2024 겨울 세미나

Model analysis for quantization

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

Presented By
양진철

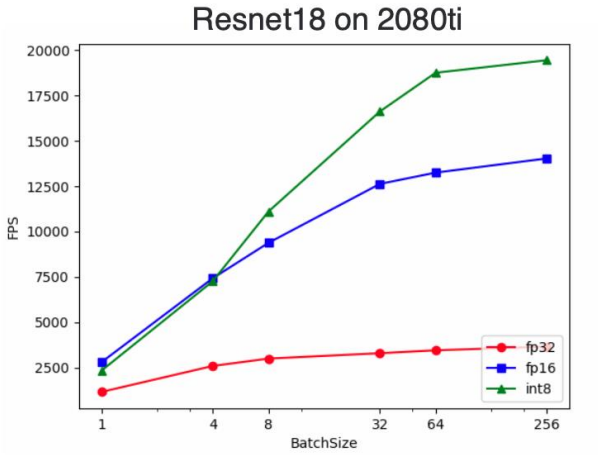
Outline

- Intro
  - What is quantization?
- Papers
  - RepQ-ViT: Scale Reparameterization for Post-Training Quantization of Vision Transformers (ICCV 2023)
  - Q-Diffusion: Quantizing Diffusion Models (ICCV 2023)
- Conclusion

Intro

- What is quantization?
  - 모델 최적화를 위한 motivation
    - Model size reduction
      - 컴퓨터 비전에서 모델들은 모델 사이즈를 크게 가지면서 성능을 향상
        - →모델 학습의 시간, latency 및 비용 증가
    - Performance benefits
      - Edge device의 부족한 메모리 용량
    - Applications such as real-time intelligent
      - health care monitoring, autonomous driving, …
  - Method for optimizing models
    - Quantization
    - Pruning
    - Knowledge Distillation
    - Efficient Network Design

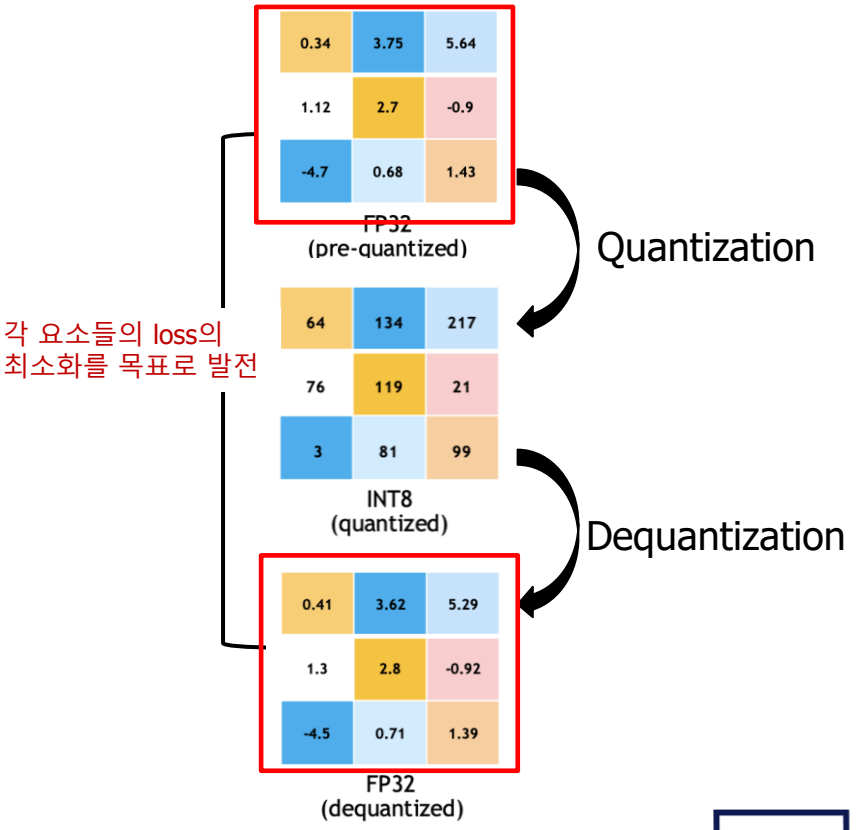
Comparison on latency



Intro

- What is quantization?
  - Quantization은 파라미터의 값(weight, activation map)의 표현 정밀도를 낮추는 과정
    - Floating point(FP32) value → INT value
  - Basic concepts
    - Quantization : $Q(r) = \left\lfloor \frac{r}{S} \right\rceil - Z; S = \frac{\beta - \alpha}{2^b - 1}$
    - Dequantization : $\tilde{r} = S(Q(r) + Z)$

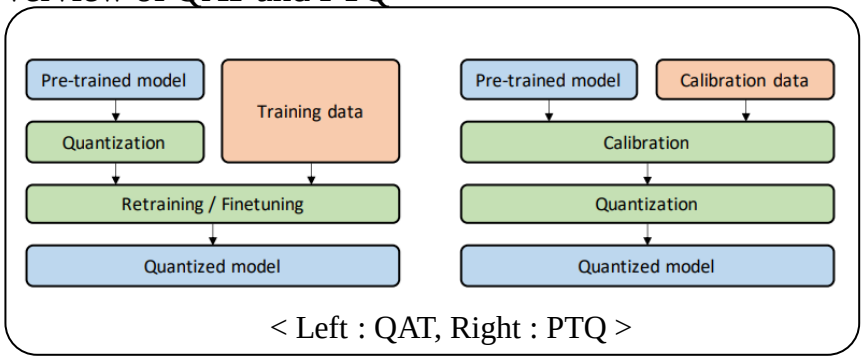
    Notations
      - $Q(r)$ = quantized representation of r
      - r = real value (FP)
      - S = scale factor
      - Z = zero-point
      - α, β =bounded range(clipping range)
      - b = bit width
      - $\lfloor \cdot \rceil$ = rounding function



Intro

- What is quantization?
  - Fine-tuning methods : PTQ vs QAT
    - Post-training quantization(PTQ)
      - Fine-tuning 없이 pre-trained model에서 모든 weight, activation quantization 파라미터를 quantization하는 방식
      - PTQ는 딥러닝 모델을 quantization하는 빠른 방법
      - QAT와 비교하여 낮은 accuracy
    - Quantization-aware training
      - Fine-tuning을 하면서 loss를 최소로 하는 최적의 파라미터 찾는 방식
      - Loss를 최소로 하는 최적의 파라미터 찾기 위해 fine-tuning에 많은 시간과 비용을 들이는 단점 존재
      - PTQ와 비교하여 높은 accuracy 달성

Overview of QAT and PTQ



< Left : QAT, Right : PTQ >

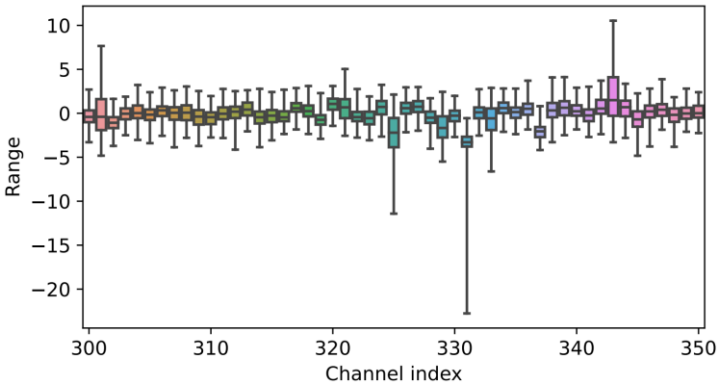
RepQ-ViT

- Keyword
  - Post training quantization (PTQ)
  - Scale reparameterization
    - Hardware-friendly framework
  - low bit (4bit)
  - Classification / object detection / instance segmentation

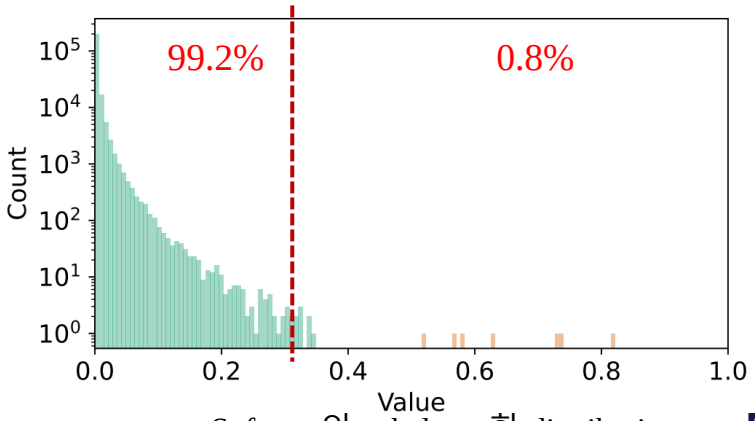
- Abstract
  - ViT-based model 분석
  - LayerNorm 과 Softmax 이후에 각각 channel-wise, $log\sqrt{2}$ 양자화 프로세스 적용
  - 양자화, 추론 프로세스를 분리한 새로운 프레임워크 제안

RepQ-ViT

- Method
  - Model analysis
    - LayerNorm 이후의 채널간 범위
      - 심각한 inter-channel 변동은 양자화 성능에 영향을 끼침
        - ✓ Channel-wise 양자화 방법 사용
          - 채널 별 양자화는 전용 하드웨어의 지원이 필요하며 추가적인 계산 오버헤드가 발생
    - Softmax 이후의 unbalance 한 범위
      - 대부분이 0에 집중되어 있고, 소수의 값만 1에 가깝게 분산
        - ✓ $log\sqrt{2}$ 양자화 방법 사용
          - $log\sqrt{2}$ 양자화는 하드웨어 친화적이지 못함



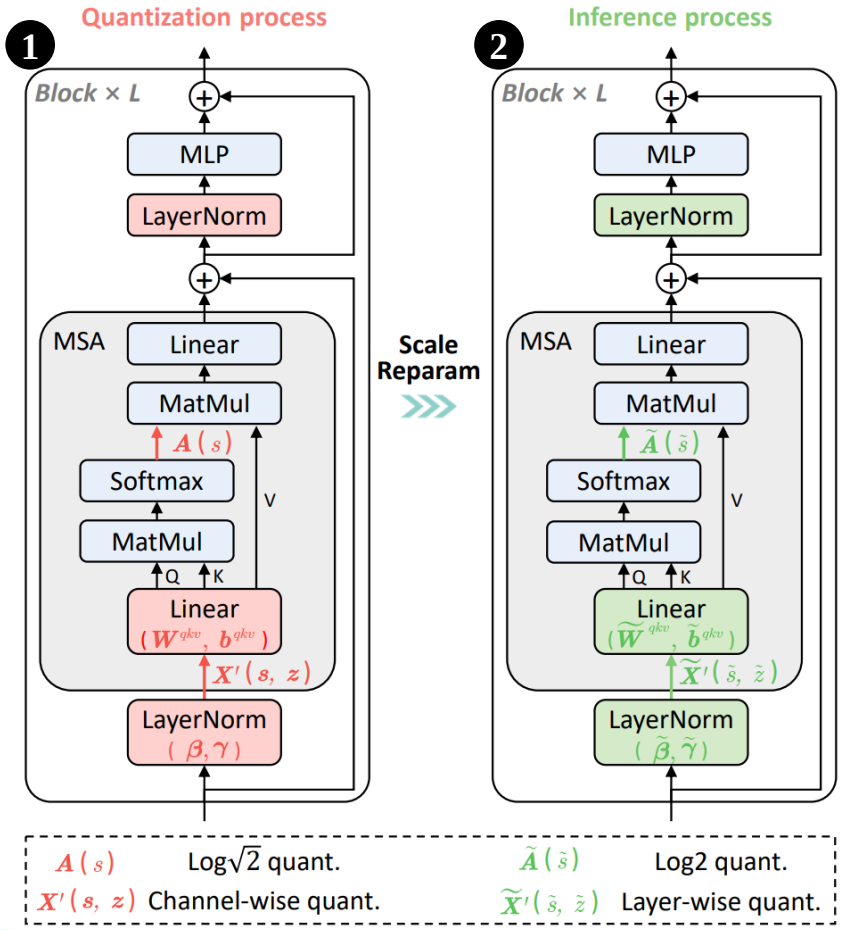
< post-LayerNorm 의 inter-channel variation>



< post- Softmax의 unbalance한 distribution>

RepQ-ViT

- Method
  - Overview of the proposed framework



① Quantization process
- Calibration* 단계에서 사용되는 방법
  - LayerNorm 과 Softmax 활성화 함수에 대해 각각 Channel-wise, $log\sqrt{2}$ 양자화 방법 사용
- Hardware non-friendly 방법

② Inference process
- Inference 단계에서 사용되는 방법
  - LayerNorm 과 Softmax 활성화 함수에 대해 각각 Layer-wise, $log2$ 양자화 방법 사용
- Hardware-friendly 방법

*PTQ 방식에서 사용되는 방법으로 학습 데이터셋의 일부 값을 통해 미리 양자화 파라미터(clipping range, scale) 조정

RepQ-ViT

- Method

Algorithm 1 Pipeline of RepQ-ViT framework.

1: **Input:** Pretrained full-precision model, Calib data
2: Initialize the quantized model with calib data and Eq. 9, where post-LayerNorm activations $\boldsymbol{X}'$ apply channel-wise quantization $(\boldsymbol{s}, \boldsymbol{z})$ and post-Softmax activations $\boldsymbol{A}$ apply $\log\sqrt{2}$ quantization (s);
   # Scale reparam for post-LayerNorm activations
3: Update the quantizer of $\boldsymbol{X}'$ via $\tilde{s} = \mathrm{E}[\boldsymbol{s}]$ and $\tilde{z} = \mathrm{E}[\boldsymbol{z}]$;
4: Calculate $\boldsymbol{r}_1 = \boldsymbol{s}/(\tilde{s} \cdot \mathbf{1})$ and $\boldsymbol{r}_2 = \boldsymbol{z} - (\tilde{z} \cdot \mathbf{1})$;
5: Update LayerNorm's affine factors $\widetilde{\boldsymbol{\beta}}$ and $\widetilde{\boldsymbol{\gamma}}$ based on Eq. 15;
6: Update next layer's weights $\widetilde{\boldsymbol{W}}^{qkv}$ and $\widetilde{\boldsymbol{b}}^{qkv}$ based on Eq. 17;
7: Re-calibrate $\widetilde{\boldsymbol{W}}^{qkv}$ with calib data;
   # Scale reparam for post-Softmax activations
8: Update quantization procedure based on Eq. 18;
9: Update $\tilde{s}$ in de-quantization procedure based on Eq. 20;
10: **Output:** Quantized model

Quantization 공식 remind

- Quant : $x_{int} = clip(\left\lfloor \frac{x}{s} \right\rfloor + z, 0, 2^b - 1)$
- DeQuant : $x_{fp} = s(x_{int} - z)$
- Scale : s = $\frac{\max(x) - \min(x)}{2^b - 1}$, Zero-point : z = $\left\lfloor -\frac{\min(x)}{s} \right\rceil$

Quantization process

- Line1-2 : Calibration dataset으로 초기화 (양자화 파라미터 조정)
  - Post-LayerNorm activation 에서는 채널 별로 양자화
  - Post-Softmax activation 에서는 $log\sqrt{2}$ 양자화
    - Quant : $A_{int} = clip(\left\lfloor -\log_{\sqrt{2}} \frac{A}{s} \right\rceil, 0, 2^b - 1)$

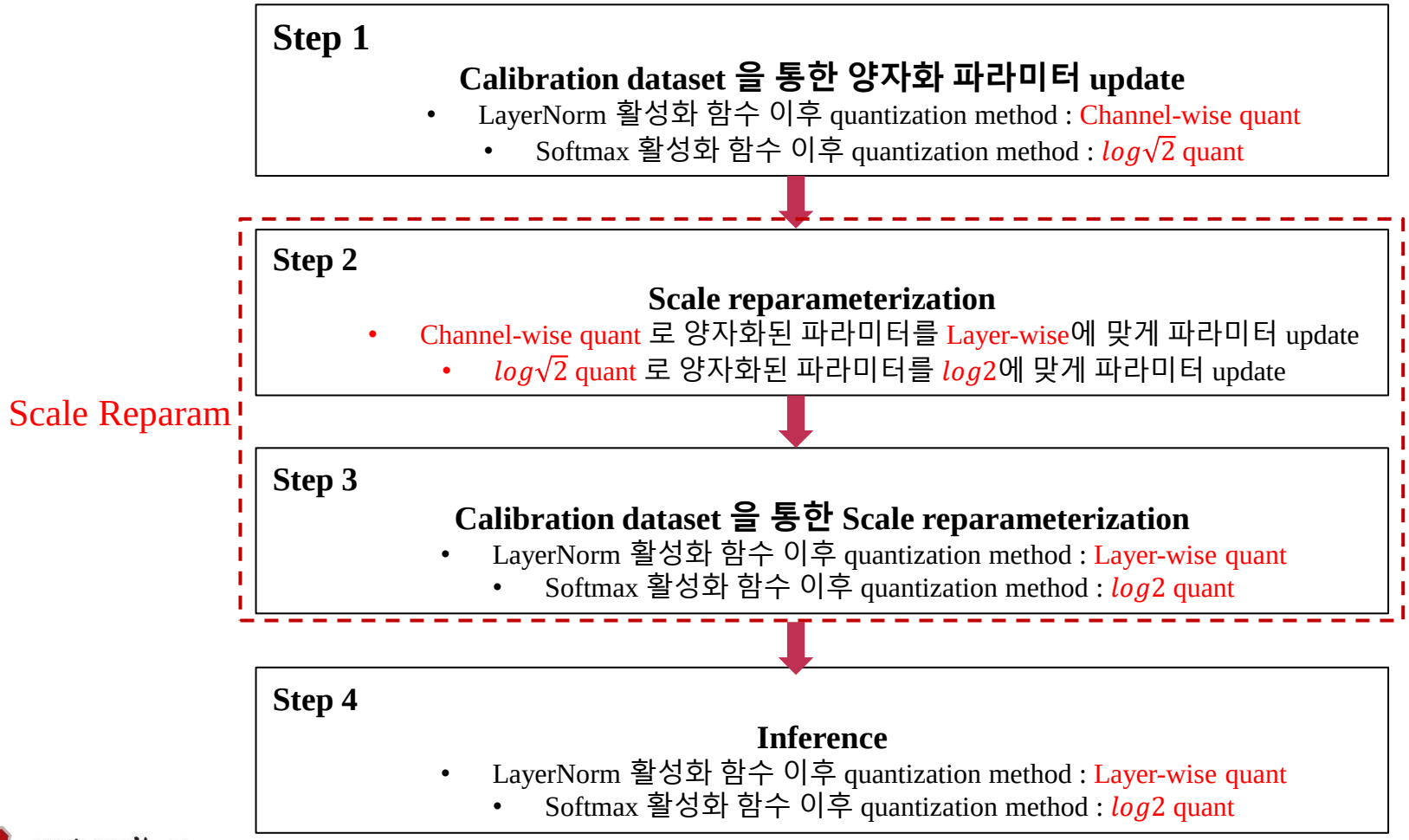
Scale reparam process

$$LayerNorm(X_{n,:}) = \frac{X_{n,:} - E[X_{n,:}]}{\sqrt{Var[X_{n,:}] + \varepsilon}} \odot \gamma + \beta$$

- Line3-7 : Post-LayerNorm activation reparameterize (scale, zero_point)
  - 채널 별 양자화 파라미터 (scale, zero-point)를 r_1, r_2를 통해 reparameterize 하여 LayerNorm의 새로운 $\tilde{\gamma}, \tilde{\beta}$ 로 update
  - Weight, bias update 후 recalibrate
- Line8-10 : Post-Softmax activation reparameterize
  - $log\sqrt{2}$ 양자화에서 $log2$ 양자화를 위한 scale 파라미터 update

RepQ-ViT

- Method
  - 전체 flow



RepQ-ViT

- Ablation Studies
  - 제안한 방법들의 효과 입증
    - W4/A4 조건 실험

post-LayerNorm 양자화

Model	Method	Hardware	Top-1 (%)
DeiT-S	Full-Precision	-	79.85
	Layer-Wise Quant.	✓	33.17
	Channel-Wise Quant.	×	**70.28**
	Scale Reparam (ours)	✓	69.03
Swin-S	Full-Precision	-	83.23
	Layer-Wise Quant.	✓	57.63
	Channel-Wise Quant.	×	**80.52**
	Scale Reparam (ours)	✓	79.45

post-Softmax 양자화

Model	Method	Hardware	Top-1 (%)
DeiT-S	Full-Precision	-	79.85
	Log2 Quant.	✓	67.71
	Log$\sqrt{2}$ Quant.	×	**69.03**
	Scale Reparam (ours)	✓	**69.03**
Swin-S	Full-Precision	-	83.23
	Log2 Quant.	✓	77.87
	Log$\sqrt{2}$ Quant.	×	**79.45**
	Scale Reparam (ours)	✓	**79.45**

Calibration data 수 비교

Model	Method	Top-1 (%)	Calib Data	GPU Min.
DeiT-S	Full-Precision	79.85	-	-
	FQ-ViT [22]	0.10	1000	**0.5**
	PTQ4ViT [35]	34.08	**32**	3.2
	RepQ-ViT (ours)	**69.03**	**32**	1.3
Swin-S	Full-Precision	83.23	-	-
	FQ-ViT [22]	0.10	1000	**1.1**
	PTQ4ViT [35]	76.09	**32**	7.7
	RepQ-ViT (ours)	**79.45**	**32**	2.9

RepQ-ViT

- Experimental results
  - ImageNet 데이터셋에 대한 classification task 의 top-1 accuracy

Method	No HP	No REC	Prec. (W/A)	ViT-S	ViT-B	DeiT-T	DeiT-S	DeiT-B	Swin-S	Swin-B
Full-Precision	-	-	32/32	81.39	84.54	72.21	79.85	81.80	83.23	85.27
FQ-ViT [22]	×	✓	4/4	0.10	0.10	0.10	0.10	0.10	0.10	0.10
PTQ4ViT [35]	×	×	4/4	42.57	30.69	36.96	34.08	64.39	76.09	74.02
APQ-ViT [4]	×	×	4/4	47.95	41.41	47.94	43.55	67.48	77.15	76.48
RepQ-ViT (ours)	✓	✓	4/4	**65.05**	**68.48**	**57.43**	**69.03**	**75.61**	**79.45**	**78.32**
FQ-ViT [22]	×	✓	6/6	4.26	0.10	58.66	45.51	64.63	66.50	52.09
PSAQ-ViT [19]	×	✓	6/6	37.19	41.52	57.58	63.61	67.95	72.86	76.44
Ranking [24]	×	×	6/6	-	75.26	-	74.58	77.02	-	-
PTQ4ViT [35]	×	×	6/6	78.63	81.65	69.68	76.28	80.25	82.38	84.01
APQ-ViT [4]	×	×	6/6	79.10	82.21	70.49	77.76	80.42	82.67	84.18
RepQ-ViT (ours)	✓	✓	6/6	**80.43**	**83.62**	**70.76**	**78.90**	**81.27**	**82.79**	**84.57**

  - COCO 데이터셋에 대한 object detection 과 instance segmentation task 의 average precision

Method	No HP	No REC	Prec. (W/A)	Mask R-CNN				Cascade Mask R-CNN			
				w. Swin-T		w. Swin-S		w. Swin-T		w. Swin-S	
				AP^{box}	AP^{mask}	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}
Full-Precision	-	-	32/32	46.0	41.6	48.5	43.3	50.4	43.7	51.9	45.0
PTQ4ViT [35]	×	×	4/4	6.9	7.0	26.7	26.6	14.7	13.5	0.5	0.5
APQ-ViT [4]	×	×	4/4	23.7	22.6	**44.7**	40.1	27.2	24.4	47.7	41.1
RepQ-ViT (ours)	✓	✓	4/4	**36.1**	**36.0**	44.2	**40.2**	**47.0**	**41.4**	**49.3**	**43.1**
PTQ4ViT [35]	×	×	6/6	5.8	6.8	6.5	6.6	14.7	13.6	12.5	10.8
APQ-ViT [4]	×	×	6/6	**45.4**	41.2	**47.9**	42.9	48.6	42.5	50.5	43.9
RepQ-ViT (ours)	✓	✓	6/6	45.1	**41.2**	47.8	**43.0**	**50.0**	**43.5**	**51.4**	**44.6**

Conclusion

- Conclusion
  - Quantization-Inference decoupling 패러다임 적용
    - Quantization process에는 복잡한 quantizer 사용
    - Inference process 에는 hardware-friendly quantizer 를 사용하며, 이 둘은 scale reparameterization 통해 명시적으로 연결
  - Low-bit(4bit) 에서 기존 방법보다 훨씬 뛰어난 성능

- Limitations
  - Calibration dataset을 작은 수(32)로 calibration을 진행하지만, 최종적으로 2번을 한 후 quantization 이 진행되므로 시간 소요가 있다.
  - 다양한 모델에 대해 진행하였지만 ViT-T, Swin-T model에 대해서는 진행하지 않았다.
    - 특히, ViT-T model을 4bit 로 양자화 한 결과 top1 accuracy는 26.422 로 성능이 좋지 않았다.
  - Classification 이외의 task 에 대한 quantization 설명 부재

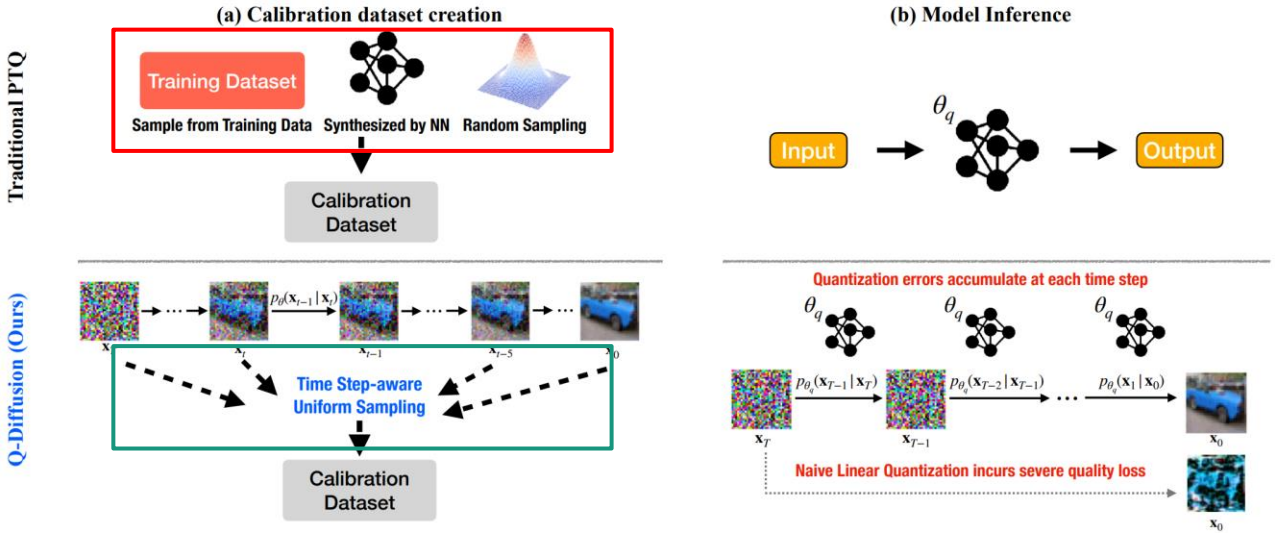
Q-Diffusion

- Keyword
  - Post training quantization (PTQ)
  - Time step-aware calibration data sampling
  - Weight low bit (4bit)
  - Diffusion

- Abstract
  - 양자화 적용에 대한 분석
    - time steps 마다 activation distribution diversity and the quantization error accumulation
  - noise estimation network 에 대한 PTQ solution
  - timestep-aware calibration and split shortcut quantization 방법 제안

Q-Diffusion

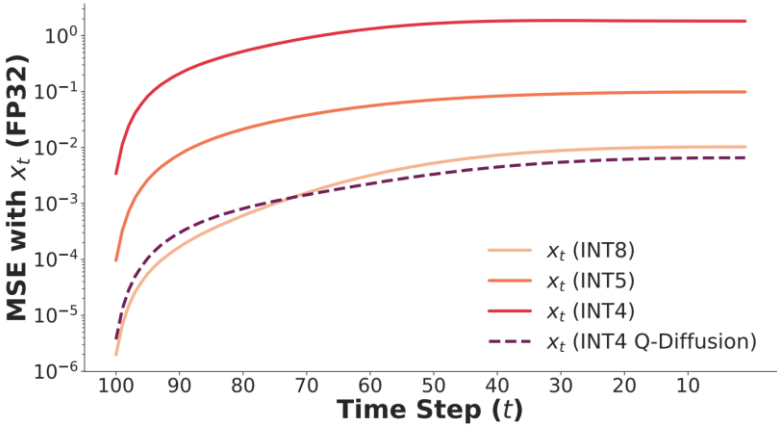
- Method
  - Analysis
    - Classification, detection task 와 달리 diffusion 모델은 UNet 아키텍처를 통해 multi-step 방식으로 학습 및 평가
      - Q-Diffusion은 데이터가 없는 방식으로 inference 중에 본 데이터를 정확하게 반영하는 입력으로 Calibration dataset 구축 → Time step 마다 노이즈 추정 네트워크 출력 분포 다름
      - 여러 번의 단계 inference 에서 누적된 양자화 → 반복 추론은 양자화 오류의 누적으로 이어짐



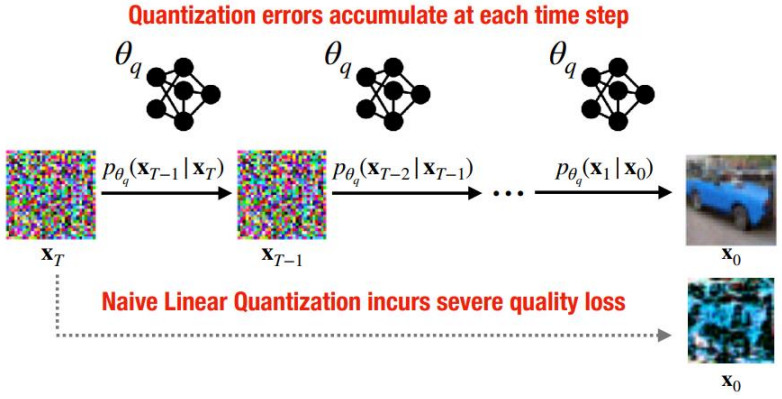
< Conventional PTQ scenarios and Q-Diffusion difference>

Q-Diffusion

- Method
  - Analysis
    - 1. Challenges under the Multi-step Denoising (Inference process)
      - Challenge 1: Quantization errors accumulate across time steps.
      - Challenge 2: Activation distributions vary across time steps.
    - Challenge 1: Quantization errors accumulate across time steps
      - Reverse process (generate an image)
        - ✓ 임의의 시간 단계 t에서 노이즈 제거 모델의 입력은 이전 시간 단계 t + 1에서 모델의 출력인 x_{t+1} 로 유도
          - $p_\theta(x_{t-1}|x_t) = N(x_{t-1}; \tilde{u}_{\theta,t}(x_t), \tilde{\beta}_t I)$



< Denoising process of DDIM on CIFAR10 >



Q-Diffusion

- Method
  - Analysis
    - Challenge 2 : Activation distributions vary across time steps

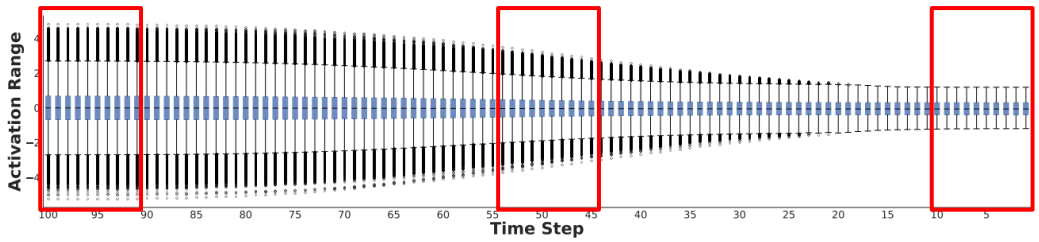
다양한 time step에 걸쳐 UNet 모델의 출력 활성화 분포를 분석

✓ 출력 활성화 분포가 time step에 따라 다르므로 양자화에 어려움

• 모든 time step 에서 일부의 time step 만을 사용하면 특정 분포에서 과적합이 발생하고 일반화되지 않기 때문

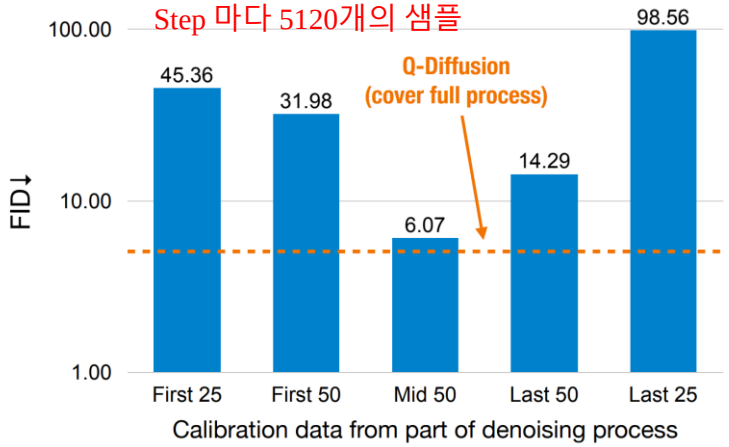
다양한 time step 의 출력 분포를 종합적으로 고려하는 방식으로 calibration dataset 선택

Step 마다 1000개의 random 샘플



< Activation ranges of x_t across all 100 time steps of FP32 DDIM model on CIFAR-10 >

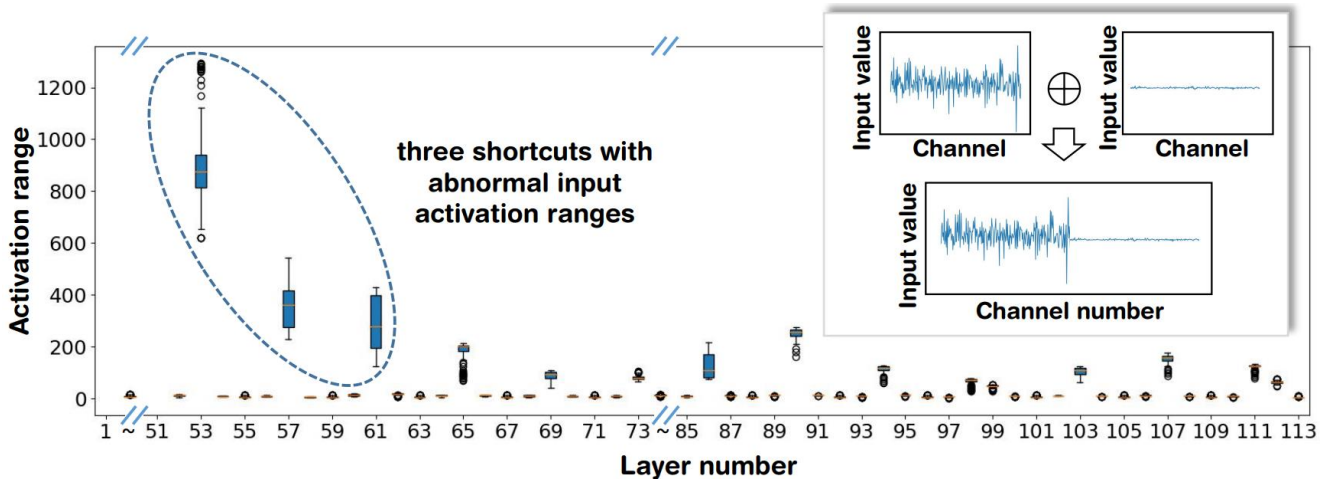
Step 마다 5120개의 샘플



< Effects of time steps in calibration dataset creation on 4-bit weights quantization results with DDIM on CIFAR10 >

Q-Diffusion

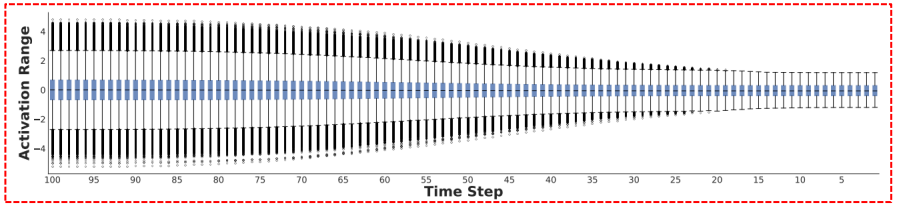
- Method
  - Analysis
    - 2. Challenges on Noise Estimation Model Quantization (UNet architecture)
      - CNN 의 UNet architecture 만을 고려하여 분석
        - ✓ Shortcut 레이어 input activation 값의 abnormal 한 범위 (인접한 레이어의 최대 x200)
      - 동일한 양자화 방법을 사용하여 전체 가중치 및 활성화를 양자화 시 큰 오류 발생



< Activation ranges of DDIM's FP32 outputs across layers averaging among all time steps >

Q-Diffusion

- Method
  - Q-Diffusion
    - Time step-aware calibration data sampling



Algorithm 1 Q-Diffusion Calibration

Require: Pretrained full precision diffusion model and the quantized diffusion model $[W_\theta, \hat{W}_\theta]$
Require: Empty calibration dataset $\mathcal{D}$
Require: Number of denoising sampling steps T
Require: Calibration sampling interval c, amount of calibration data per sampling step n

for $t = 1, \ldots, T$ time step **do**
  if t % c = 0 **then**
    Sample n intermediate inputs $\mathbf{x}_t^{(1)}, \ldots, \mathbf{x}_t^{(n)}$ randomly at t from W_θ and add them to $\mathcal{D}$
  end if
end for

for all $i = 1, \ldots, N$ blocks **do**
  Update the weight quantizers of the i-th block in $\hat{W}_\theta$ with $\mathcal{D}$ and W_θ
end for

if do activation quantization **then**
  for all $i = 1, \ldots, N$ blocks **do**
    Update the activation quantizers step sizes of the i-th block with $\hat{W}_\theta, W_\theta, \mathcal{D}$.
  end for
end if

Time step-aware calibration
- 실험적으로, 샘플링 된 calibration dataset은 INT4 양자화 모델의 성능을 대부분 복구
- 양자화 오류 수정을 위한 calibration dataset 방법으로 효과적인 샘플링 방식이라는 것을 확인

Sampling steps T 에서 모든 time step 에 걸쳐 고정된 간격(c)으로 중간 입력을 균일하게 무작위로 샘플링하여 calibration dataset D 구성

Calibration dataset D 로 모델의 block의 weight 양자화 파라미터(clipping range, scale) update
(quantized output과 FP32 output 간의 MSE 최소화)

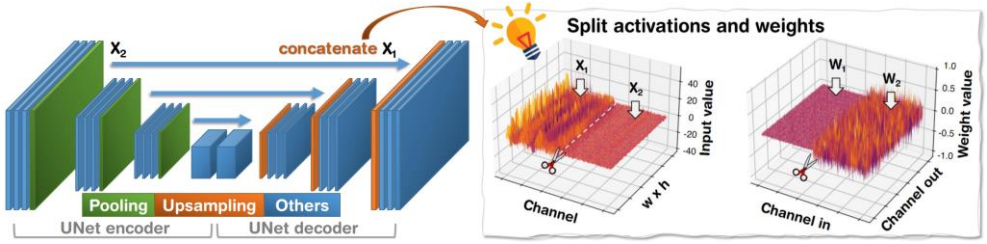
Calibration dataset D 로 모델의 block의 activation 양자화 파라미터(scale) update

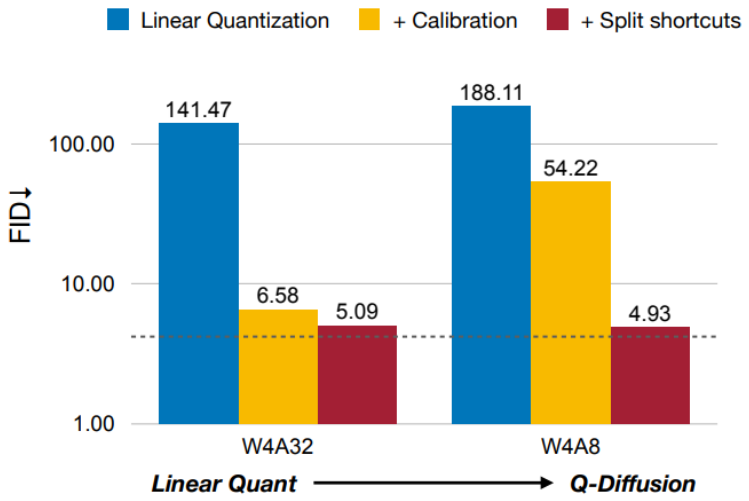
Q-Diffusion

- Method
  - Q-Diffusion
    - Shortcut-splitting quantization
      - Shorcut-layer의 비정상적인 활성화 및 가중치 분포를 해결하기 위한 방법
      - X_1과 X_2 concate 과정 전에 양자화를 수행하는 "split 양자화 기법" 제안
        - ✓ 추가적인 메모리나 계산 리소스 필요하지 않음

Split quantization method

$$Q_X(X) = Q_{X_1}(X_1) \oplus Q_{X_2}(X_2)$$
$$Q_W(W) = Q_{W_1}(W_1) \oplus Q_{W_2}(W_2)$$
$$Q_X(X)Q_W(W) = Q_{X_1}(X_1)Q_{W_1}(W_1) + Q_{X_2}(X_2)\,Q_{W_2}(W_2)$$





Q-Diffusion

- Experimental results
  - FID, IS metric 에 대한 정량적 결과
  - Quantization results for unconditional image generation

Method	Bits (W/A)	Size (Mb)	GBops	FID↓	IS↑
Full Precision	32/32	143.2	6597	4.22	9.12
Linear Quant	8/32	35.8	2294	4.71	8.93
SQuant	8/32	35.8	2294	4.61	8.99
Q-Diffusion	8/32	35.8	2294	**4.27**	**9.15**
Linear Quant	4/32	17.9	1147	141.47	4.20
SQuant	4/32	17.9	1147	160.40	2.91
Q-Diffusion	4/32	17.9	1147	**5.09**	**8.78**
Linear Quant	8/8	35.8	798	118.26	5.23
SQuant	8/8	35.8	798	464.69	1.17
Q-Diffusion	8/8	35.8	798	**3.75**	**9.48**
Linear Quant	4/8	17.9	399	188.11	2.45
SQuant	4/8	17.9	399	456.21	1.16
Q-Diffusion	4/8	17.9	399	**4.93**	**9.12**

< DDIM on CIFAR-10 (32 × 32) >

Method	Bits (W/A)	Size (Mb)	TBops	FID↓
Full Precision	32/32	1179.9	22.17	4.06
Linear Quant	8/32	295.0	10.73	**3.84**
SQuant	8/32	295.0	10.73	4.01
Q-Diffusion	8/32	295.0	10.73	4.03
Linear Quant	4/32	147.5	5.36	32.54
SQuant	4/32	147.5	5.36	33.77
Q-Diffusion	4/32	147.5	5.36	**4.45**
Linear Quant	8/8	295.0	2.68	14.62
SQuant	8/8	295.0	2.68	54.15
Q-Diffusion	8/8	295.0	2.68	**3.65**
Linear Quant	4/8	147.5	1.34	14.92
SQuant	4/8	147.5	1.34	24.50
Q-Diffusion	4/8	147.5	1.34	**4.12**

< LDM-4 on LSUN-Bedrooms (256 × 256) >

Method	Bits (W/A)	Size (Mb)	TBops	FID↓
Full Precision	32/32	1096.2	107.17	2.98
Linear Quant	8/32	274.1	37.28	3.02
SQuant	8/32	274.1	37.28	**2.94**
Q-Diffusion	8/32	274.1	37.28	2.97
Linear Quant	4/32	137.0	18.64	82.69
SQuant	4/32	137.0	18.64	149.97
Q-Diffusion	4/32	137.0	18.64	**4.86**
Linear Quant	8/8	274.1	12.97	6.69
SQuant	8/8	274.1	12.97	4.92
Q-Diffusion	8/8	274.1	12.97	**4.40**
Linear Quant	4/8	137.0	6.48	24.86
SQuant	4/8	137.0	6.48	95.92
Q-Diffusion	4/8	137.0	6.48	**5.32**

< LDM-8 on LSUN-Churches (256 × 256) >

Q-Diffusion

- Experimental results
  - 제안한 방법에 대한 정성적 결과
  - Quantization results for unconditional image generation

Bedroom Q-Diffusion (W4A8)

Bedroom Linear Quantization (W4A8)

Church Q-Diffusion (W4A8)

Church Linear Quantization (W4A8)

Q-Diffusion

- Experimental results
  - 제안한 방법에 대한 정성적 결과
  - Quantization results for text-guided image synthesis results

Prompt : "*A puppy wearing a hat.*"

Full Precision

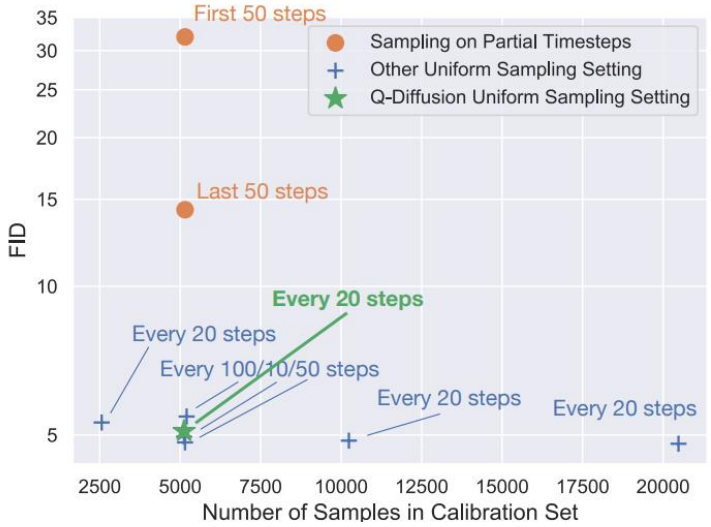
Q-Diffusion (W4A8)

Linear Quantization (W4A8)

Conclusion

- Conclusion
  - Q-Diffusion model 제안
    - Denoising process에서 multiple time steps 로 calibration 수행
  - Low-bit(4bit) 에서 Q-Diffusion model은 FP32 모델과 비슷한 결과를 달성

- Limitations
  - 제안한 uniform sampling setting 한계
    - Other Uniform Sampling Setting 과의 차이점 미미



Thank you