💜

[9주차] Post-training Quantization on Diffusion Models

0. Abstract

📌 **denoising diffusion 생성 모델**의 **생성 속도를 개선**하는 방법 제안

➡️ denoising diffusion 모델은 현재 생성 과정이 매우 느리다는 단점이 존재

➡️ 반복적인 노이즈 추정 과정에서 **무거운 신경망**을 사용하기 때문

➡️ 해당 논문에서는 모델 압축 기법 중 하나인 **Post-Training Quantization(PTQ)**을 활용하여 노이즈 추정 네트워크를 **경량화**하는 방법을 제안

- Post-Training Quantization(PTQ)을 통한 가속화
  - **PTQ**: 모델 **재학습 없이** 모델을 압축할 수 있어 denoising diffusion 모델에 적용하기에 적합함
  - 하지만 denoising diffusion 모델의 **노이즈 추정 네트워크**는 시간 단계에 따라 출력 분포가 변하기 때문에, **기존의 PTQ 방법은 적용하기 어려움**
- denoising diffusion 모델 전용 PTQ 방법 제안
  - 해당 논문에서는 denoising diffusion 모델의 고유한 **다중 시간 단계 구조**를 고려하여 PTQ 방법을 **개선**함
  - **양자화 연산**, **보정 데이터셋**, **보정 지표** 등 다양한 측면에서 탐구하여 **최적의 PTQ 방법을 도출**함
  - 실험 결과, 위 방법을 통해 높은 정밀도의 denoising diffusion 모델을 8비트 모델로 압축할 수 있었으며, 성능 저하 없이 생성 속도를 높일 수 있었음
  - 위 방법은 다른 빠른 샘플링 기법(DDIM 등)과 호환되어 **플러그인 형태**로 사용할 수 있음
- denoising diffusion 모델의 활용

- 이미지 생성 외에도 음성 합성, 텍스트 생성, 3D 모델 생성 등에 활용될 수 있음
- 모델의 실용화를 위해 **생성 속도 향상**이 매우 중요하며, 해당 논문에서 제안한 PTQ 기반 가속화 방법은 큰 의미가 있음

1. Introduction

- denoising diffusion 모델의 성과와 문제점
  - 최근 denoising diffusion 생성 모델은 이미지, 오디오, 비디오, 그래프 등 다양한 분야에서 성과를 거두고 있음
  - 이러한 모델은 실제 데이터를 점진적으로 **가우시안 노이즈**로 변환하고, 이 과정을 **역으로 수행**하여 **실제 데이터를 생성**함
  - 그러나 이와 같은 생성 과정은 **수천 번의 반복적인 노이즈 추정 과정**을 거치므로 매우 느림 ➡️ denoising diffusion 모델의 실용화를 어렵게 만드는 요인
- 기존 가속화 방법의 한계
  - 기존 연구에서는 **샘플링 경로 최적화** 등 다양한 방법으로 생성 속도를 높이고자 함
  - 그러나 이러한 방법들은 반복적인 노이즈 추정 과정의 **비용**을 고려하지 않았다는 한계가 존재함
- **모델 압축**을 통한 가속화 제안
  - 해당 논문에서는 모델 압축 기법 중 하나인 Post-Training Quantization(PTQ)을 활용하여 노이즈 추정 네트워크를 경량화하는 방법을 제안함
  - 기존 PTQ 방법은 **단일 시간 단계 시나리오**를 대상으로 하기 때문에, denoising diffusion 모델의 **다중 시간 단계 구조**에 적용하기 어려웠음

  ➡️ denoising diffusion 모델의 **고유한 특성을 고려**하여 PTQ 방법을 개선하고자 함
  - **양자화 연산**, **보정 데이터셋**, **보정 지표** 등 다양한 측면에서 탐구하여 **최적의 PTQ 방법**을 도출할 예정
  - 이를 통해 높은 정밀도의 denoising diffusion 모델을 경량화하여 생성 속도를 높이고자 함

📍 **양자화(Quantization)**

➡️ 양자화는
연속적인 값을 유한한 개수의 **이산적인 값**으로 변환하는 과정
➡️ 본 논문에서 양자화는 가중치와 활성화 함수 값을 낮은 비트 수로 표현하여 모델 크기와 연산 속도를 줄이는 데 사용됨

양자화 매개변수
1️⃣
s(스케일링 팩터): 입력 데이터 x를 정수로 변환하기 위해 나누는 값
2️⃣
z(오프셋 값): 반올림된 값에서 빼는 값으로 출력 값의 범위를 조정
* 양자화 매개변수들은 입력 데이터 x와 양자화된 출력 x_int 사이의 오차를 최소화하도록 선택됨

- 연구 목적
  - denoising diffusion 모델에 PTQ 기법을 적용하여 모델 가속화를 달성
  - PQT 기법의 핵심 요소(양자화 연산, 보정 데이터셋, 보정 지표)가 양자화된 denoising diffusion 모델의 최종 성능에 어떤 영향을 미치는지 분석하고자 함
- 문제 인식
  - 기존 PTQ 방법을 denoising diffusion 모델에 단순히 적용하면 큰 성능 저하가 발생
  - 이는 denoising diffusion 모델의 노이즈 추정 네트워크가 시간 단계에 따라 출력 분포가 변화하기 때문
  - 이러한 **시간 단계 의존성**은 기존 PTQ의 핵심 모듈인 **보정 과정**을 denoising diffusion 모델에 적용하기 어렵게 만듦
- 제안 방법: **PTQ4DM**
  - 위 문제를 해결하기 위해 "**Normally Distributed Time-step Calibration(NDTC)**"라는 **denoising diffusion 모델 전용 보정 방법**을 제안
  - NDTC는 시간 단계를 편향된 정규 분포에서 샘플링하여 보정 데이터를 생성함으로써 시간 단계 간 분포 차이를 강조함
  - 이를 통해 PTQ4DM이라는 denoising diffusion 모델 전용 가속화 방법을 개발함

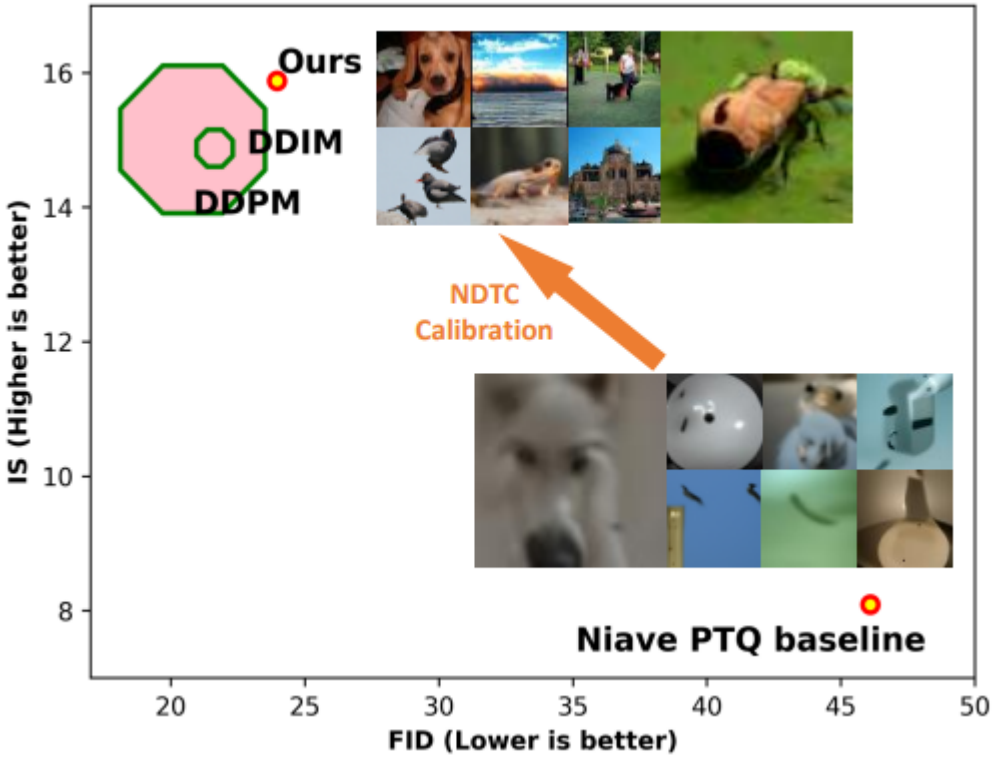
- 기여

  1️⃣ PTQ를 denoising diffusion 모델 가속화에 처음으로 도입

  2️⃣ denoising diffusion 모델의 **시간 단계 의존성** 문제를 분석하고, **NDTC**라는 해결책을 제안함

  3️⃣ PTQ4DM을 통해 사전 학습된 denoising diffusion 모델을 8비트로 양자화할 수 있었으며, 이는 denoising diffusion 모델 가속화 분야의 새로운 성과라고 볼 수 있음

  4️⃣ PTQ4DM은 다른 denoising diffusion 모델 가속화 기법과 호환되는 **플러그인 모듈**로 활용될 수 있음



2. Related Work

2.1 Diffusion Model Acceleration

- 확산 모델의 가속화 필요성
  - **확산(diffusion) 모델**은 이미지 생성을 위해 많은 denoise 단계를 거치므로, 노이즈 추정 속도(노이즈 제거 과정의 속도)가 느리다는 문제가 있다
  - 확산 모델의 **성능을 유지하면서도 노이즈 추정 속도를 높이는** 가속화 방법이 필요함

        - 노이즈 추정 속도가 빠르다 = 비교적 적은 단계로 노이즈를 제거할 수 있다
        - 노이즈 추정 속도가 느리다 = 노이즈 제거를 위해 많은 단계가 필요하다
        - 노이즈 추정 속도와 정확도 사이에는 **트레이드오프 관계**가 존재
- 기존 확산 모델의 가속화 방법

  1️⃣ **짧은 샘플링 경로 찾기**

    - 그리드 서치를 통해 6단계의 효과적인 경로를 찾았으나, 경로의 길이가 길어질수록 시간 복잡도가 기하급수적으로 증가함
    - 동적 프로그래밍을 활용하여 경로 탐색 문제를 해결함
    - 비마르코프 확산 과정을 활용하여 역과정 샘플링 속도를 높임

  2️⃣ **연속 시간 확산 모델 가속화**

    - 확산 모델을 상미분 방정식 형태로 표현하고, 빠른 ODE 솔버를 활용함
    - 적응형 SDE 솔버를 사용하여 역과정 샘플링을 가속화
    - 사전 학습된 스코어 기반 모델을 활용하여 분산과 KL 발산을 효율적으로 추정
- 본 연구의 차별점
    - 기존 연구들은 주로 **샘플링 경로 최적화**에 초점을 맞췄으나, 본 연구는 각 노이즈 추정 반복에 대한 **네트워크 압축**을 통해 추가적인 가속화를 달성하고자 함
    - 본 연구에서 제안하는 PTQ4DM 방법은 기존 가속화 기법들과 동일한 접근법이며, 이들 기법과 호환되는 **플로그인 모듈**로 활용될 수 있음
    - 이는 post training 모델을 양자화하는 최초의 연구라는 점에서 의의가 있음

2.2 Post-training Quantization

- 신경망 압축을 위한 **양자화(Quantization)** 기법

  1️⃣ **양자화 인지 학습**(Quantization-aware Training, **QAT**): **학습 과정**에서 양자화를 고려하는 방식

  2️⃣ **포스트 트레이닝 양자화**(Post-training Quantization, **PTQ**): **학습 완료 후** 양자화를 수행하는 방식
- 포스트 트레이닝 양자화(PTQ)
    - 신경망 모델을 재학습하지 않고도 가중치와 활성화 함수의 양자화 매개변수를 결정하는 빠른 방법
    - PTQ는 QAT에 비해 시간과 계산 자원이 적게 들어 널리 사용됨

- PTQ의 핵심은 **각 레이어의 가중치**와 **활성화 함수**에 대한 **양자화 파라미터(스케일링 계수, 영점)를 설정**하는 것

$$x_{int} = \text{clamp}(\lfloor \frac{x}{s} \rceil - z, p_{min}, p_{max}).$$

  - x: 입력 데이터(가중치 또는 활성화 함수 값)
  - s: 스케일링 팩터
  - ⌊x/s⌉: 입력 데이터 x를 스케일링 팩터 s로 나누고 가장 가까운 정수로 반올림
  - z: 오프셋 값
  - p_min, p_max: 출력 값의 최소값과 최대값을 제한하는 경계 값(clamp 함수)

  1️⃣ **스케일링**: 입력 데이터 x를 스케일링 팩터 s로 나누어 크기를 조정

  2️⃣ **반올림**: 스케일링된 값을 가장 가까운 정수로 반올림

  3️⃣ **오프셋 조정**: 반올림된 값에서 오프셋 z를 빼서 값을 조정

  4️⃣ **경계 제한**: clamp함수를 사용하여 출력 값의 범위를 p_min과 p_max 사이로 제한

  5️⃣ **양자화**: 위 식의 결괏값인 x_int는 입력 데이터 x가 양자화된 후의 정수 값을 의미

- 일반적으로 이 파라미터들은 원본 텐서와 양자화된 텐서 간의 **MSE**(평균 제곱 오차)**를 최소화**하도록 선택됨
- 다른 거리 지표(L1 거리, 코사인 거리, KL 발산 등)도 사용될 수 있음

- 보정 샘플(Calibration Samples)
  - PTQ를 수행하려면 네트워크의 활성화 값을 계산하기 위해 소량의 보정 샘플이 필요함
  - 선택된 보정 샘플에 따라 양자화 파라미터가 달라질 수 있음
  - 보정 샘플 수의 영향에 대한 연구도 진행됨
- 제로샷 양자화(Zero-shot Quantization, ZSQ)
  - ZSQ는 PTQ의 특수한 경우로, 네트워크 내부 정보(배치 정규화 레이어의 평균/분산 등)를 활용하여 보정 샘플을 생성함

- 이를 통해 실제 샘플의 활성화 분포와 유사한 분포를 가진 입력 샘플을 생성
- 확산 모델의 노이즈 입력에서 이미지 생성 과정은 이전 ZSQ 방법과 다르므로, 이에 대한 새로운 접근이 필요함

📍 **보정 샘플(Calibration Sample)**

- 보정 샘플은
출력된 확률 분포와 **실제 정답 분포** 사이의 차이를 측정하고 이를 보정하기 위해 사용되는 데이터 샘플
- 딥러닝 모델에서 종종
과도하게 확산하거나 **불확실한 예측**을 내놓는 문제가 존재
➡️ 보정 샘플을 통해 이러한 문제를 해결하고
모델의 출력 확률이 실제 정답 분포와 잘 맞도록 조정할 수 있음

3. PTQ on Diffusion Models

3.1 Preliminaries

📌 Diffusion Models

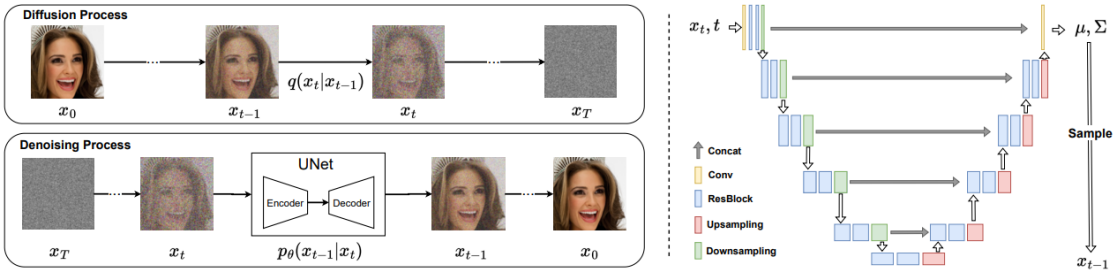


Figure 2. Brief illustration of Diffusion Model. The inference of diffusion models is extremely slow due to their two fundamental characteristics: **(1, Left)** the lengthy iterative process for denoising from noise input to synthetic images; and **(2, Right)** the cumbersome networks for estimating the noise in each denoising iteration.

- 확산 확률 모델의 개요
  - 확산 확률 모델(DPM)은 변분 하한(Variational Lower Bound, LVLB)을 최적화하여 학습됨
  - DPM을 간단히 일반화하여 PTQ 기법을 적용하기는 어려움

📌 **변분 하한(Variational Lower Bound)**

변분 추론(Variational Inference): 복잡한 분포를 간단한 분포로 근사하는 기법
- 변분 하한: 변분 추론 과정에서 최적화하는 목적 함수
- 변분 하한은 원래 분포와 근사 분포 사이의 거리를 최소화하는 것이 목표

변분 하한의 정의
- 원래 분포를 p(x, z), 근사 분포를 q(z|x)라고 할 때, 변분 하한은 다음과 같음:

➡️

log p(x) ≥ L(x, q) = Eq[log p(x, z)] - Eq[log q(z|x)]

-

여기서 Eq[·]는 q(z|x)에 대한 기댓값을 의미함

변분 하한의 의미
- 변분 하한 L(x, q)는 원래 분포 logp(x)의 하한
- 따라서 L(x, q)를 최대화하면 원래 분포 log p(x)를 최대화할 수 있음
- 이를 통해 복잡한 원래 분포 대신 간단한 근사 분포 q(z|x)를 사용할 수 있음

- 확산 과정(Diffusion process)
  - 실제 데이터 분포 x0~q(x0)에서 시작
  - 점진적으로 작은 양의 등방성 가우시안 노이즈를 추가하여 x1, ..., xT의 일련의 잠재 변수를 생성
  - 노이즈 추가 과정(전체 확산 과정)은 다음과 같은 수식으로 표현됨

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \quad (2)$$

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \quad (3)$$

    - T가 충분히 크고 βt(노이즈 추가 비율)의 스케줄이 잘 정의되면 xT는 등방성 가우시안 분포와 동일해짐
- 임의 시점 t에서의 샘플링
    - 입력 x0에 대해 임의의 시점 t에서 xt를 직접 샘플링할 수 있음

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

    - 여기서 αt=1-βt, ᾱt = Π(i=1~t) αi
- 노이즈 추정 모델 학습
    - 실제 데이터 분포 q(x0)는 알 수 없기 때문에, 노이즈 추정 모델 pθ(xt-1|xt)를 신경망으로 학습해야 함
    - 변분 하한(LVLB)을 최적화하여 모델을 학습
    - LVLB =

$$\mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] \geq -\mathbb{E}_{q(\mathbf{x}_0)} \log p_\theta(\mathbf{x}_0)$$

    - LVLB를 KL divergence와 엔트로피 항으로 다시 쓸 수 있음
- 샘플 생성
    - 학습된 노이즈 추정 모델 pθ(xt-1|xt)를 이용하여 반복적으로 xt-1을 샘플링하여 최종 x0를 생성

➡️ DPM은 점진적인 노이즈 추가와 제거 과정을 통해 데이터를 생성하는 모델로, 과정이 매우 복잡하고 시간이 오래걸림. 또한 변분 하한을 최적화하여 모델을 학습하는데, 이는 실제 데이터 분포와 모델 분포의 차이를 최소화하는 것을 목표로 함

📌 Post-training Quantization

- 양자화 과정
    - 가중치 텐서와 활성화 텐서에 대한 양자화 매개변수(s, z)를 선택

    - 이를 통해 full-precision(32비트 부동 소수점) 텐서를 양자화된 텐서로 변환할 수 있음
    - 양자화로 인한 오차를 최소화하는 방향으로 매개변수 값을 결정

$$X_{sim} = s(\text{clamp}(\lfloor \frac{X_{fp}}{s} \rceil - z, p_{min}, p_{max}) + z),$$

        - X_sim: 시뮬레이션을 통해 얻은 full-precision 텐서
        - X_fp: 실제 full-precision 텐서
    - 양자화 오차 L_quant를 최소화하는 것이 목표

$$L_{quant} = \text{Metric}(X_{sim}, X_{fp}),$$

        - Metric은 두 텐서 간의 거리 측정 함수로, 일반적으로 L2 노름이나 MSE가 사용됨
        - Metric은 MSE, cosine distance, L1 distance, KL divergence 등 다양한 방식으로 정의할 수 있음
- 가중치 및 활성화 양자화
    - **가중치 텐서**는 직접 양자화하여 오차를 최소화할 수 있음
    - 하지만 **활성화 텐서**는 입력 데이터가 없으면 양자화할 수 없음

    ➡️ **calibration dataset**이라는 소규모 unlabeled 입력 데이터를 사용하여 **full-precision 활성화 텐서를 수집**함
- PTQ 적용 과정
    - 어떤 연산을 양자화할지 선택
        - 일부 특수 함수(소프트맥스 ,GeLU 등)는 full-precision으로 유지
    - calibration dataset을 수집
        - 실제 데이터 분포와 유사해야 함
    - 가중치 텐서와 활성화 텐서에 대한 양자화 매개변수를 선택하여 양자화 오차를 최소화함
    - 양자화 과정을 수식으로 나타내면 다음과 같음:

$$\arg\min_{s,z} L_{quant}.$$

- s는 스케일링 팩터, z는 양자화 레벨, Lquant는 양자화 손실 함수
- 양자화 손실 함수 Lquant는 원래의 full-precision 텐서 XF와 양자화된 텐서 XQ 사이의 차이를 최소화하는 것을 목표로 함

3.2 Exploration on Operation Selection

- 확산 모델의 이미지 생성 과정
  - 확산 모델은 반복적으로 xt에서 xt-1을 생성함
  - 각 시간 단계에서 네트워크의 **입력은 xt와 t**이고, **출력은 평균 μ와 분산 Σ**
  - xt-1은 이 분포에서 샘플링됨
- 확산 모델 네트워크 구조
  - 확산 모델 네트워크는 일반적으로 UNet과 같은 CNN 구조를 가짐
  - 이전 PTQ 방법과 마찬가지로, 계산이 복잡한 **합성곱 레이어**와 **완전 연결 레이어**를 양자화해야 함
  - 배치 정규화는 합성곱 레이어에 통합될 수 있음
  - SiLU, softmax와 같은 특수 함수는 full-precision으로 유지
- 추가 고려 사항
  - 네트워크의 출력 μ와 σ를 양자화할 수 있는가?
  - 샘플링된 이미지 xt-1을 양자화할 수 있는가?
    - 실험 결과, μ, σ, xt-1은 양자화에 민감하지 않은 것으로 관찰됨
    - 이들 연산을 양자화할 수 있다고 제안!

Table 1. Exploration on operation selection for 8-bit quantization. The diffusion model is for unconditional ImageNet 64x64 image generation with a cosine noise schedule. DDIM (250 timesteps) is used to generate 10K images. IS is the inception score.

	IS	FID	sFID
FP	14.88	21.63	17.66
quantize μ	15.51	21.38	17.41
quantize Σ	15.47	21.96	17.62
quantize x_{t-1}	15.26	21.94	17.67
quantize μ+Σ+x_{t-1}	14.94	21.99	17.84

3.3 Exploration on Calibration Dataset

- 확산 모델의 PTQ를 위한 Calibration Dataset 구축
  - 일반적인 네트워크 양자화에서는 학습 데이터셋을 calibration dataset으로 사용할 수 있음
  - 하지만 확산 모델의 경우, 학습 데이터셋인 x0은 네트워크 입력이 아니고 **네트워크의 실제 입력은 생성된 샘플 xt**임
  - 그렇다면 어떤 샘플을 calibration dataset으로 사용해야 할까
    - 확산 과정에서 생성된 샘플?
    - 디노이징 과정에서 생성된 샘플?
    - 어떤 시간 단계(t)에서 샘플을 수집해야 할까?
- 다양한 PTQ 기준 탐색
  - 여러 가지 직관적인 PTQ 기준을 종합적으로 조사하여 4가지 의미 있는 관찰 결과를 얻었으며, 이를 바탕으로 방법을 설계함
  - 이 방법을 통해 8비트 PTQ 확산 모델이 32비트 모델과 유사한 성능을 달성할 수 있었음

📌 **4가지 의미 있는 관찰 결과**

0️⃣ **번째 관찰**: 활성화 분포가 시간 단계에 따라 변화한다

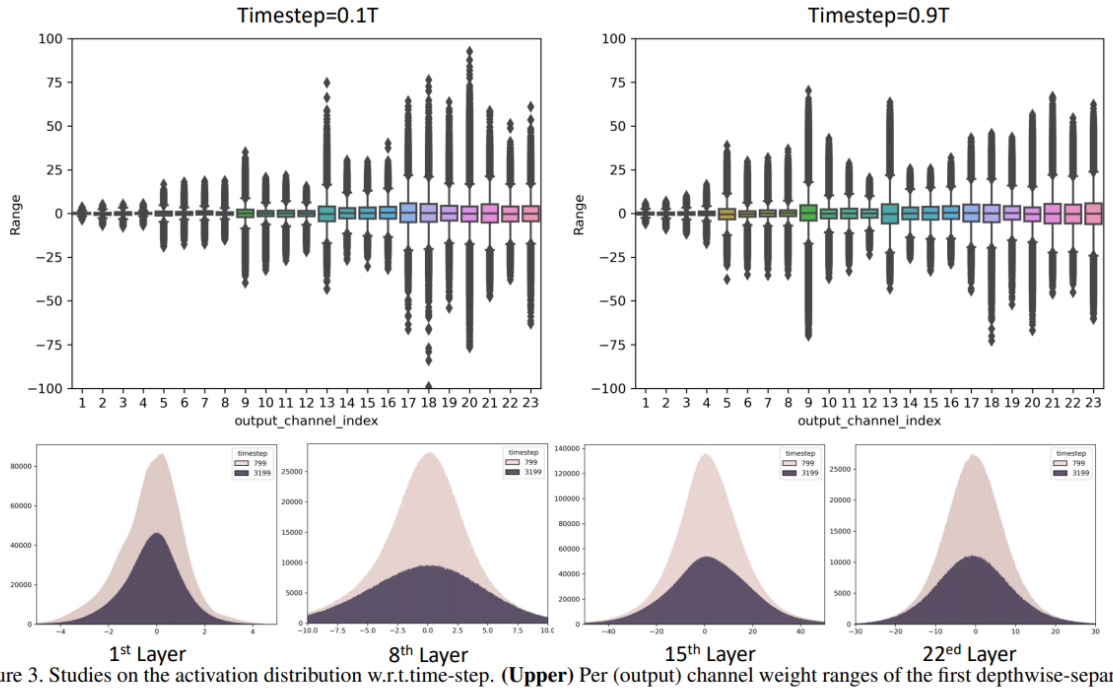


Figure 3. Studies on the activation distribution w.r.t.time-step. **(Upper)** Per (output) channel weight ranges of the first depthwise-separable layer in diffusion model on different timestep. In the boxplot, the min and max values, the 2nd and 3rd quartile, and the median are plotted for each channel. We only include the layers in the decoder of UNet for noise estimation, as the ranges of the encoder and decoder are quite different. **(Bottom)** Histograms of activations on different time-steps by various layers. We can observe that the distribution of activations changes dramatically with time-step, which makes traditional single-time-step PTQ calibration methods inapplicable for diffusion models.

- 확산 모델의 출력 분포를 시간 단계별로 분석한 결과, **시간 단계에 따라 큰 차이**가 있음을 확인함
- 이는 기존 PTQ calibration 방법이 시간에 따라 변하지 않는 것을 가정하기 때문에, 확산 모델에 적용하기 어렵다는 것을 의미함

1️⃣ **번째 관찰: 디노이징 과정에서 생성된 샘플**이 calibration에 더 도움이 된다

Table 2. Results of calibration using noise (input of the denoising process), image (input of the diffusion process), and samples generated by Eq. 3 (Mimicking the diffusion model training).

	IS ↑	FID ↓	sFID ↓
Noise Samples	13.92	33.15	20.38
Image Samples	6.90	128.63	90.04
Training-mimic	12.91	34.55	25.18

- 확산 과정의 원본 이미지와 디노이징 과정의 가우시안 노이즈를 각각 캘리브레이션 데이터셋으로 사용하여 실험한 결과, **디노이징 과정의 샘플이 더 효과적**인 것으로 나타남

- 또한 "이미지 + 가우시안 노이즈" 방식으로 학습 데이터셋에서 샘플을 생성하는 "training-mimic" 기준선 실험에서도, 디노이징 과정의 샘플이 더 좋은 성능을 보임

2️⃣ **번째 관찰: 실제 이미지 x0에 가까운 xt 샘플**이 calibration에 더 도움이 된다

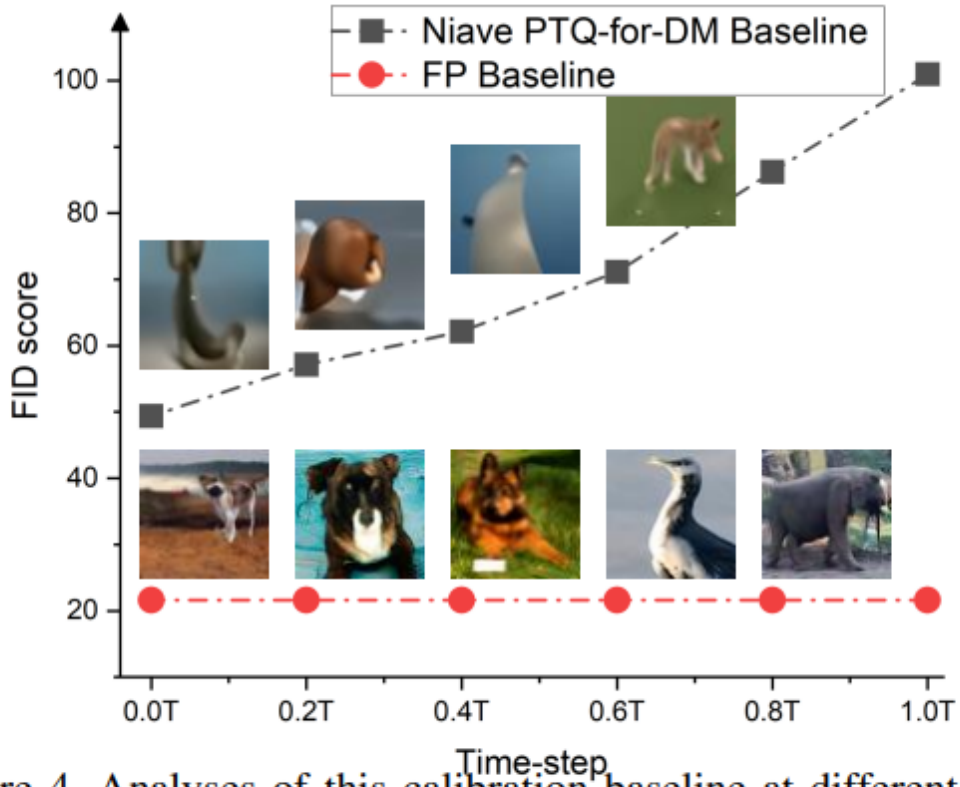


Figure 4. Analyses of this calibration baseline at different time-steps. FP Baseline denotes the 32-bit model, which does not require to be calibrated.

- 앞서 설명한 관찰 결과를 바탕으로, 저자들은 기존 연구에 기반한 DM용 PTQ 캘리브레이션 기준선을 수립함
- 이 기준선은 시간 단계 t에 해당하는 xt 샘플 집합을 캘리브레이션 데이터셋으로 사용하는 간단한 접근임
- 실험 결과, 이 기준선으로 캘리브레이션한 8비트 모델은 만족스러운 이미지를 생성하지 못함
- 하지만 실험에서 중요한 관찰 결과를 얻었는데, **시간 단계 t가 실제 이미지 x0에 가까워질수록 PTQ 캘리브레이션이 더 도움이 된다는 것**입니다.
- 이는 디노이징 과정에서 **t가 감소할수록 pθ(xt-1|xt) 네트워크의 출력 분포가 실제 이미지 분포와 유사**해지기 때문임

3️⃣ **번째 관찰**: 동일한 시간 단계의 샘플이 아닌, **다양한 시간 단계의 샘플**이 필요하다

- 기존 PTQ 캘리브레이션 방법은 단일 시간 단계를 가정하지만, 확산 모델의 경우 **다중 시간 단계**를 다루어야 함
- 따라서 저자들은 **다양한 시간 단계의 샘플로 구성된 캘리브레이션 데이터셋이 필요하다**고 가설을 세움
- 이를 검증하기 위해 0부터 T 사이의 균일 분포에서 **N개의 시간 단계 ti를 샘플링**하고, 이에 해당하는 **xti를 생성하여 캘리브레이션 데이터셋을 구축**함
- 실험 결과, 위 방법이 더 효과적인 것으로 나타나며 시간 단계 차이를 반영하는 캘리브레이션 데이터셋이 필요하다는 것을 검증했음

📌 Normally Distributed Time-step Calibration

- 앞서 관찰한 내용을 바탕으로 확산 모델에 적합한 PTQ calibration 데이터셋 수집 방법을 제안

  1️⃣ **디노이징 과정으로 생성**: 전체 정밀도 확산 모델을 사용하여 노이즈 xT에서 시작하여 xt를 생성함

  2️⃣ **실제 이미지 x0에 상대적으로 가까운 샘플**: 시간 단계 t가 T/2 이하인 샘플을 생성함

  3️⃣ **다양한 시간 단계 포함**: 시간 단계 t를 편향 정규 분포(Skew Normal Distribution)에서 샘플링함

- 위 세 가지 조건들은 서로 상충되는 면이 있어, 균형을 맞추는 것이 중요함
  - 편향 정규 분포 N(μ, T/2)에서 N개의 시간 단계 ti를 샘플링하는데, 여기서 μ는 T/2 이하로 설정하여 실제 이미지에 가까운 샘플을 생성하도록 함
  - 샘플링된 ti와 초기 노이즈 xT를 사용하여 전체 정밀도 확산 모델로 xti를 생성함
  - 최종적으로 {xti} 집합을 캘리브레이션 데이터셋으로 사용

➡️ **확산 모델의 특성을 고려하여 실제 이미지에 가까우면서도 다양한 시간 단계를 포함하는 NDTC 캘리브레이션 데이터셋 수집 방법을 제안**

- 이 NDTC 방법의 효과성은 기존 PTQ 기준선 및 전체 정밀도 확산 모델과 비교하여 평가되었음(Tab. 3, Fig. 6)

Table 3. Quantitative results of the intuitive baselines for the observations and our proposed *NDTC* calibration method. With our method, the performance of PTQ for DM has been significantly improved, even exceeding full-precision DM performance w.r.t.IS and sFID.

	IS ↑	FID ↓	sFID ↓
Full precision DDIM	14.88	21.63	17.66
Baseline in Observation 2	11.92	49.37	41.33
Baseline in Observation 3	14.99	26.19	19.51
NDTC (ours)	**15.68**	**24.26**	**17.28**

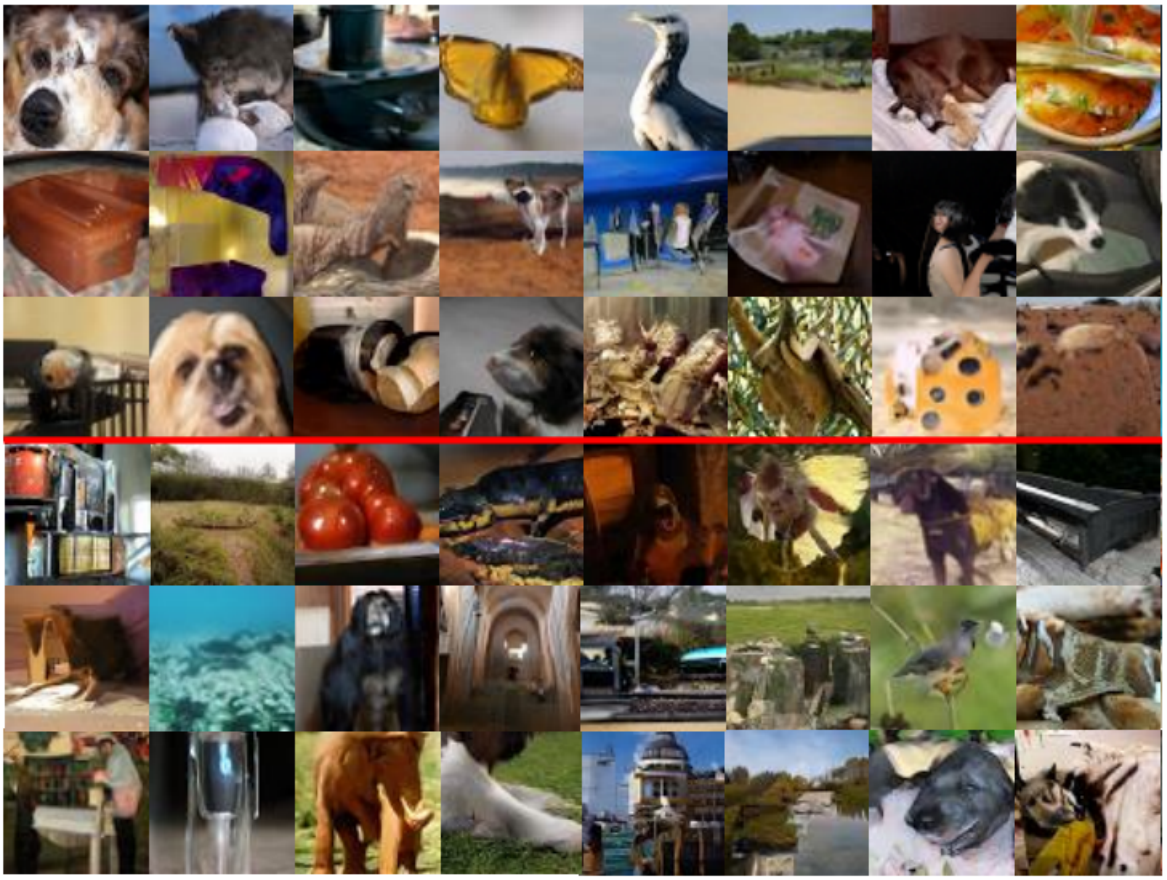


Figure 6. Non-cherry-picked generated samples. **(Upper)** Samples synthesized by full precision DDPM [11]. **(Bottom)** Samples synthesized by 8-bit model quantized by our method. Note that `PTQ4DM` can directly output an 8-bit diffusion with the pre-trained 32-bit diffusion model as input in a *training-free* manner.

3.4 Exploration on Parameter Calibration

- Calibration 샘플을 수집한 후, 확산 모델의 텐서에 대한 **양자화 파라미터**를 선택해야 함
- 텐서를 보정하기 위한 매트릭스를 탐색
- 매트릭스 비교 결과

Table 4. Exploration on calibration metric for 8-bit quantization. We set p=2.4 for MSE metrics.

	IS ↑	FID ↓	sFID ↓
L1 distance	7.38	100.52	63.01
Cosine distance	12.85	34.81	23.75
KL divergence	11.74	47.27	45.08
MSE	13.76	30.46	19.42

  - **MSE**가 L1 거리, 코사인 거리, KL 발산보다 더 좋은 성능을 보임
  - 이에 따라 확산 모델을 양자화할 때 MSE를 매트릭으로 사용하기로 결정

4. More Experiments

- 실험 설정
  - 저자들은 CIFAR10 32×32 이미지와 ImageNet 64×64 다운샘플 이미지를 생성하는 확산 모델을 선택함
  - DDPM(4000 step)과 DDIM(100, 250 step) 모델을 실험에 사용
  - 3.3절에서 제안한 방법으로 1024개의 캘리브레이션 샘플을 생성
  - 네트워크를 8비트로 양자화
  - 10,000개의 이미지를 생성하여 평가를 진행
- 실험 결과

Table 5. Experiment on 8-bit quantized diffusion models generating CIFAR10 image or ImageNet image.

Task	Method	IS ↑	FID ↓	sFID ↓
ImageNet 64x64 DDIM 100 steps	FP	15.38	21.70	17.93
	PTQ4DM	15.52	24.92	17.36
ImageNet 64x64 DDIM 250 steps	FP	14.88	21.63	17.66
	PTQ4DM	15.88	23.96	17.67
ImageNet 64x64 DDPM 4000 steps	FP	15.93	20.82	17.42
	PTQ4DM	15.28	23.64	17.29
CIFAR 32x32 DDIM 100 steps	FP	9.18	10.05	19.71
	PTQ4DM	9.31	14.18	22.59
CIFAR 32x32 DDIM 250 steps	FP	9.19	8.91	18.43
	PTQ4DM	9.70	11.66	19.71
CIFAR 32x32 DDPM 4000 steps	FP	9.28	7.14	17.09
	PTQ4DM	9.55	7.10	17.02

    - 32×32 CIFAR10 이미지 생성 시, 8비트 DDPM 모델이 full-precision DDPM 모델보다 성능이 더 좋음
- 중요한 발견
    - 확산 모델의 두 가지 병목 요인(긴 샘플링 반복, 복잡한 노이즈 추정 네트워크)에 대한 새로운 관점을 제시
    - 이전 연구에서는 반복 길이 측면의 모델 중복성을 밝혀냈지만, 이번 실험에서는 노이즈 추정 네트워크 자체에도 중복성이 있음을 발견함

5. Conclusion

- 확산 모델의 병목 요인

  1️⃣ 노이즈에서 이미지를 샘플링하는 과정의 긴 반복 횟수

  2️⃣ 각 반복 단계에서 노이즈를 추정하는 복잡한 네트워크

- 기존 연구와의 차별점
  - 기존 연구들은 주로 첫 번째 병목 요인(긴 반복 횟수)에 초점을 맞추어 가속화 방법을 제안함
  - 해당 논문에서는 두 번째 병목 요인(노이즈 추정 네트워크)에 주목하여 새로운 접근법을 제시
- 제안하는 방법: PTQ4DM
  - "Post-Training Quantization for Diffusion Models (PTQ4DM)"이라는 방법을 제안
  - 위 방법을 통해 사전 학습된 확산 모델을 8비트로 직접 양자화할 수 있으며, 성능 저하가 크지 않음
  - 또한 이 방법은 DDIM과 같은 기존의 빠른 샘플링 기법과 함께 사용할 수 있음

논문에 대한 의견 및 의문점(꼭지)

➡️ 본 논문에서는 양자화 후에도 성능 저하가 크지 않다고 언급되어있는데, 이 부분에 있어서 양자화가 모델의 생성 품질, 다양성, 안정성 등 다양한 측면에 어떤 영향을 미치는지 심층적으로 분석해볼 필요가 있다고 봄