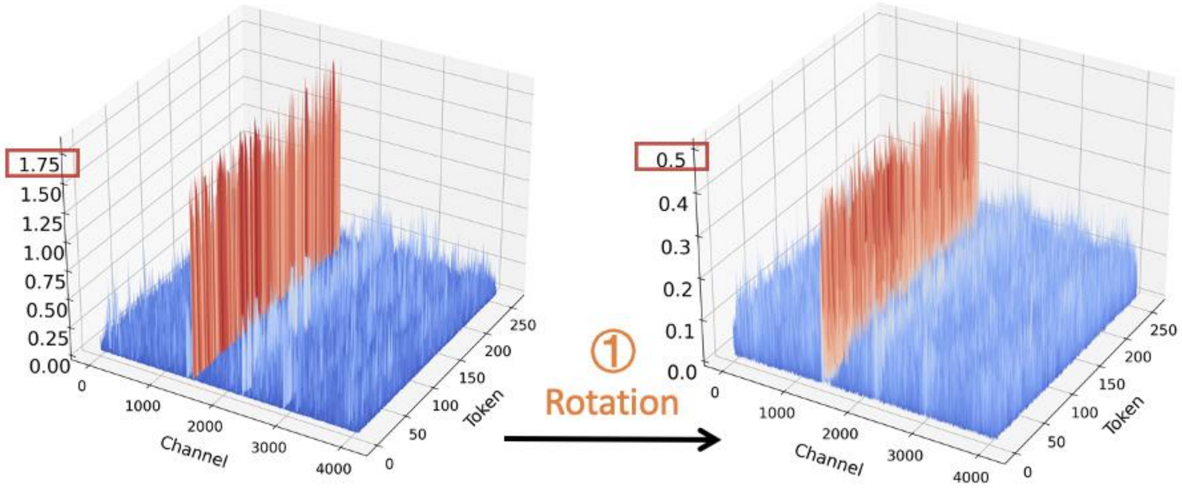
① Rotation
② Permutation
③ Rotation
Channel
Token
(a) Normal outlier
(b) Massive outlier
Abs Max
Outlier
Activation Matrix
Sorted Index
Zigzag Index
Easy to Quantize
Variance
1.65
Variance
0.175
(c) Example of Rotation and Permutation Transformation

# Outline

# Rotation

## ➢Motivation

- Use rotation matrix to distribute the outliers to adjacent channels
- Ideal rotation matrix $\mathbf{R}$
  - Orthogonal $\mathbf{R}\mathbf{R}^\top = \mathbf{I} \quad |\mathbf{R}| = \pm 1$
  - Target the positions of outliers and mitigate them through matrix multiplication




## ➢Rotation with prior knowledge

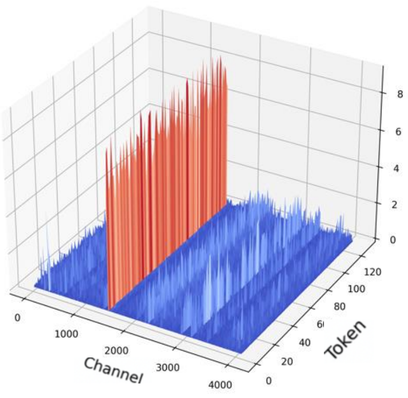
- Use greedy search with prior knowledge (the feature dimension of outlier) to compute a rotation matrix $\hat{\mathbf{R}}$

# Rotation

## ➢Rotation with prior knowledge

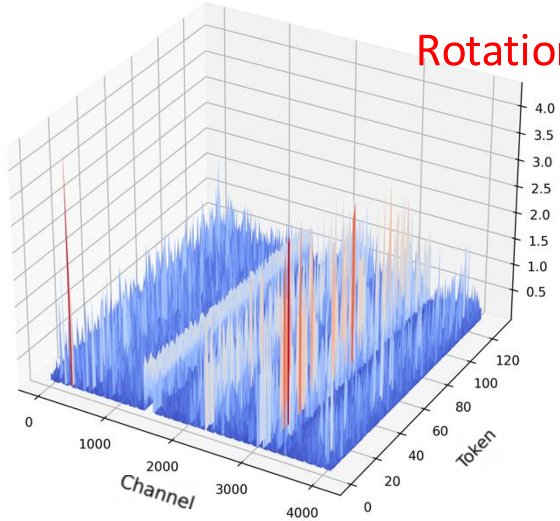
- Use greedy search with prior knowledge (the feature dimension of outlier) to compute a rotation matrix $\hat{\mathbf{R}}$
- The feature dimension $d^{(1)} = \arg\max_j(\max_i |\mathbf{X}_{ij}|)$
- Construct the rotation matrix by:

$$\mathbf{R}^1 = \mathbf{E}_{d^{(1)}} \tilde{\mathbf{R}} \mathbf{Q} \mathbf{E}_{d^{(1)}}, \qquad \mathbf{Q} = \begin{bmatrix} 1 & \mathbf{O} \\ \mathbf{O} & \mathbf{Q}' \end{bmatrix}$$




- $\tilde{\mathbf{R}}$ : an orthogonal initialized rotation matrix, first row is specifically uniformly distributed
- $\mathbf{E}_{d^{(1)}}$ : switching matrix used to swap the first and the $d^{(1)}$ column of the activation
- $\tilde{\mathbf{R}}$ can mitigate outliers in the first column after the transformation by $\mathbf{E}_{d^{(1)}}$
- $\mathbf{Q}$ : further increase the randomness of the rotation operation, $\mathbf{Q}'$ is a random orthogonal matrix
- Greedy search for $N$ steps (once rotation may induce new outliers)

$$\hat{\mathbf{R}} = \mathbf{R}^1\mathbf{R}^2 \cdots \mathbf{R}^n \qquad n = \arg\min_{k\in[1:N]} \left(\max_{i,j} |(\mathbf{X}\mathbf{R}^1 \cdots \mathbf{R}^k)_{ij}|\right)$$




## ➢Block-wise rotation

- For time and memory efficiency, we use block-wise rotation matrix

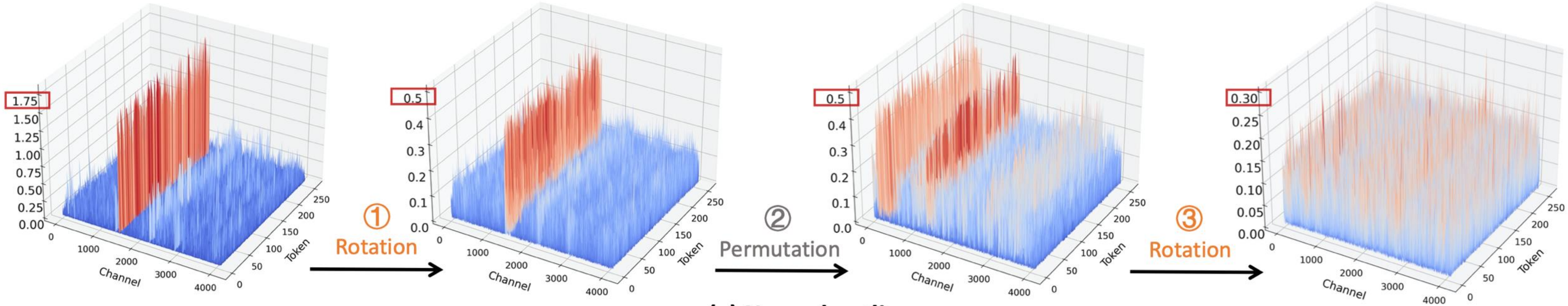
$$\hat{\mathbf{R}} = \text{BlockDiag}(\hat{\mathbf{R}}_{b_1}, ..., \hat{\mathbf{R}}_{b_K}) \quad \hat{\mathbf{R}} \in \mathbb{R}^{C_{in}\times C_{in}} \quad \hat{\mathbf{R}}_{b_i} \in \mathbb{R}^{2^n\times 2^n}$$

# Outline

# Permutation

## ➢Limitation of Rotation

- Block-wise rotation: uneven outlier magnitudes across different blocks
- Measurement: Compute the variance of different blocks $\mathrm{Var}([M_{b_1}, M_{b_2}, ..., M_{b_K}])$
  - For $i$ block, the $M_{b_i}$ represents the mean values of all $O_j$, $O_j$ is the largest outlier in dimension $d_j$




**(a) Normal outlier**

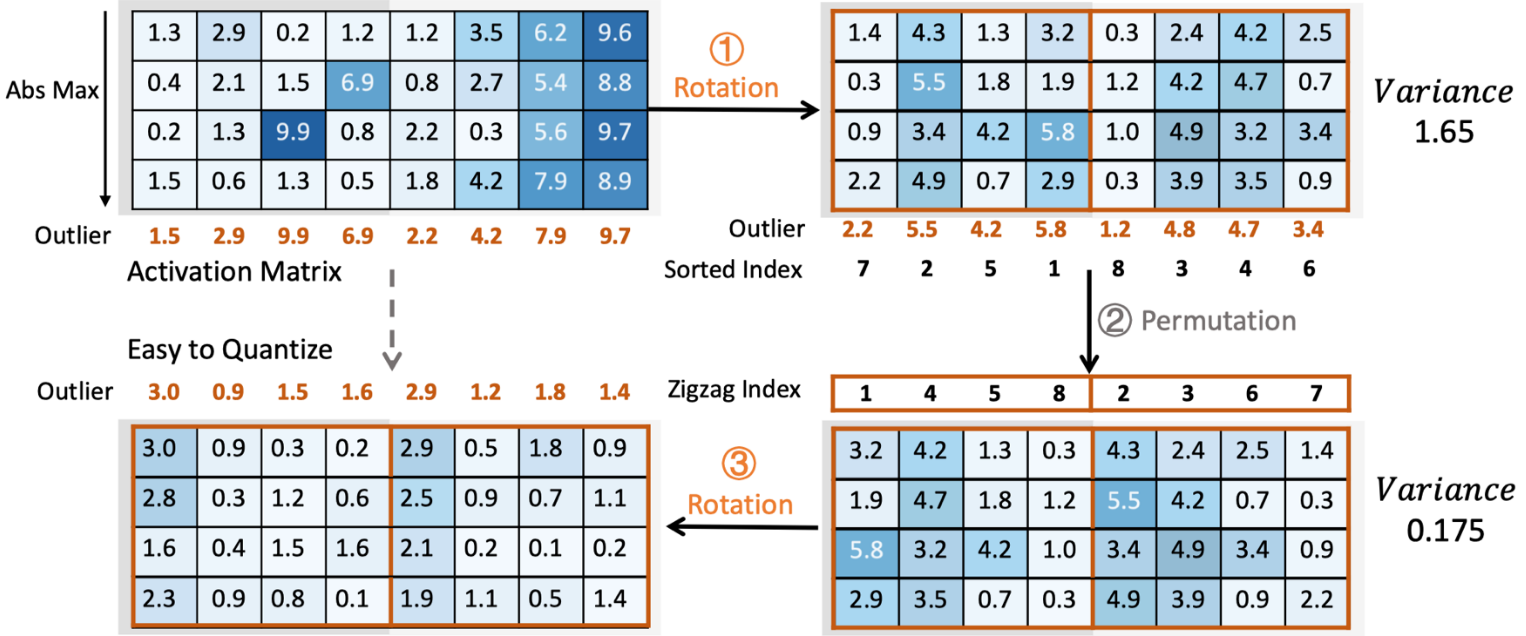
## ➢Solution

- Channel permutation to balance the distribution of outliers across blocks
- Permutation transformation is also orthogonal, denote as $\mathbf{P}$
- After permutation, employ another rotation transformation to further smooth the activations

# Zigzag Permutation

## ➢Zigzag Order

- Distribute the channels with the highest activations across the blocks in a back-and-forth pattern
- Fast with strong performance




| | LLaMA2-7B | | | | LLaMA2-13B | | | |
|---|---|---|---|---|---|---|---|---|
| Permutation Method | WikiText2 ↓ | C4 ↓ | Variance | Time/s | WikiText2 ↓ | C4 ↓ | Variance | Time/s |
| w.o. Permutation | 7.92 | 10.64 | 3.9e-2 | 27.5 | 5.96 | 7.94 | 3.1e-2 | 44.7 |
| Random | 6.40 | 8.08 | 4.9e-3 | 89.5 | 5.43 | 7.07 | 3.9e-3 | 148.6 |
| Simulated Annealing | 6.26 | 7.89 | 1.7e-4 | 769.6 | 5.42 | 7.06 | 1.5e-4 | 1257.8 |
| Zigzag | 6.28 | 7.90 | 3.0e-4 | 48.6 | 5.42 | 7.05 | 2.5e-4 | 74.0 |

# DuQuant

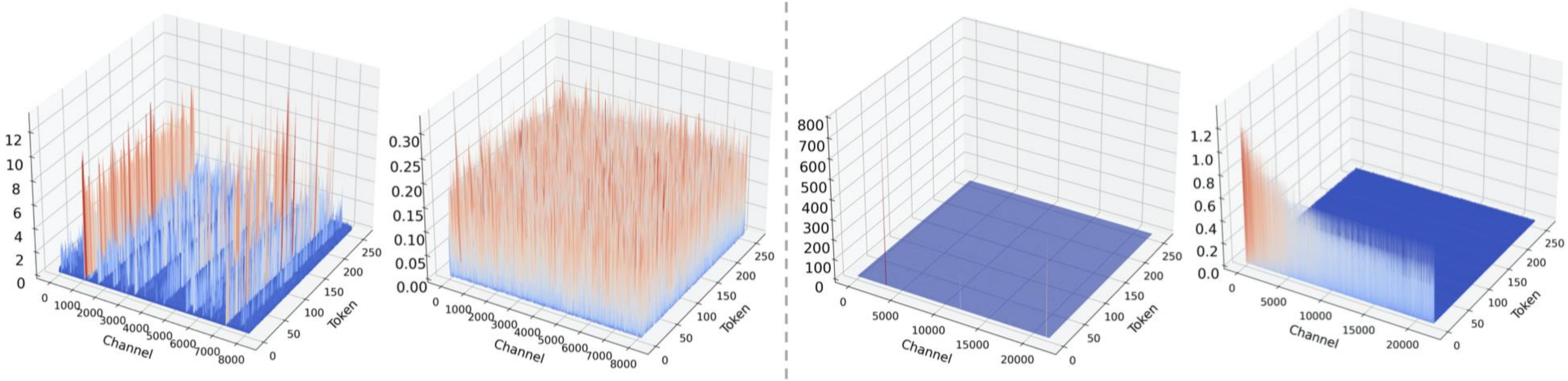
## ➢Linear Layer

- Smooth techniques (SmoothQuant)
- Block-wise Rotation (block size: 128)
- Permutation along with second Rotation

Remark: DuQuant simultaneously smooth the weight

$$\mathbf{Y} = \mathbf{X} \cdot \mathbf{W} = [(\mathbf{X} \cdot \underbrace{\mathbf{\Lambda})\hat{\mathbf{R}}_{(1)} \cdot \mathbf{P} \cdot \hat{\mathbf{R}}_{(2)}}_{\mathbf{G}}] \cdot [\underbrace{\hat{\mathbf{R}}_{(2)}^{\top} \cdot \mathbf{P}^{\top} \cdot \hat{\mathbf{R}}_{(1)}^{\top}(\mathbf{\Lambda}^{-1}}_{\mathbf{G}^{-1}} \cdot \mathbf{W})]$$

## ➢Visualization




(a) LLaMA1_65B_Layer11_Attn_K_Proj (b) After Rotation and Permutation

**Normal Outliers**

(c) LLaMA1_65B_Layer2_FFN_Down_Proj (d) After Rotation and Permutation

**Massive Outliers**

# Outline

# Experiments

- **DuQuant: Rotation twice, Permutation once**
  - **LWC**: adjusts weights by training parameters $\gamma, \beta \in [0,1]$ to compute the step size $\Delta = \frac{\gamma \max(\mathbf{X}) - \beta \min(\mathbf{X})}{2^b - 1}$
- **Models: LLaMA1, LLaMA2, LLaMA3, Vicuna, Mistral**
- **Tasks: Language generation (PPL), Commonsense QA, MMLU, MT-Bench, LongBench**

| Dataset | #Bit | Method | 1-7B | 1-13B | 1-30B | 1-65B | 2-7B | 2-13B | 2-70B |
|---|---|---|---|---|---|---|---|---|---|
| WikiText2 | FP16 | - | 5.68 | 5.09 | 4.10 | 3.53 | 5.47 | 4.88 | 3.31 |
| | W4A4 | SmoothQuant | 25.25 | 40.05 | 192.40 | 275.53 | 83.12 | 35.88 | 26.01 |
| | | OmniQuant | 11.26 | 10.87 | 10.33 | 9.17 | 14.26 | 12.30 | NaN |
| | | AffineQuant | 10.28 | 10.32 | 9.35 | - | 12.69 | 11.45 | - |
| | | QLLM | 9.65 | 8.41 | 8.37 | 6.87 | 11.75 | 9.09 | 7.00 |
| | | Atom | 8.15 | 7.43 | 6.52 | 5.14 | 8.40 | 6.96 | NaN |
| | | **DuQuant** | 6.40 | 5.65 | 4.72 | 4.13 | 6.28 | 5.42 | 3.79 |
| | | **DuQuant+LWC** | **6.18** | **5.47** | **4.55** | **3.93** | **6.08** | **5.33** | **3.76** |
| C4 | FP16 | | 7.08 | 6.61 | 5.98 | 5.62 | 6.97 | 6.46 | 5.52 |
| | W4A4 | SmoothQuant | 32.32 | 47.18 | 122.38 | 244.35 | 77.27 | 43.19 | 34.61 |
| | | OmniQuant | 14.51 | 13.78 | 12.49 | 11.28 | 18.02 | 14.55 | NaN |
| | | AffineQuant | 13.64 | 13.44 | 11.58 | - | 15.76 | 13.97 | - |
| | | QLLM | 12.29 | 10.58 | 11.51 | 8.98 | 13.26 | 11.13 | 8.89 |
| | | Atom | 10.34 | 9.57 | 8.56 | 8.17 | 10.96 | 9.12 | NaN |
| | | **DuQuant** | 7.84 | 7.16 | 6.45 | 6.03 | 7.90 | 7.05 | 5.87 |
| | | **DuQuant+LWC** | **7.73** | **7.07** | **6.37** | **5.93** | **7.79** | **7.02** | **5.85** |

# Experiments

- ➢ **Models: LLaMA1, LLaMA2, LLaMA3, Vicuna, Mistral**
- ➢ **Tasks: Language generation (PPL), Commonsense QA, MMLU, MT-Bench, LongBench**

| Model | Method | PIQA | ARC-E | ARC-C | BoolQ | HellaSwag | WinoGrande | Avg. |
|---|---|---|---|---|---|---|---|---|
| LLaMA1-7B W4A4 | FP16 | 77.47 | 52.48 | 41.46 | 73.08 | 73.00 | 67.07 | 64.09 |
| | SmoothQuant | 49.80 | 30.40 | 25.80 | 49.10 | 27.40 | 48.00 | 38.41 |
| | OS+ | 62.73 | 39.98 | 30.29 | 60.21 | 44.39 | 52.96 | 48.43 |
| | OmniQuant | 66.15 | 45.20 | 31.14 | 63.51 | 56.44 | 53.43 | 52.65 |
| | AffineQuant | 69.37 | 42.55 | 31.91 | 63.73 | 57.65 | 55.33 | 53.42 |
| | QLLM | 68.77 | 45.20 | 31.14 | - | 57.43 | 56.67 | 51.84 |
| | Atom | 71.44 | 47.74 | 35.49 | 67.71 | 63.89 | 55.01 | 56.88 |
| | **DuQuant** | **76.44** | **50.04** | **38.99** | **70.98** | 69.39 | **64.72** | **61.76** |
| | **DuQuant+LWC** | 76.22 | **50.04** | 38.31 | 70.09 | **69.82** | 62.59 | 61.18 |
| LLaMA1-13B W4A4 | FP16 | 79.10 | 59.89 | 44.45 | 68.01 | 76.21 | 70.31 | 66.33 |
| | SmoothQuant | 61.04 | 39.18 | 30.80 | 61.80 | 52.29 | 51.06 | 49.36 |
| | OS+ | 63.00 | 40.32 | 30.38 | 60.34 | 53.61 | 51.54 | 49.86 |
| | OmniQuant | 69.69 | 47.39 | 33.10 | 62.84 | 58.96 | 55.80 | 54.37 |
| | AffineQuant | 66.32 | 43.90 | 29.61 | 64.10 | 56.88 | 54.70 | 52.58 |
| | QLLM | 71.38 | 47.60 | 34.30 | - | 63.70 | 59.43 | 55.28 |
| | Atom | 71.38 | 49.07 | 36.69 | 64.53 | 68.00 | 58.56 | 58.04 |
| | **DuQuant** | 77.26 | **58.04** | **41.55** | **67.55** | 73.62 | **66.69** | **64.12** |
| | **DuQuant+LWC** | **77.64** | 57.32 | 41.21 | 66.79 | **74.12** | 65.98 | 63.84 |
| LLaMA1-30B W4A4 | FP16 | 80.08 | 58.92 | 45.47 | 68.44 | 79.21 | 72.53 | 67.44 |
| | SmoothQuant | 58.65 | 35.53 | 27.73 | 60.42 | 35.56 | 48.06 | 44.83 |
| | OS+ | 67.63 | 46.17 | 34.40 | 60.70 | 54.32 | 52.64 | 52.62 |
| | OmniQuant | 71.21 | 49.45 | 34.47 | 65.33 | 64.65 | 59.19 | 56.63 |
| | AffineQuant | 70.84 | 49.41 | 37.12 | 70.12 | 65.53 | 58.64 | 58.61 |
| | QLLM | 73.83 | 50.67 | 38.40 | - | 67.91 | 58.56 | 57.87 |
| | Atom | 71.98 | 49.07 | 40.02 | 66.85 | 70.45 | 58.64 | 59.50 |
| | **DuQuant** | 78.56 | **56.99** | 42.32 | 66.73 | 76.70 | 69.61 | 65.15 |
| | **DuQuant+LWC** | **78.73** | 56.52 | **43.17** | **68.84** | **77.53** | **70.96** | **65.96** |
| LLaMA1-65B W4A4 | FP16 | 80.79 | 58.71 | 46.24 | 82.29 | 80.72 | 77.50 | 71.04 |
| | SmoothQuant | 64.47 | 40.44 | 29.82 | 59.38 | 39.90 | 52.24 | 47.71 |
| | OS+ | 68.06 | 43.98 | 35.32 | 62.75 | 50.73 | 54.30 | 52.52 |
| | OmniQuant | 71.81 | 48.02 | 35.92 | 73.27 | 66.81 | 59.51 | 59.22 |
| | QLLM | 73.56 | 52.06 | 39.68 | - | 70.94 | 62.90 | 59.83 |
| | Atom | 74.48 | 51.60 | 40.61 | 73.76 | 73.78 | 62.12 | 62.73 |
| | **DuQuant** | 79.71 | 57.95 | **45.05** | **79.82** | 78.66 | **72.29** | **68.91** |
| | **DuQuant+LWC** | **79.98** | **58.29** | 44.80 | 77.89 | **79.22** | 72.21 | 68.73 |

# Experiments

- Models: LLaMA1, LLaMA2, LLaMA3, Vicuna, Mistral
- Tasks: Language generation (PPL), Commonsense QA, MMLU, MT-Bench, LongBench

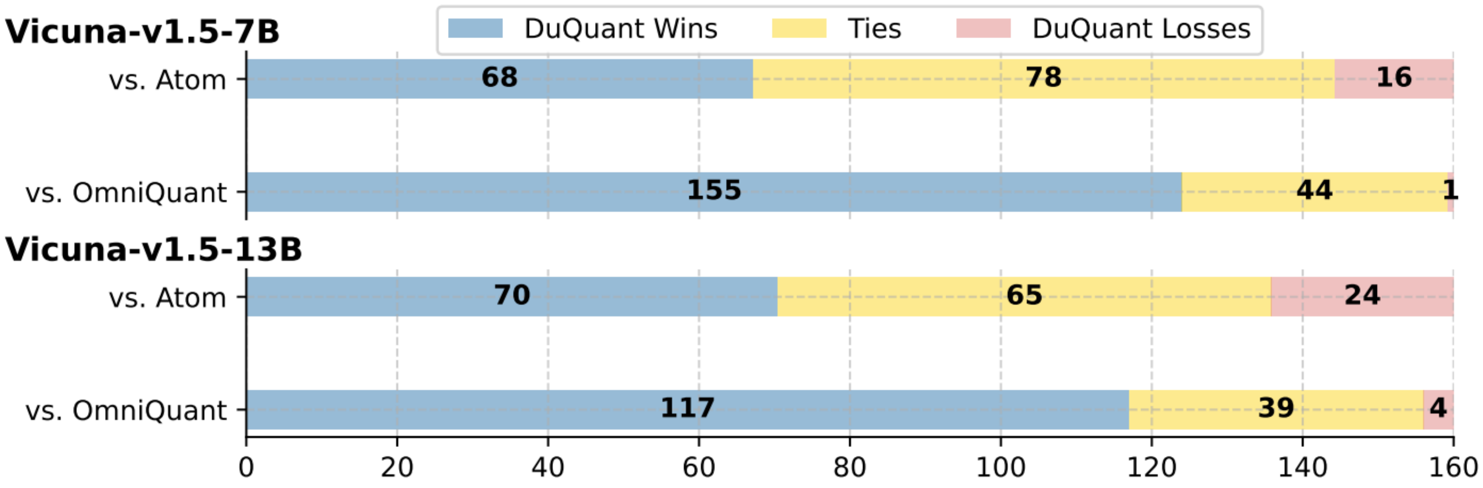
| #Bits | Method | WikiText2 ↓ | C4 ↓ | PTB ↓ | PIQA | ARC-E | ARC-C | BoolQ | HellaSwag | WinoGrande | Avg. ↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FP16 | - | 6.14 | 8.88 | 9.91 | 80.85 | 77.78 | 53.41 | 81.28 | 79.16 | 72.84 | 74.22 |
| LLaMA3-8B W6A6 | SmoothQuant | 7.07 | 9.57 | 11.69 | 78.94 | 75.88 | 49.49 | 77.58 | 77.39 | 70.8 | 71.68 |
| | OmniQuant | 7.24 | 9.82 | 11.90 | 78.90 | 73.95 | 47.35 | 74.95 | 76.77 | 70.56 | 70.41 |
| | AffineQuant | 7.35 | 9.99 | 12.30 | 78.73 | 73.32 | 46.08 | 74.59 | 77.08 | 70.88 | 70.11 |
| | **DuQuant** | **6.27** | **8.38** | **10.77** | **80.20** | 77.27 | 52.05 | **80.12** | **79.14** | 72.77 | 73.59 |
| | **DuQuant+LWC** | **6.27** | **8.38** | 10.78 | 79.71 | **77.57** | **53.07** | 80.00 | 78.70 | **73.09** | **73.69** |
| LLaMA3-8B W4A4 | SmoothQuant | 210.19 | 187.93 | 278.02 | 54.57 | 31.9 | 24.23 | 52.72 | 31.26 | 51.14 | 40.97 |
| | OmniQuant | 3.64e3 | 2.80e3 | 3.09e3 | 50.22 | 26.94 | 24.57 | 37.98 | 26.55 | 50.20 | 36.08 |
| | AffineQuant | 21.21e3 | 34.60e3 | 16.72e3 | 50.71 | 25.93 | 26.02 | 40.55 | 26.07 | 48.46 | 36.29 |
| | Atom | 22.14 | 31.83 | 40.04 | 62.95 | 49.45 | 30.12 | 60.31 | 53.75 | 56.04 | 52.10 |
| | **DuQuant** | 8.56 | 11.98 | 13.66 | 75.68 | 68.48 | 41.81 | 71.99 | 73.07 | 66.22 | 66.21 |
| | **DuQuant+LWC** | **8.06** | **11.29** | **13.19** | **76.22** | **70.41** | **43.69** | **74.34** | **73.87** | **67.80** | **67.72** |

| #Bits | Method | WikiText2 ↓ | C4 ↓ | PTB ↓ | PIQA | ARC-E | ARC-C | BoolQ | HellaSwag | WinoGrande | Avg. ↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FP16 | - | 2.9 | 6.9 | 8.2 | 82.4 | 86.9 | 60.3 | 85.2 | 84.9 | 80.6 | 80.1 |
| LLaMA3-70B W4A4 | SmoothQuant | 9.6 | 16.9 | 17.7 | 76.9 | 75.8 | 43.5 | 64.4 | 62.9 | 58.9 | 63.7 |
| | **DuQuant** | **4.9** | **8.3** | **8.7** | **81.1** | **80.8** | **57.3** | **81.3** | **82.1** | **77.0** | **76.6** |

# Experiments

- Models: LLaMA1, LLaMA2, LLaMA3, Vicuna, Mistral
- Tasks: Language generation (PPL), Commonsense QA, MMLU, MT-Bench, LongBench

| Model | Method | MMLU (0 shot) ↑ | | | | | MMLU (5 shot) ↑ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | STEM | Hums | Social | Others | Avg. | STEM | Hums | Social | Others | Avg. |
| Vicuna-v1.5-13B W4A4 | FP16 | 43.70 | 50.48 | 62.72 | 62.74 | 54.54 | 44.96 | 51.97 | 65.26 | 62.40 | 55.78 |
| | SmoothQuant | 21.70 | 24.29 | 22.13 | 23.16 | 22.82 | 25.31 | 24.97 | 26.00 | 27.08 | 25.84 |
| | OmniQuant | 26.81 | 26.57 | 30.35 | 28.75 | 28.12 | 28.79 | 27.29 | 31.13 | 28.99 | 29.05 |
| | Atom | 32.54 | 39.60 | 46.02 | 46.11 | 41.07 | 35.35 | 39.21 | 59.72 | 45.77 | 45.01 |
| | **DuQuant** | **40.82** | 46.61 | 58.73 | 57.59 | 50.94 | 40.92 | **48.78** | **60.42** | 57.71 | **51.96** |
| | **DuQuant+LWC** | 40.13 | **47.48** | **58.86** | **57.83** | **51.08** | **41..42** | 48.52 | 58.73 | **57.74** | 51.61 |




| DuQuant v.s. FP16 | Former Win | Tie | Former Loss |
|---|---|---|---|
| Vicuna-v1.5-7B | 36 | 56 | 68 |
| Vicuna-v1.5-13B | 43 | 53 | 64 |

# Experiments

- Models: LLaMA1, LLaMA2, LLaMA3, Vicuna, Mistral
- Tasks: Language generation (PPL), Commonsense QA, MMLU, MT-Bench, LongBench

| Vicuna-v1.5-7B | RepoBench-P | MultiFieldQA-en | GovReport | MultiNews | DuReader | 2WikiMQA | TriviaQA |
|---|---|---|---|---|---|---|---|
| FP16 | 48.23 | 38.30 | 27.93 | 26.91 | 25.53 | 18.02 | 82.59 |
| SmoothQuant | 25.92 | 4.66 | 2.62 | 6.05 | 4.24 | 2.02 | 1.62 |
| OmniQuant | 14.97 | 2.30 | 2.51 | 2.64 | 1.87 | 0.48 | 0.81 |
| Atom | 29.34 | 31.15 | 23.60 | 24.60 | 19.41 | **17.10** | 67.20 |
| **DuQuant** | **47.66** | **35.62** | **25.66** | **25.85** | **23.15** | 15.09 | **78.91** |

| Vicuna-v1.5-7B | QMSum | MultiFieldQA-zh | NarrativeQA | Qasper | SAMSum | TREC | Avg |
|---|---|---|---|---|---|---|---|
| FP16 | 21.07 | 32.56 | 14.96 | 23.27 | 41.06 | 66.00 | 35.88 |
| SmoothQuant | 2.00 | 0.88 | 1.75 | 4.11 | 1.55 | 15.00 | 5.57 |
| OmniQuant | 3.93 | 1.40 | 1.10 | 1.62 | 0.61 | 1.00 | 2.71 |
| Atom | 20.24 | 21.55 | 11.57 | 17.97 | 37.94 | 58.00 | 29.21 |
| **DuQuant** | **21.15** | **29.56** | **11.31** | **19.98** | **42.24** | **64.00** | **33.86** |

| Vicuna-v1.5-13B | RepoBench-P | MultiFieldQA-en | GovReport | MultiNews | DuReader | 2WikiMQA | TriviaQA |
|---|---|---|---|---|---|---|---|
| FP16 | 43.08 | 42.69 | 28.43 | 26.53 | 27.57 | 29.40 | 86.81 |
| SmoothQuant | 11.57 | 1.64 | 2.81 | 3.54 | 6.71 | 1.39 | 1.83 |
| OmniQuant | 8.46 | 4.32 | 0.74 | 2.83 | 13.83 | 0.75 | 1.13 |
| Atom | 37.31 | 37.31 | 19.34 | 23.39 | 21.79 | 15.16 | 80.75 |
| **DuQuant** | **38.09** | **44.12** | **26.97** | **26.59** | **26.02** | **22.07** | **83.04** |

| Vicuna-v1.5-13B | QMSum | MultiFieldQA-zh | NarrativeQA | Qasper | SAMSum | TREC | Avg |
|---|---|---|---|---|---|---|---|
| FP16 | 21.24 | 40.44 | 15.41 | 24.41 | 41.97 | 68.00 | 40.64 |
| SmoothQuant | 2.95 | 0.82 | 0.97 | 2.18 | 0.35 | 1.50 | 4.21 |
| OmniQuant | 1.78 | 1.06 | 0.62 | 0.68 | 0.45 | 9.00 | 4.58 |
| Atom | 20.23 | 28.02 | 8.81 | 17.67 | 38.72 | 59.00 | 33.58 |
| **DuQuant** | **20.72** | **30.85** | **13.36** | **18.93** | **42.67** | **66.50** | **38.13** |

1. Single-Document QA tasks:
   Qasper, MultiFieldQA, and NarrativeQA (F1 score)
2. Multi-Document QA tasks:
   DuReader (Rouge-L score) and 2WikiMultihopQA (F1 score)
3. Summarization task:
   MultiNews (Rouge-L score)
4. Few-shot Learning tasks:
   TREC (Accuracy CLS), TriviaQA (F1 score), and SAMSum (Rouge-L score)
5. Code Completion task:
   RepoBench-P (similarity score)

# Ablation

- Influence of different **components** in DuQuant

| Modules | | | | LLaMA2-7B | | LLaMA2-13B | |
|---|---|---|---|---|---|---|---|
| Smooth | Rotation 1 | Permutation | Rotation 2 | WikiText2 ↓ | C4 ↓ | WikiText2 ↓ | C4 ↓ |
| ✓ | | | | NaN | 1379.46 | 160.30 | 203.87 |
| | ✓ | | | 8.48 | 10.63 | 14.32 | 21.73 |
| ✓ | ✓ | | | 7.92 | 10.64 | 5.96 | 7.94 |
| | ✓ | ✓ | ✓ | 6.79 | 8.51 | 6.06 | 8.03 |
| ✓ | ✓ | ✓ | ✓ | **6.28** | **7.90** | **5.42** | **7.05** |

- Quantization **runtime** on single A100

| Model | Omni. | Affine. | QLLM | Atom | DuQuant |
|---|---|---|---|---|---|
| LLaMA2-7B | 2.0h | 9.1h | 1.1h | 20min | 50s |
| LLaMA2-13B | 3.2h | 16.0h | 1.7h | 36min | 71s |
| LLaMA2-70B | 14.6h | 18.6h | 9.3h | 3.5h | 270s |

- **Calibration-free**: use random data

| LLaMA2-7B | | Eval. WikiText2 ↓ | C4 ↓ |
|---|---|---|---|
| Calib. | Randomly Generated | 6.25 | 7.86 |
| | WikiText2 | 6.25 | 7.87 |

| LLaMA2-13B | | Eval. WikiText2 ↓ | C4 ↓ |
|---|---|---|---|
| Calib. | Randomly Generated | 5.45 | 7.05 |
| | WikiText2 | 5.44 | 7.05 |

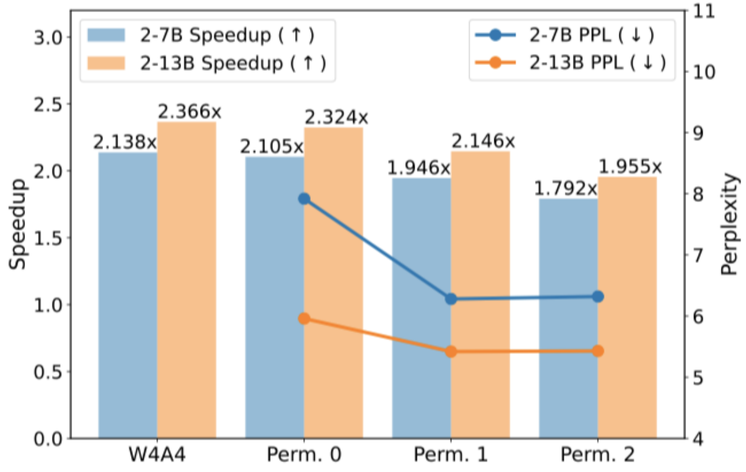
# Experiments

- Settings: LLaMA2-7B, Measure on RTX 3090, Input seq --- 2048, Decoding --- 128 steps
- Pre-filling stage --- computational bound, measure the speedup
- Decoding stage --- memory bound, measure the memory usage

| INT4, BS=1 | Time (ms) | Saving Factor | Memory (GB) | Saving Factor | WiKi↓ | QA avg.↑ |
|---|---|---|---|---|---|---|
| FP16 | 568 | - | 13.638 | - | 5.47 | 63.72 |
| SmoothQuant | 248 | 2.290x | 3.890 | 3.506x | 83.12 | 44.52 |
| QLLM | 435 | 1.306x | 3.894 | 3.502x | 9.09 | 51.60 |
| QuaRot | 284 | 2.000x | 3.891 | 3.505x | 6.39 | 61.25 |
| DuQuant | 288 | 1.972x | 3.893 | 3.503x | 6.28 | 61.76 |

Table E13: Decoding phase results of one LLaMA2-7B layer with a batch size of 64.

| Method | Time (ms) | Saving Factor | Memory (GB) | Saving Factor |
|---|---|---|---|---|
| FP16 | 659 | - | 3.550x | - |
| SmoothQuant | 437 | 1.508x | 1.669 | 2.127x |
| QLLM | OOM | - | OOM | - |
| QuaRot | 457 | 1.442x | 1.678 | 2.116x |
| DuQuant | 499 | 1.321x | 1.677 | 2.117x |



# Outline

# Summary

- **The motivation of our DuQuant is straightforward and insightful --- massive outliers at Down_proj**
- **Rotation and permutation demonstrates effective and fast for outlier management**
- **These two transformations are also highly motivated and easy to understand**

- **Discussion or interesting questions:**
  - The speedup for decoding stage requires better kernel.
  - How to fuse these transformations into LLMs?
  - Why LLaMA3 suffers the performance degradation?
  - What’s the influence of calibration data for LLM compression?

# Thanks for listening

# Q & A

haokun.lin@cripac.ia.ac.cn