From Causal Analysis to Outlier-Aware Diffusion Quantization

2026 동계 세미나 – 26.02.06

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

Presented By
최재민

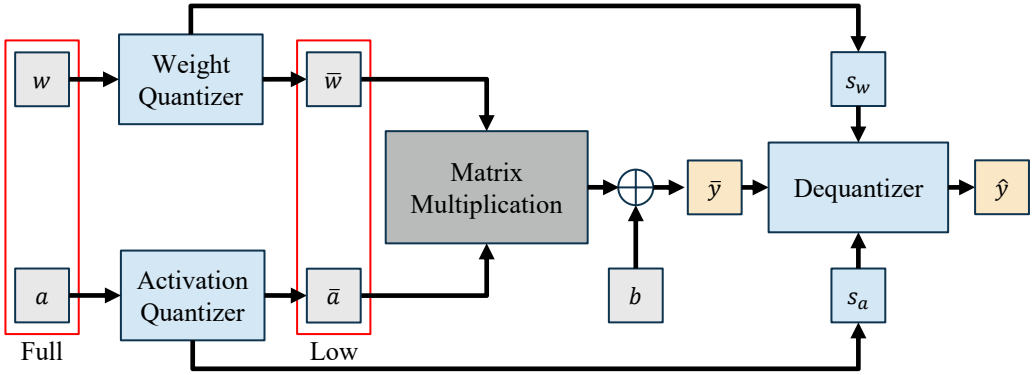
Outline

- Background
  - Quantization
  - PTQ vs QAT
  - Quantizer
  - Quantization difficulty

- Papers
  - Systematic Outliers in Large Language Models [ICLR 2025]
  - DGQ: Distribution-Aware Group Quantization for Text-to-Image Diffusion Models [ICLR 2025]

Background

- Quantization
  - 일반적으로 performance와 model size는 비례하는 경향이 있음
    - Model size ↑ → inference time ↑, computational cost ↑
    - 메모리 용량이 제한적인 edge device 환경에서 한계가 존재함
  - Full-precision → Low-precision
    - Weight, activation을 lower precision으로 낮춘 후 연산을 수행하는 가속화 기법

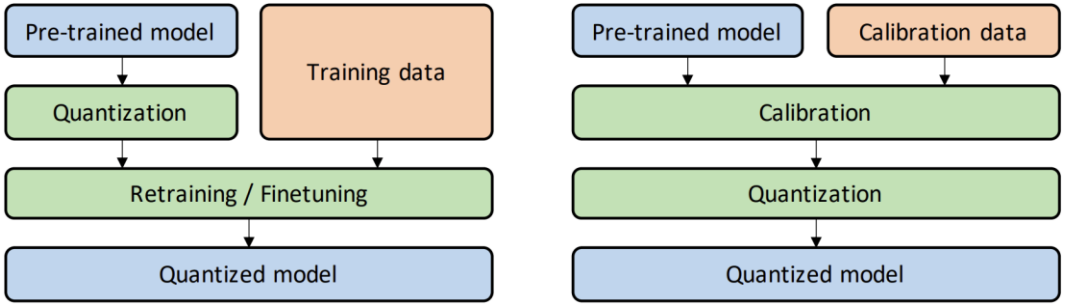


< Fig 1. Quantization의 가속화 원리 >

- Quantization process : $\bar{x} = Q(x) = clamp\left(\left\lfloor\frac{x}{s}\right\rceil + z, 0, 2^{bit}-1\right), s = \frac{\max(x)-\min(x)}{2^{bit}-1}, z = \left\lfloor -\frac{\min(x)}{s}\right\rceil$
- Dequantization process : $\hat{x} = s \cdot \bar{x}$
- Fully-Connected layer에서의 연산 : $f = \underline{WX} + B$ → FP32 Matmul
- Quantized FC layer에서의 연산 : $\hat{f} = \hat{W}\hat{X} + B = (s_W\bar{W})(s_X\bar{X}) + B = s_W s_X(\underline{\bar{W}\bar{X}}) + B$ → INT8 Matmul

Background

- PTQ vs QAT



< Fig 2. Comparison between QAT and PTQ[1)] >

▪ Quantization-Aware Training (QAT)

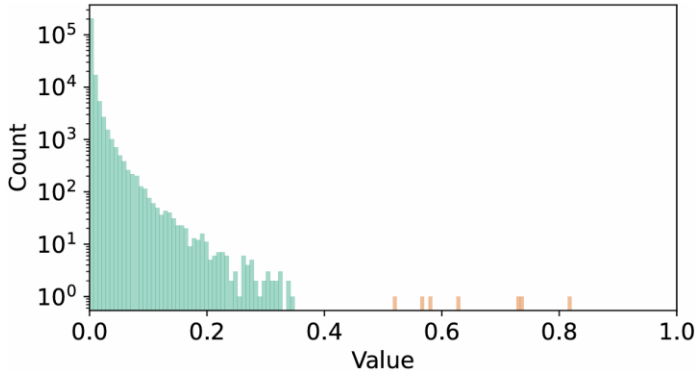
- Quantization 적용 후 pre-trained model의 train dataset으로 retraining/fine-tuning하는 방식
- Retraining/fine-tuning 과정에서 많은 시간이 필요하지만 PTQ에 비해 좋은 성능

▪ Post-Training Quantization (PTQ)

- 소량의 데이터(calibration dataset)만으로 pre-trained model에서의 quantization parameter 설정
- Calibration을 통하여 lower-bit에 mapping, 이후 inference 수행 → inference time ↓
- 소량의 데이터만을 사용하기 때문에 적은 시간만이 필요하지만 QAT에 비해 낮은 성능

Background

- Quantizer
  - Linear quantizer
    - Ex) $Q(x) = \bar{x} = clamp\left(\left\lfloor \frac{x}{s} \right\rceil + z, 0, 2^{bit} - 1\right), DQ(\bar{x}) = \hat{x} = \bar{x} \cdot (s - z)$
      - Uniform distribution과 같이 x의 분포가 고루 퍼져 있는 경우에 적합
      - CNN-based model의 activation에 주로 사용
  - Non-linear quantizer
    - Ex) $Q(x) = \bar{x} = clamp\left(\left\lfloor -\log_2 \frac{x}{s} \right\rceil, 0, 2^{bit} - 1\right), DQ(\bar{x}) = 2^{-\bar{x}} \cdot s$
      - Power-law distribution과 같이 x가 작은 값에 쏠려 있는 경우에 적합
      - Self-attention의 특성을 효율적으로 반영할 수 있어 transformer-based model에서 주로 사용

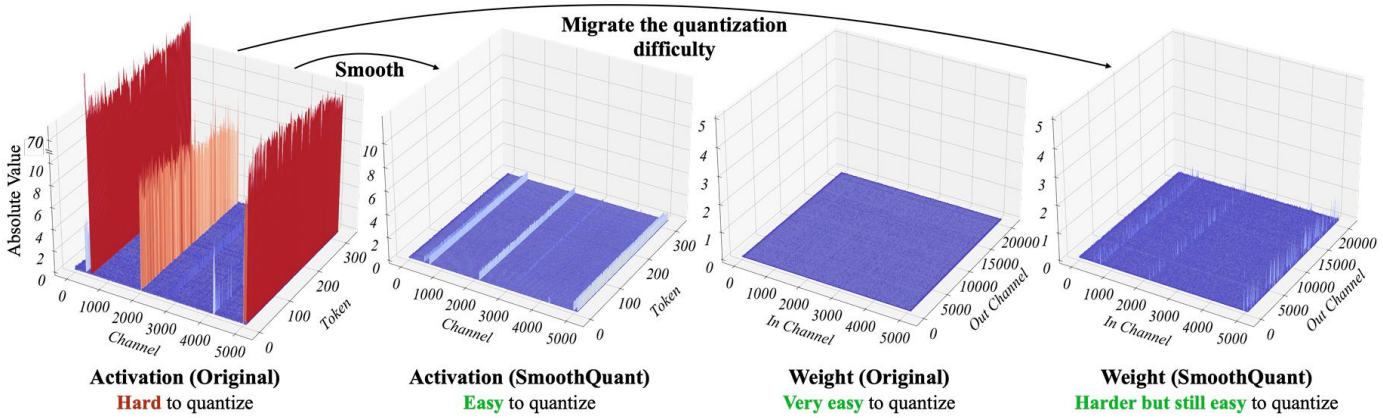


< Fig 3. Histogram of the post-Softmax activations[1] >

1) Xiao, Guangxuan, et al. "Smoothquant: Accurate and efficient post-training quantization for large language models." *International conference on machine learning*. PMLR, 2023.

Background

- Quantization difficulty



< Fig 3. Quantization difficulty[1)] – Weight/Activation distribution >

▪ Weight distribution

- 일반적으로 균일한 분포 (uniform, gaussian distribution)
- Min-max range가 작기 때문에 quantization error가 작음

▪ Activation distribution

- Transformer-based model의 경우 attention 연산에 의해 비균일한 분포 (power-law distribution)
- Min-max range가 크기 때문에 quantization error가 큼

Systematic Outliers in Large Language Models [ICLR 2025]

1) An, Yongqi, et al. "Systematic outliers in large language models." *arXiv preprint arXiv:2502.06415* (2025).

Systematic Outliers[1)]

- Problem statements
  - 최근 경량화 분야 연구에서 outlier의 존재가 성능을 크게 판가름 하고 있음
  - 연구자들은 outlier의 존재에 의해 경량화 기법의 성능이 크게 변한다는 것을 인지하고 있음
  - 그러나, 대부분의 연구들은 outlier를 고려하여 경량화 하는 것에 초점을 맞춤
  - 즉, outlier의 존재 자체에 대한 분석은 아직 부족한 상태

- Key contributions
  - 본 연구에서는 transformer model에서 관찰되는 outlier의 발생 원인과 기능을 탐구
  - 본 연구의 한 줄 요약
    - Weight, activation, attention outlier는 주로 weak semantic token에 집중되며, 이들은 self-attention의 softmax에 의해 발생한다.

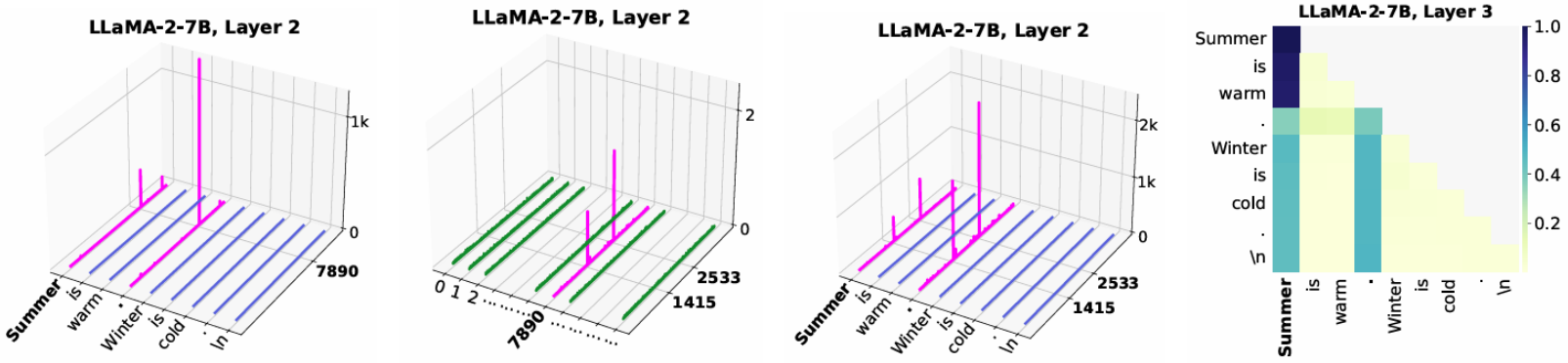
Systematic Outliers[1)]

- Objective
  - 본 연구의 목표 설정
    - Outlier의 발생 원인과 기능 탐구
    - 이를 기반으로 quantization-friendly model을 설계

- Definition of Outliers in LLMs (Systematic Outliers)
  - Activation Outliers
    - Layer outputs $\mathbf{h}_l \in \mathbb{R}^{B\times H}$ (batch size B, hidden dimension H)
    - $O_{activation} = \{(i,j) \mid |h_{i,j}| > \tau \cdot \mu_h\},\ \mu_h = \frac{1}{B\cdot H}\sum_{i,j}|h_{i,j}|$
  - Weight Outliers
    - Projection weights $\mathbf{W} \in \mathbb{R}^{O\times I}$ (output dimension O, input dimension I)
    - $O_{weight} = \{(i,j) \mid |w_{i,j}| > \tau \cdot \mu_{w_i}\},\ \mu_{w_i} = \frac{1}{I}\sum_{j}|w_{i,j}|$
  - Attention Outliers
    - Cumulative attention scores $\mathbf{A} \in \mathbb{R}^{L\times L}$ (sequence length L)
    - $O_{attention} = \{j \mid \hat{A}_j > \tau \cdot \mu_A\},\ \mu_A = \frac{1}{L}\sum_{j}\hat{A}_j$

Systematic Outliers[1)]

- Observation – Systematic outliers



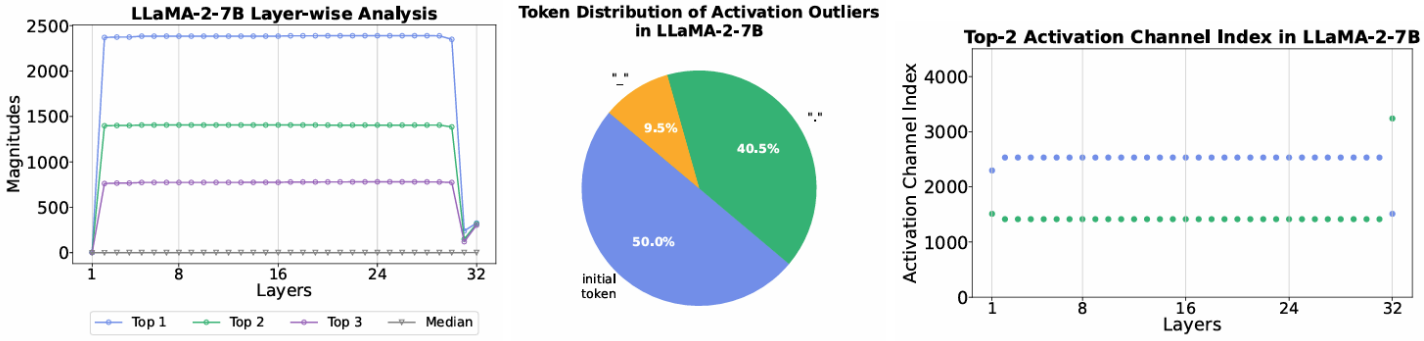
(a) activation: $\mathbf{x}_\ell^{\text{down}}$ **(b)** weight: $\mathbf{W}_\ell^{\text{down}}$ **(c)** activation: $\mathbf{h}_\ell$ **(d)** attention: $\mathbf{A}_\ell^i$

< Fig 1. Systematic outliers in LLaMA2-7B >

- Outlier의 발생 위치
  - Outlier가 발생하는 위치는 어떤 패턴을 가지고 있다.
    - MLP_down layer의 input activation
    - MLP_down_layer의 weight
    - Layer output
    - Attention score

Systematic Outliers[1)]

- Observation – Activation outliers



(a) Which layers? **(b)** Which sequences? **(c)** Which features?

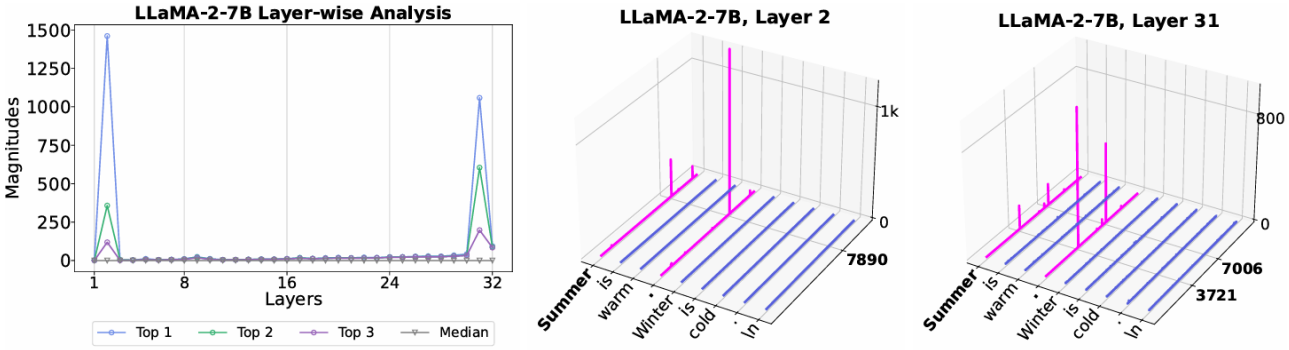
< Fig 2. Distribution of activation outliers (Layer output) >

▪ Outlier의 발생 위치

- (a) 1, 31, 32번 레이어를 제외한 대부분의 layer에서 outlier가 발생
- (b) 해당 outlier는 대부분 initial token, “-”, “.”에서 발생
- (c) Outlier가 발생했다면, 해당 outlier의 channel index는 대부분 동일

Systematic Outliers[1)]

- Observation – Activation outliers



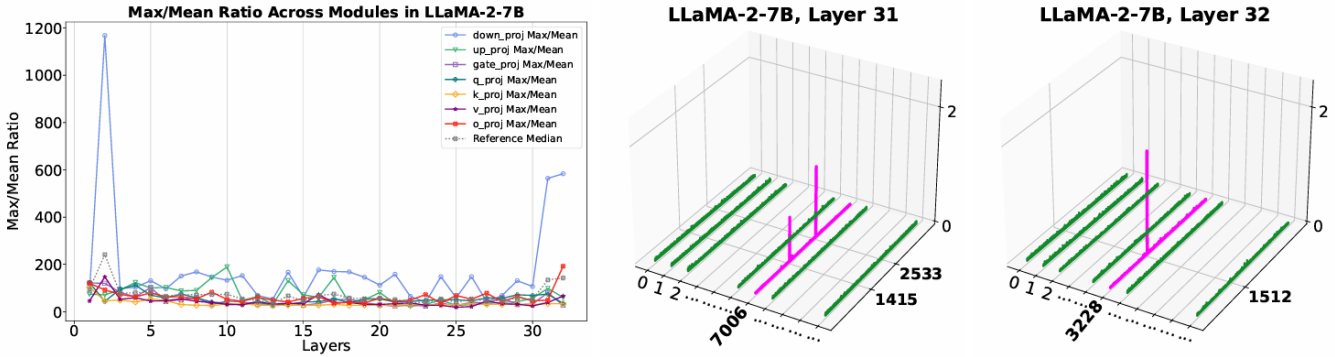
(a) Which layers? **(b)** Layer 2 **(c)** Layer 31

< Fig 3. Distribution of activation outliers (MLP_down_input) >

- Outlier의 발생 위치
  - (a) 2, 31번 레이어에서 outlier가 크게 발생
  - (b) Token 축에서는 "Summer", "."에, channel 축에서는 index 7890에서 outlier 발생
  - (c) Token 축에서는 "Summer", "."에, channel 축에서는 index 3721, 7006에서 outlier 발생

Systematic Outliers[1)]

- Observation – Weight outliers



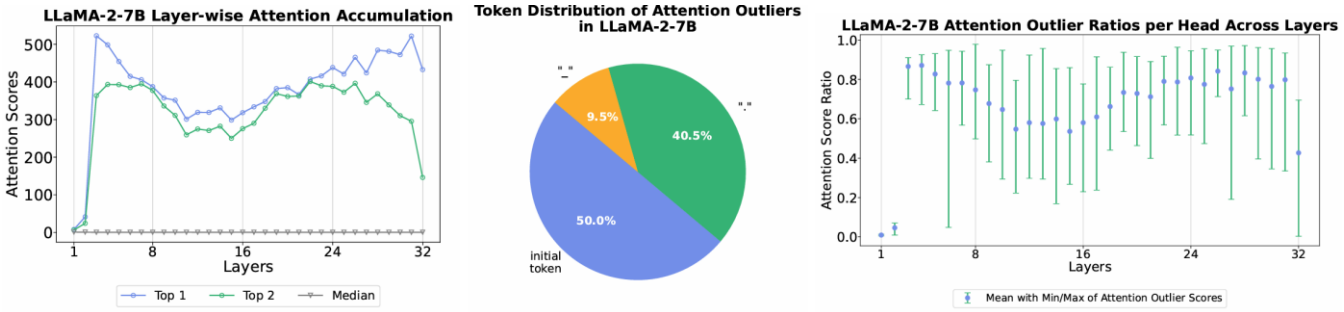
(a) Which layers? **(b)** Layer 31 **(c)** Layer 32

< Fig 4. Distribution of weight outliers >

- Outlier의 발생 위치
  - (a) MLP_down_weight에서 outlier가 크게 발생하며, 2, 31, 32번 레이어에 집중
  - (b) Input channel index 7006에서 outlier가 크게 발생, 이는 Fig 6.에서와 일치
  - (c) Input channel index 3228에서 outlier가 크게 발생

Systematic Outliers[1)]

- Observation – Attention outliers



(a) Which layers? **(b)** Which keys? **(c)** Which heads?

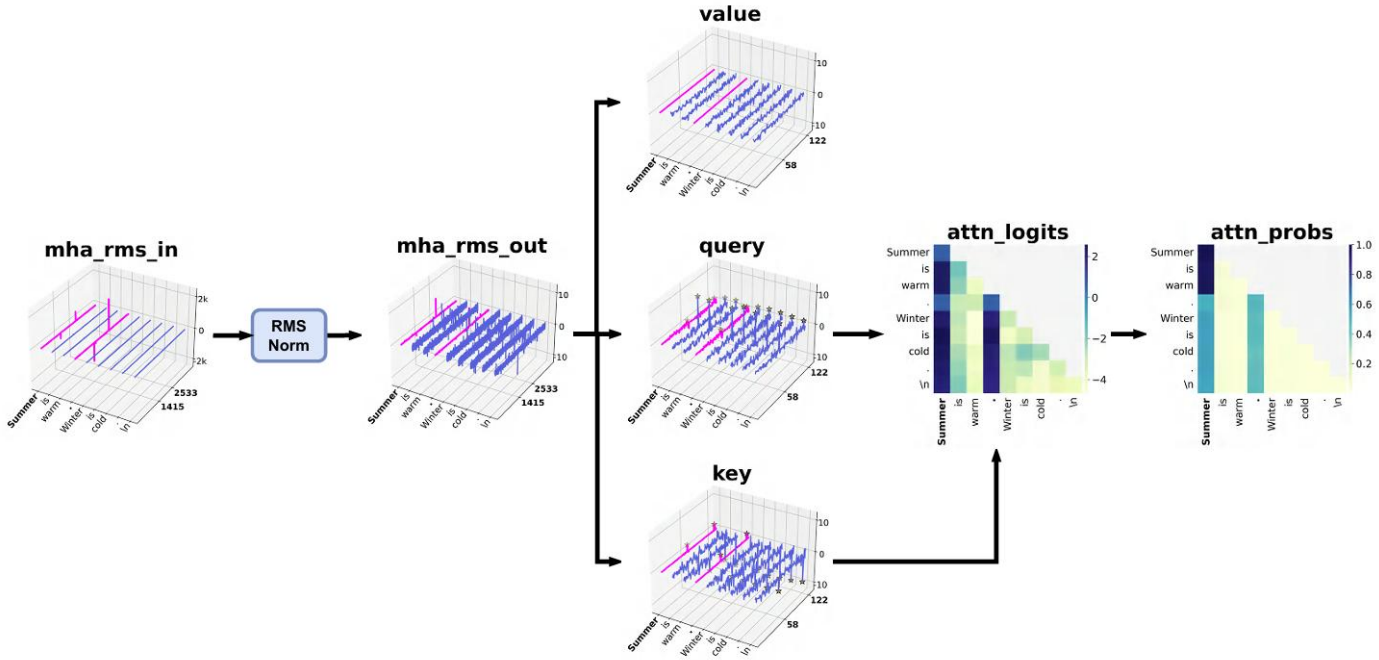
< Fig 5. Distribution of attention outliers >

▪ Outlier의 발생 위치

- (a) 초반 레이어를 제외한 대부분의 레이어에서 outlier가 발생
- (b) 해당 outlier는 대부분 initial token, "-", "."에서 발생
- (c) 대부분의 layer에서 outlier가 발생하며, head마다 outlier score에 일부 편차가 존재

Systematic Outliers[1)]

- Observation – Query, key, value vectors



< Fig 6. The spread of attention outliers from activation outliers >

- Query, key, value
  - Mha_rms_in의 outlier token은 query, key vector로 계산되었을 때 높은 similarity를 보임
    - 이는 attention outlier를 야기함
  - 그러나, value vector를 보면 상당히 낮은 값을 보이고 있음

Systematic Outliers[1)]

- Key observations
  - 1. Activation outlier, weight outlier, attention outlier에는 서로 연관이 있다.
    - Activation outlier와 weight outlier는 feature dimension에서 완벽히 일치
    - Activation outlier와 attention outlier의 sequence dimension에서 95%만큼 일치
  - 2. Activation outlier, attention outlier는 문장 내 의미 정보가 적은 토큰에 집중된다.
    - Initial token, “-”, “.”, …
  - 3. Outlier token에서 query, key는 높은 similarity를 보이나, value는 상당히 낮은 값을 갖는다.

Outlier Type 1	Outlier Type 2	Dimension	Consistency
weight outliers in $\mathbf{W}_\ell^{\text{down}}$	activation outliers in $\mathbf{x}_\ell^{\text{down}}$	feature	100%
weight outliers in $\mathbf{W}_\ell^{\text{down}}$	activation outliers in $\mathbf{h}_\ell$	feature	100%
activation outliers in $\mathbf{x}_\ell^{\text{down}}$	activation outliers in $\mathbf{h}_\ell$	sequence	100%
activation outliers in $\mathbf{h}_\ell$	attention outliers in $\mathbf{A}_\ell^i$	sequence	95%

< Fig 7. Consistency of different types of outliers across dimensions >

Systematic Outliers[1)]

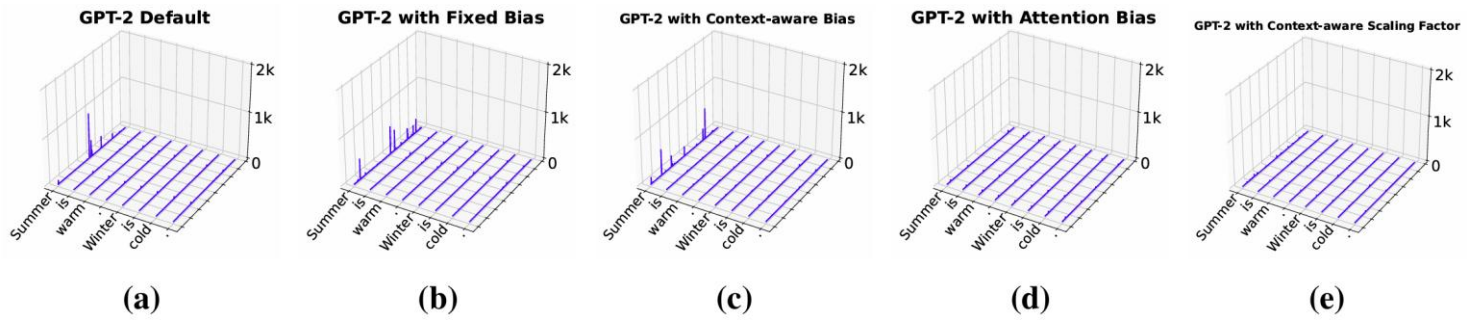
- Hypotheses
  - 1. Fixed but important biases
    - Context와 무관하게 모델의 동작에 지속적으로 영향을 미치는 고정된 biases
  - 2. Context-aware biases
    - Input sequence에 따라 자신의 영향력을 동적으로 조절하는 bias
  - 3. Context-aware scaling factors
    - Fig 9에 따르면 attention outlier와 value vector의 크기가 서로 반비례
    - 특정 토큰에 대한 영향력을 감소시켜 불필요한 업데이트를 줄이는 scaling factors
- Empirical validation of systematic outliers hypotheses
  - Formulation
    - 5개의 attention variants에 대하여 재학습 수행 후 분포 관찰

ID	Attention Variant	Formulation
(a)	Default Attention (Vaswani, 2017)	$\text{Attn}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V$
(b)	Explicit Fixed Bias	$\text{Attn}(Q,K,V;\mathbf{v}') = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V + \mathbf{v}'$
(c)	Explicit Context-Aware Bias	$\text{Attn}(Q,K,V;\mathbf{k}',\mathbf{v}') = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V + \text{softmax}\left(\frac{Q[K^T\ \mathbf{k}']}{\sqrt{d}}\right)\begin{bmatrix}0^T\\ \mathbf{v}'^T\end{bmatrix}$
(d)	Attention Bias	$\text{Attn}(Q,K,V;\mathbf{k}',\mathbf{v}') = \text{softmax}\left(\frac{Q[K^T\ \mathbf{k}']}{\sqrt{d}}\right)\begin{bmatrix}V\\ \mathbf{v}'^T\end{bmatrix}$
(e)	Explicit Context-Aware Scaling Factor	$\text{Attn}(Q,K,V) = S_c(x)\cdot\text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V$

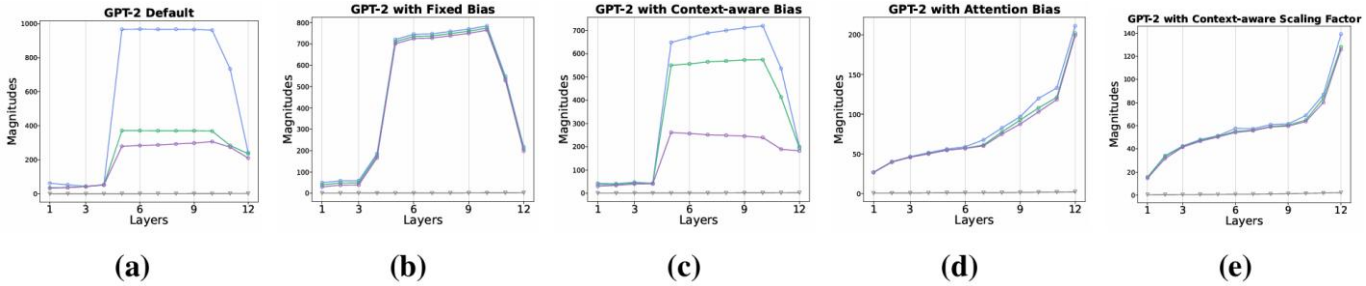
< Fig 8. Attention variant & Formulation >

Systematic Outliers[1)]

- Empirical validation of systematic outliers hypotheses
  - Experiments



< Fig 9. Activation outliers across different attention formulations >



< Fig 10. Top-3 largest activation outliers for each layers >

- (a), (b), (c) : 여전히 outlier 존재
- (d), (e) : outlier가 대부분 사라진 것을 확인할 수 있음

Systematic Outliers[1)]

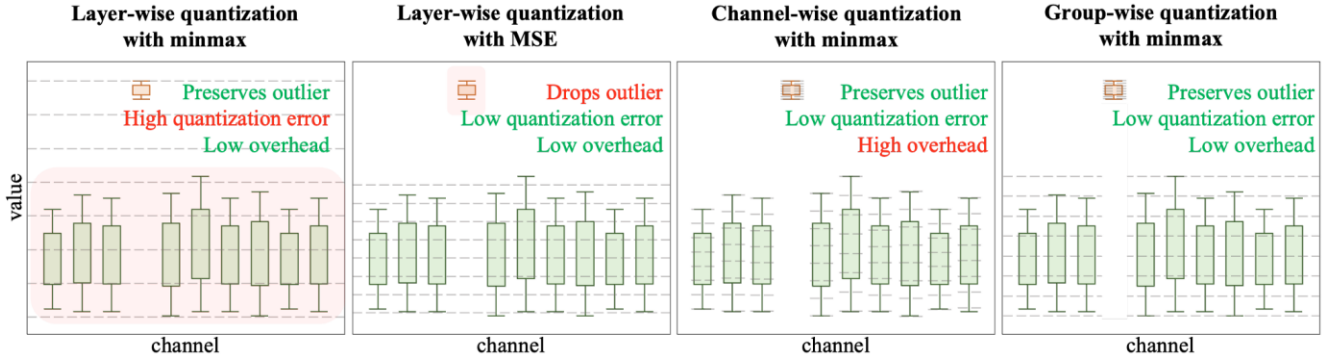
- Further Analysis of Systematic Outliers
  - Outlier의 발생 원인
    - **Outlier는 학습 과정에서 softmax의 제약조건에 의해 발생한다.**
    - 1. Attention에 포함되어 있는 softmax는 합이 1이 되도록 강제한다.
    - 2. 충분히 학습이 되어 weight update를 줄여야 하는 시점에도 합은 1로 유지되어야 한다.
    - 3. 학습이 충분히 된 토큰(정보량이 높은 토큰)은 weight update를 줄이기 위해 attention score를 0에 가깝게 낮춰야만 한다.
    - 4. 학습이 충분히 된 토큰은 0에, 정보량이 낮은 토큰은 1에 가깝게 attention score를 할당하게 된다.
    - 5. Attention score를 1에 가깝게 높이기 위해서는 필연적으로 query, key의 값이 크고 similarity가 높아야 한다.
      - $(Q \cdot K = |Q||K| \cos(\theta))$
    - 6. 이로 인해 정보량이 낮은 토큰에 대한 W_Q와 W_K가 높은 값을 갖도록 학습된다. → weight outlier
    - 7. Weight outlier가 발생하고, 이로 인해 activation outlier가 연달아 발생한다.

DGQ: Distribution-Aware Group Quantization for Text-to-Image Diffusion Models [ICLR 2025]

1) Ryu, Hyogon, NaHyeon Park, and Hyunjung Shim. "Dgq: Distribution-aware group quantization for text-to-image diffusion models." *arXiv preprint arXiv:2501.04304* (2025).

DGQ[1)]

- Problem statements
  - 기존 diffusion model에 대하여, low-bit quantization 시 극심한 오차 발생
  - Quantization granularity 선택에 있어, 기존 방식들은 outlier에 의한 오차가 큼

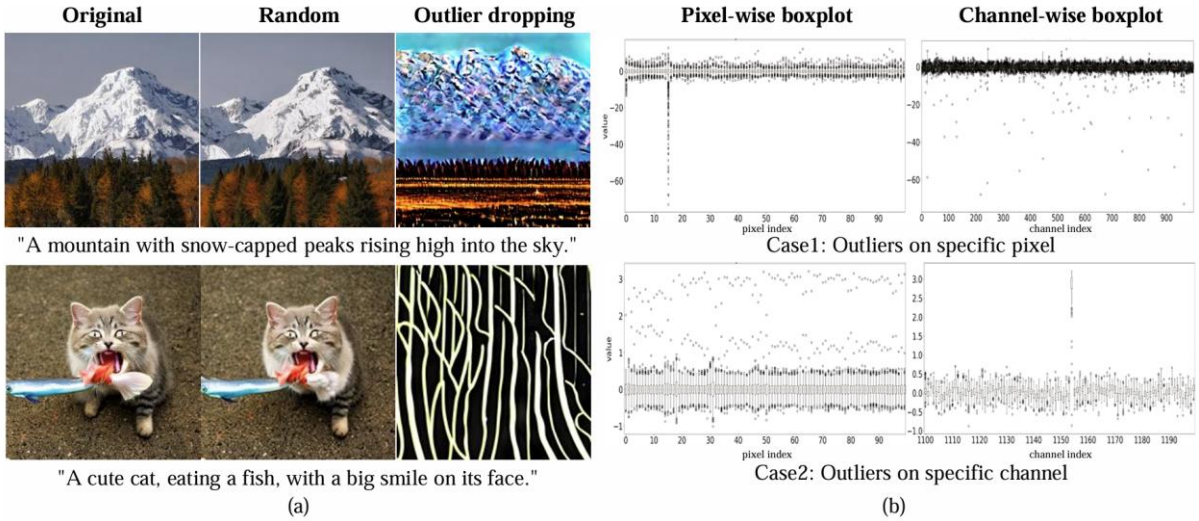


< Fig 1. Quantization granularity >

- Key contributions
  - 본 연구에서는 diffusion model에서 activation outlier의 역할을 발견
  - Activation outlier를 보존하는 quantization 방법론 제안
  - 본 연구의 한 줄 요약
    - Diffusion model에서 activation outlier는 weak semantic token에 집중되며, 이는 생성되는 이미지의 배경 정보를 담당한다.

DGQ[1)]

- Observations



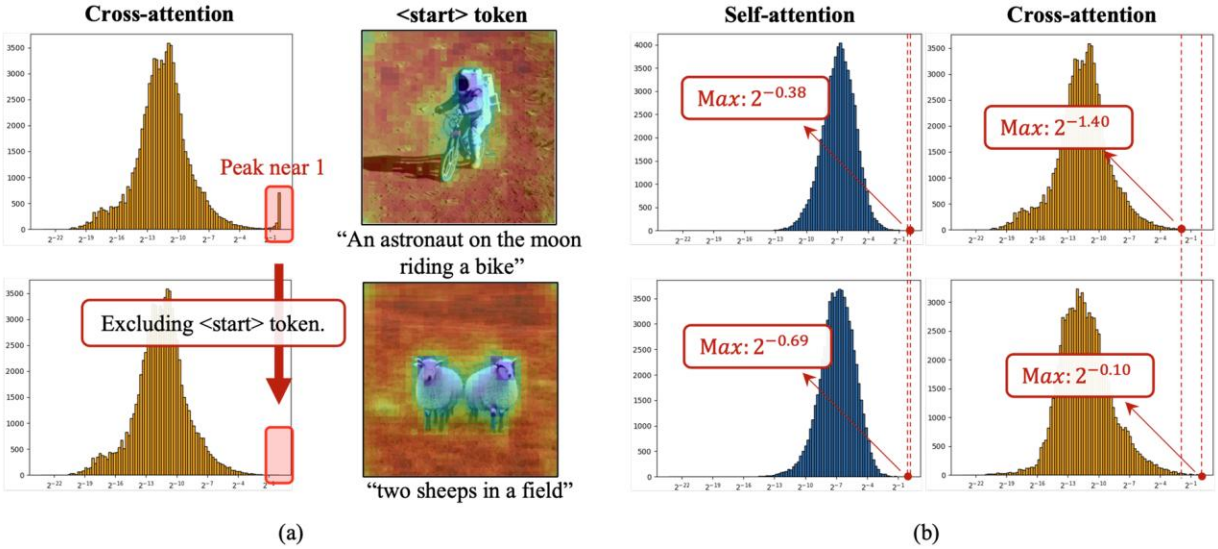
< Fig 2. Characteristics of activation outliers >

- Outlier dropping 실험
  - Outlier는 특정 pixel 또는 특정 channel에서 발생함
  - Random value를 drop한 경우 생성된 이미지가 일부 변할 뿐, 전체적인 품질은 유지됨
  - Activation outlier를 drop한 경우 생성된 이미지의 품질이 급락함

Methods	PSNR↑	LPIPS↓
random drop	17.81	0.295
outlier drop	9.34	0.773

DGQ[1)]

- Observations



< Fig 3. Characteristics of cross-attention scores >

- Cross-attention & <start> token
  - <start> token에 1에 가까운 attention이 할당
  - <start> token은 배경 정보에 높은 attention score를 갖는 경향이 있음
- Cross-attention vs Self-attention
  - Cross-attention은 input에 따른 min-max range 변화가 큼
    - ∴ 즉, cross-attention은 input prompt에 대한 종속성이 큼
  - Self-attention은 cross-attention에 비해 input에 따른 min-max range 변화가 작음

Statistic	Value
Std of cross-attention	0.826
Std of self-attention	0.334
Mean ratio of each layer's attention std	3.210

DGQ[1)]

- Methods
  - Observations
    - 1. Activation outlier는 이미지 품질 유지에 필수적이다.
    - 2. Activation outlier는 특정 channel 또는 pixel dimension에서 발생한다.
  - Outlier-preserving group quantization
    - Quantization을 적용하기 전에, granularity 선택을 위한 metric D_d를 정의

$$D_d = \left(\max_i a_{i,d}^{max} - \min_i a_{i,d}^{max}\right) + \left(\max_i a_{i,d}^{min} - \min_i a_{i,d}^{min}\right)$$

    - $d \in \{channel, pixel\}$
    - $a_{i,d}^{max}$: d dimension에서 i번째 벡터의 최대값
    - $a_{i,d}^{min}$: d dimension에서 i번째 벡터의 최소값

$$d^* = argmax_d D_d$$

    - D_d 를 최대로 만드는 dimension을 결정 후 K-means clustering을 통해 K group quantization 수행

DGQ[1)]

- Methods
  - Observations
    - 1. <start> token에 1에 가까운 attention score가 할당되며, 이들은 배경 정보에 집중한다.
    - 2. Cross-attention은 input prompt에 대한 종속성이 크다.
  - Attention-aware quantization

$$\mathbf{A}^{q}_{[:,1:]} = \text{clamp}\left(\left\lfloor -\log_2\left(\frac{\mathbf{A}_{[:,1:]}}{s}\right)\right\rfloor, 0, 2^b - 1\right), \qquad where\ s = \max\left(\mathbf{A}_{[:,1:]}\right)$$

$$\widehat{\mathbf{A}} = \left[\mathbf{A}_{[:,0]}, s \cdot 2^{-\mathbf{A}^{q}_{[:,1:]}}\right], \widehat{\mathbf{A}}\widehat{\mathbf{V}} = \left[\mathbf{A}_{[:,0]}\widehat{\mathbf{V}}_{[0,:]}, s \cdot 2^{-\mathbf{A}^{q}_{[:,1:]}}\widehat{\mathbf{V}}_{[1:,:]}\right]$$

- 위 수식에 따라 <start> token에는 quantization을 수행하지 않음으로써 정보 보존
- 위 수식을 dynamic quantization으로 적용함으로써 input prompt에 따라 동적으로 대응

DGQ[1)]

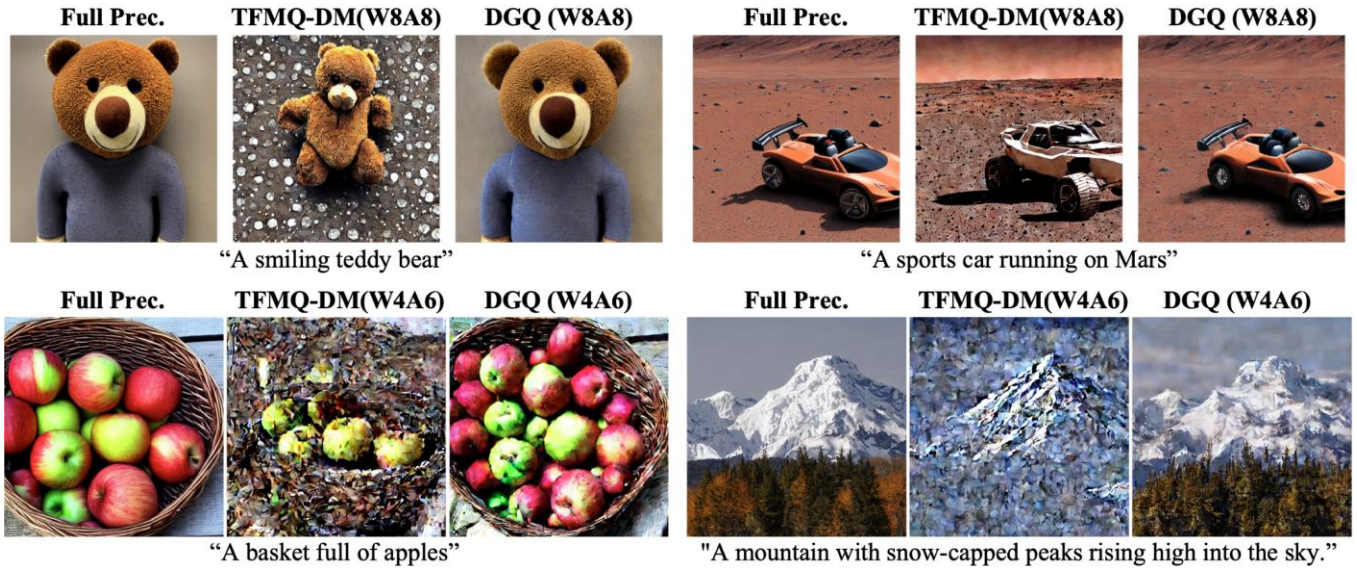
- Experiments
  - Quantitative Comparison

Model	Method	Bits(W/A)	Model Size	BOPs	*MS-COCO*			*PartiPrompts*
					IS↑	FID↓	CLIP↑	CLIP↑
SD v1.4	**Full Precision**	32/32	3,438MB	823T	36.52	14.44	0.298	0.293
	Q-Diff	8/8	871MB	51.4T	27.65	26.12	0.273	0.275
	TFMQ	8/8	871MB	51.4T	32.79	18.85	0.286	0.286
	DGQ (#groups=8)	8/8	871MB	51.4T	**35.38**	13.26	**0.297**	**0.292**
	DGQ (#groups=16)	8/8	871MB	51.4T	35.22	**13.15**	**0.297**	**0.292**
	Q-Diff	8/6	871MB	38.6T	4.12	221.76	0.080	0.120
	TFMQ	8/6	871MB	38.6T	6.57	175.16	0.146	0.178
	DGQ (#groups=8)	8/6	871MB	38.6T	22.65	37.76	0.268	0.277
	DGQ (#groups=16)	8/6	871MB	38.6T	**24.77**	**31.36**	**0.273**	**0.279**
	Q-Diff	4/8	436MB	25.7T	26.52	28.06	0.269	0.271
	TFMQ	4/8	436MB	25.7T	30.85	19.98	0.281	0.281
	DGQ (#groups=8)	4/8	436MB	25.7T	**33.91**	**13.28**	**0.294**	**0.289**
	DGQ (#groups=16)	4/8	436MB	25.7T	33.56	13.74	**0.294**	0.288
	Q-Diff	4/6	436MB	19.3T	3.37	242.75	0.072	0.108
	TFMQ	4/6	436MB	19.3T	5.24	229.64	0.127	0.155
	DGQ (#groups=8)	4/6	436MB	19.3T	20.14	51.94	0.257	0.272
	DGQ (#groups=16)	4/6	436MB	19.3T	**22.17**	**43.66**	**0.263**	**0.274**
SDXL Turbo (4 steps)	**Full Precision**	32/32	10,269MB	6,927T	35.97	21.25	0.308	0.309
	TFMQ	8/8	2,567MB	433T	12.24	111.69	0.067	0.069
	DGQ (#groups=8)	8/8	2,567MB	433T	**34.79**	**22.46**	**0.299**	**0.294**
	TFMQ	8/6	2,567MB	325T	4.27	163.02	-0.002	0.025
	DGQ (#groups=8)	8/6	2,567MB	325T	**28.56**	**34.31**	**0.251**	**0.223**
	TFMQ	4/8	1,284MB	216T	13.00	109.56	0.068	0.069
	DGQ (#groups=8)	4/8	1,284MB	216T	**28.33**	**29.22**	**0.289**	**0.291**
	TFMQ	4/6	1,284MB	162T	1.99	270.45	0.022	0.049
	DGQ (#groups=8)	4/6	1,284MB	162T	**22.93**	**45.00**	**0.245**	**0.226**

< Tab 1. Quantitative Comparison >

DGQ[1)]

- Experiments
  - Qualitative Comparison

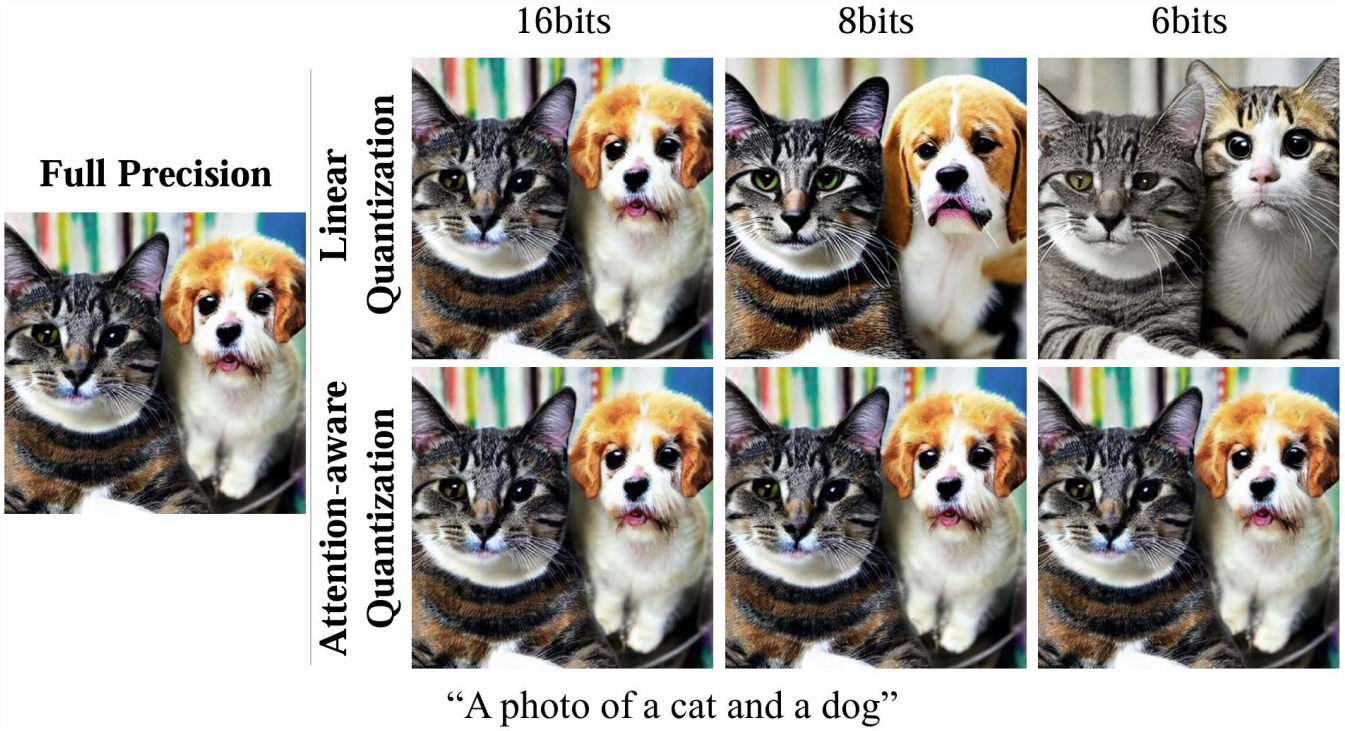


< Fig 4. Qualitative Comparison >

1) Ryu, Hyogon, NaHyeon Park, and Hyunjung Shim. "Dgq: Distribution-aware group quantization for text-to-image diffusion models." *arXiv preprint arXiv:2501.04304* (2025).

DGQ[1)]

- Experiments
  - Qualitative Comparison



< Fig 5. Qualitative Comparison – Linear quantization vs Attention-aware quantization>

1) An, Yongqi, et al. "Systematic outliers in large language models." *arXiv preprint arXiv:2502.06415* (2025).
2) Ryu, Hyogon, NaHyeon Park, and Hyunjung Shim. "Dgq: Distribution-aware group quantization for text-to-image diffusion models." *arXiv preprint arXiv:2501.04304* (2025).

Conclusion

- Systematic outliers[1)]
  - Key contributions
    - 1. LLMs에서 systematic outliers가 발생하는 패턴 분석
    - 2. Systematic outliers가 모델 내에서 어떤 역할을 하는지 실험을 통해 규명
    - 3. Systematic outliers가 왜 발생하게 되는지에 대한 고찰
    - 4. Quantization 적용 시, systematic outliers를 줄이는 것이 성능 보존에 유리함을 확인

Model	PPL (FP16)	PPL (AbsMax W8)	PPL (50% Sparse)
GPT-2 Default	27.24	93.44	7235.68
GPT-2 + Context-aware Scaling	26.95	29.22	39.47

- DGQ[2)]
  - Key contributions
    - 1. Diffusion model에서 activation outlier가 발생하는 패턴 분석
    - 2. Activation outlier가 어떤 역할을 하는지 실험을 통해 확인
    - 3. Cross-attention의 peak가 <start> token에서 발생하는 것을 확인
    - 4. Cross-attention과 self-attention의 distribution 분석
    - 5. Outlier와 attention distribution를 보존하기 위한 quantization 방법론 제안

감사합니다.