ILLINOIS INSTITUTE OF TECHNOLOGY

JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

PTQ4DM: Post-training Quantization on Diffusion Models

Yuzhang Shang[1,4,*], Zhihang Yuan[2,*], Bin Xie[1], Bingzhe Wu[3], Yan Yan[1]

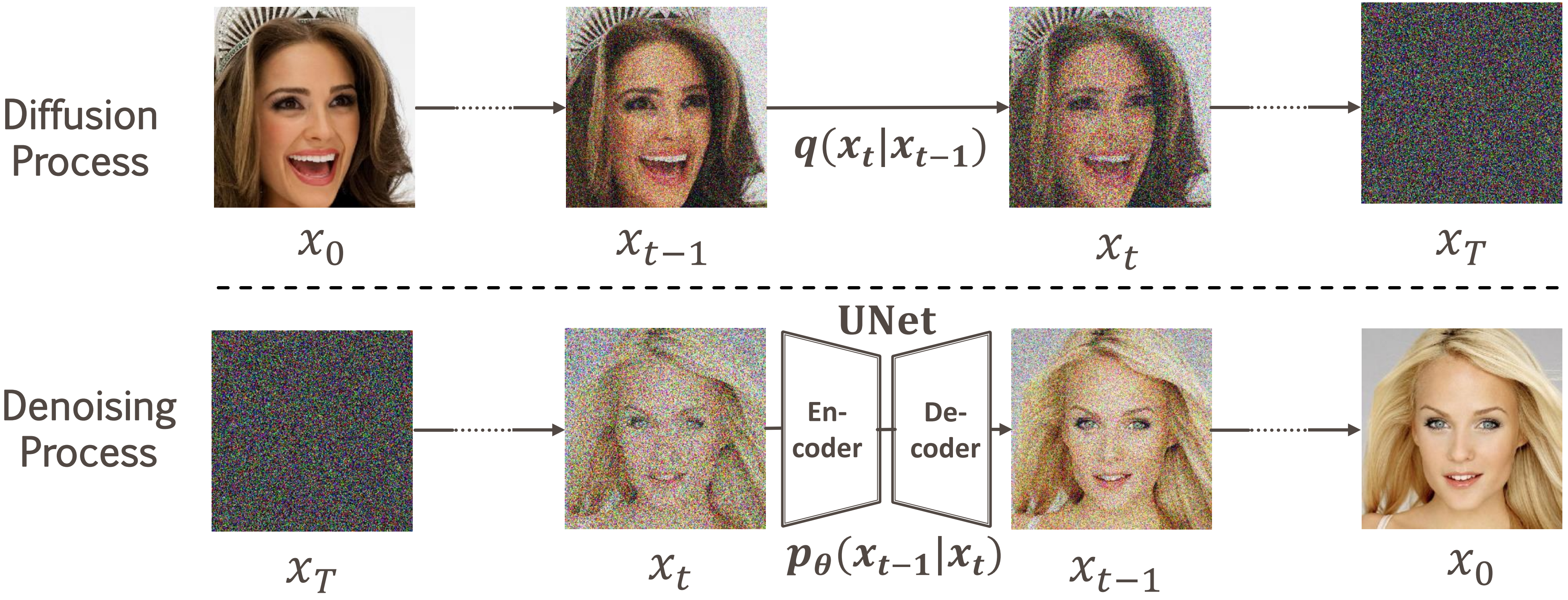
[1]Illinois Institute of Technology
[2]Houmo AI
[3]Tencent AI Lab
[4]Cisco Research

Diffusion Model for AIGC

Input: An astronaut riding a horse in photorealistic style.

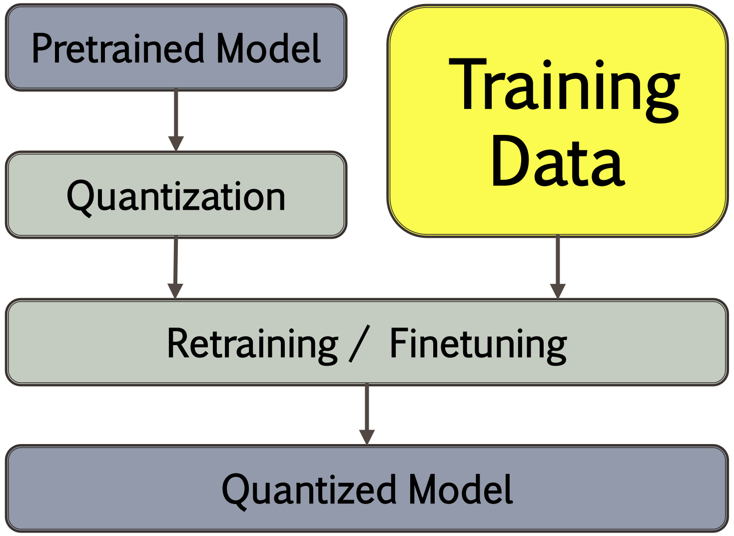
Output:

Diffusion Models

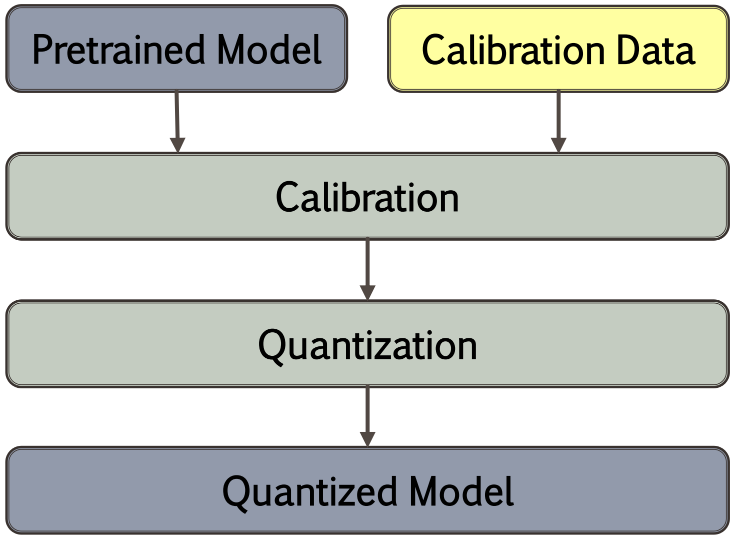


- The goal of diffusion models is to learn the latent structure of a dataset by modeling the way in which data points diffuse through the latent space.
- A neural network is trained to denoise images blurred with Gaussian noise by learning to reverse the diffusion process.

Post Training Quantization (PTQ)



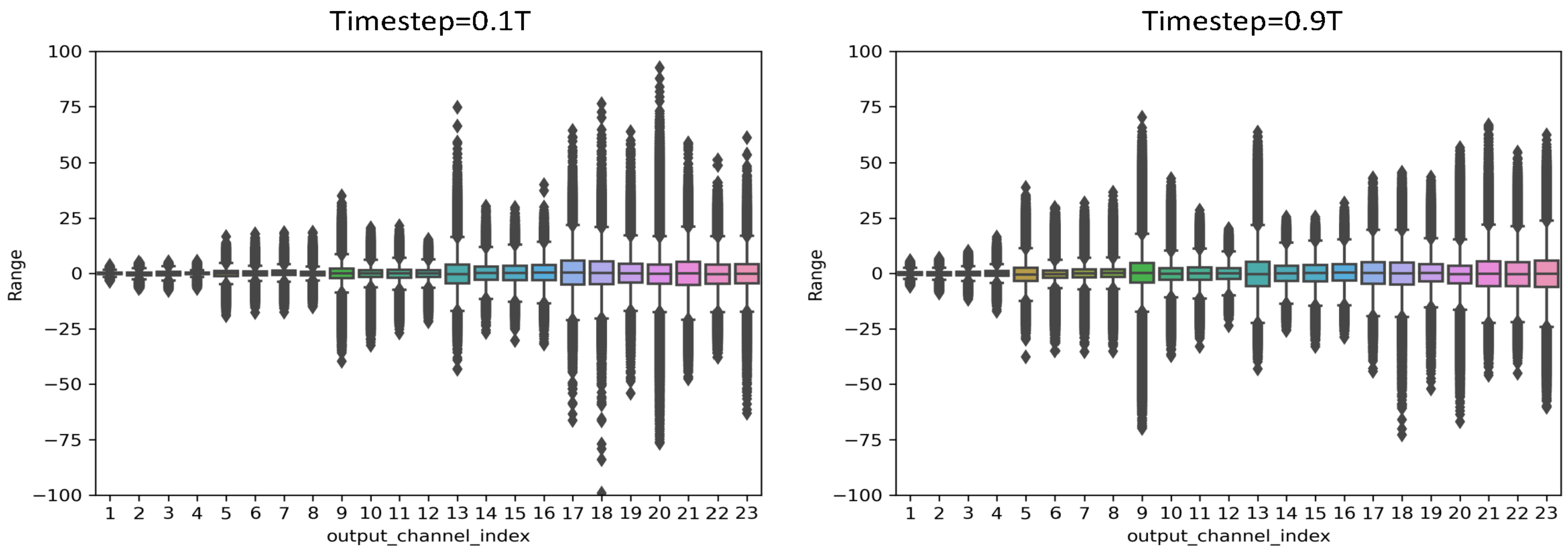
Quantization-Aware Training (QAT)



Post-Training Quantization (PTQ)
Training Data FREE!

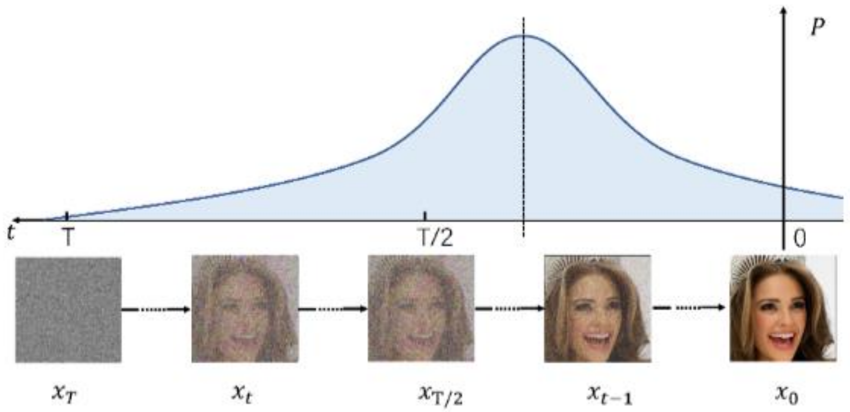
- In QAT, a pre-trained model is quantized and then finetuned using training data to adjust parameters and recover accuracy degradation.
- In PTQ, a pre-trained model is **calibrated using calibration data** (e.g., a small subset of training data) to compute the clipping ranges and the scaling factors.

Exploration



- Generated samples in the denoising process are more constructive for post-training quantization calibration.
- Sample x_t close to real image x_0 is more beneficial for calibration.
- Instead of a set of samples generated at the same time-step, calibration samples should be generated with varying time-steps.

Method:



Algorithm 1 Normally Distributed Time-step Calibration Collection (*DNTC*) Algorithm.

Input: The size of calibration set N, and a mean of the Normal distribution μ, and the full-precision noise estimation network $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ in Eq. 5.

Output: Obtain a Calibration Set $\mathcal{C}$.

Collecting Calibration Set:
for $i = 1$ to N **do**
  Sample t_i from distribution $\mathcal{N}(\mu, \frac{T}{2})$ in Eq. 15;
  Round down t_i into a integer, *i.e.*, $t_i = \lfloor t_i \rfloor$;
  Clamp t_i between $[0, T]$, *i.e.*, $t_i = \text{Clamp}(0, T, t_i)$;
  Produce sample on t_i time-step:
  for $t = T$ to t_i **do**
    Generate a Gaussian Noise $\mathbf{x}_T$ as initialization;
    Sample $\mathbf{x}_{t-1}$ using $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$;
  end for
  Output sample x_{t_i};
end for
Output a calibration set $\mathcal{C} = \{\mathbf{x}_{t_i}\}_{i=1}^N$.

- We desire the calibration samples:
  - Generated by the denoising process with the full-precision diffusion model.
  - Relatively close to clean images and far away from noise.
  - Covered by various time-steps.

Experimental Results

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

- Speedup diffusion models 4 times while maintaining comparable performance.
- PTQ4DM can quantize the pre-trained diffusion models to 8-bit without significant performance loss for the first time. Importantly, PTQ4DM can serve as a plug-and-play module for other state-of-the-art diffusion model acceleration methods.

Contributions

- To accelerate denoising diffusion models, we introduce PTQ into DM acceleration where noise estimation networks are directly quantized in a post-training manner. **This is the first work to investigate diffusion model acceleration from the perspective of training-free network compression**.
- After all-inclusively investigations of PTQ and DMs, we observe the performance drop induced by PTQ for DMs can be attributed to the **discrepancy of output distributions in various time-steps**. Targeting this observation, we explore PTQ from different aspects and propose PTQ4DM.

Thanks!

For our code, please visit our project GitHub website:
https://github.com/42Shawn/PTQ4DM