Low-Bitwidth Floating-Point Quantization for Diffusion Models

Cheng Chen[1], Christina Giannoula[1,2], Andreas Moshovos[1,2]

[1]University of Toronto

[2]Vector Institute

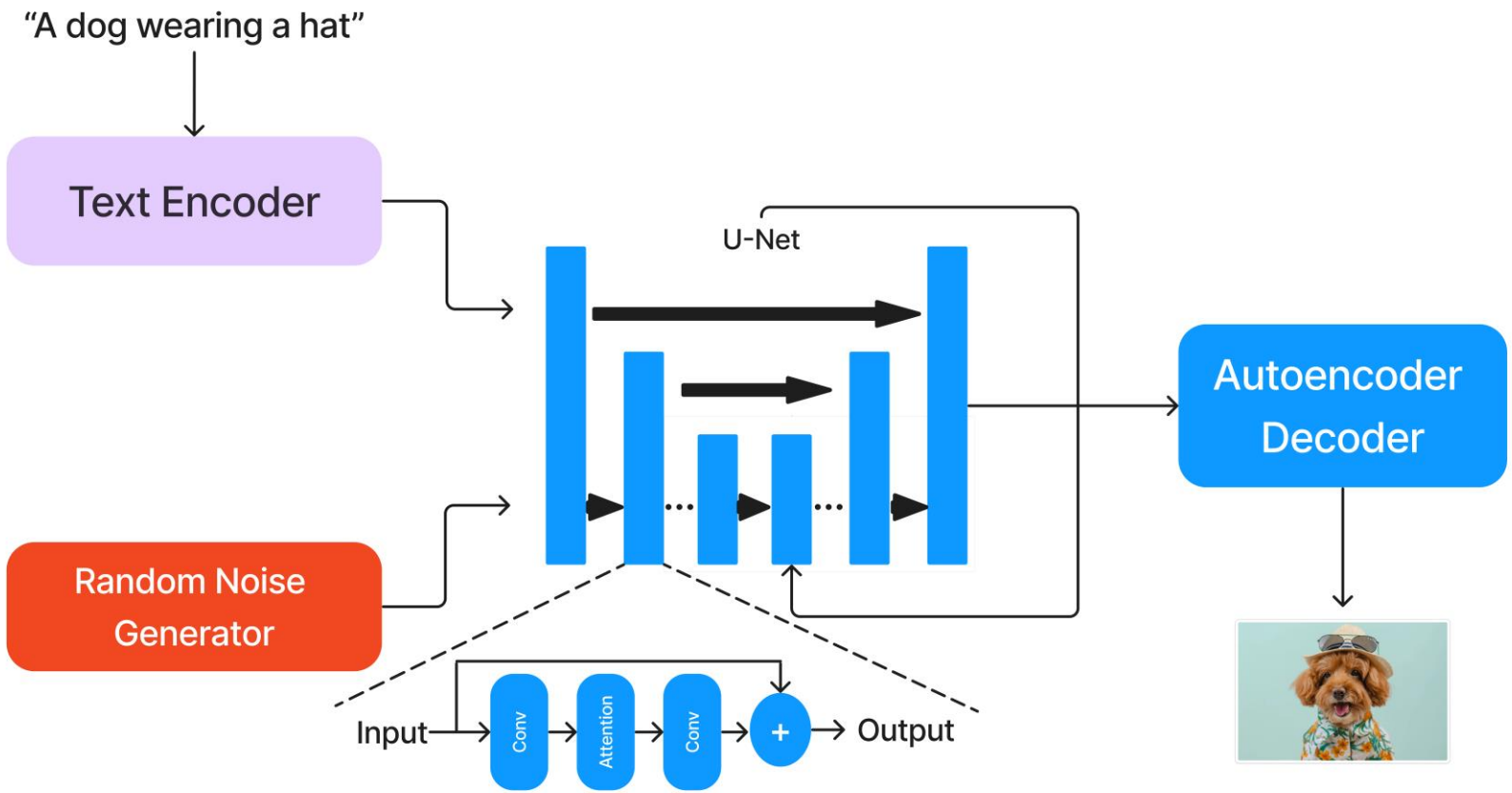
SOTA Image generative models are great

Imagen

DALLE

Midjourney

Example: Stable Diffusion



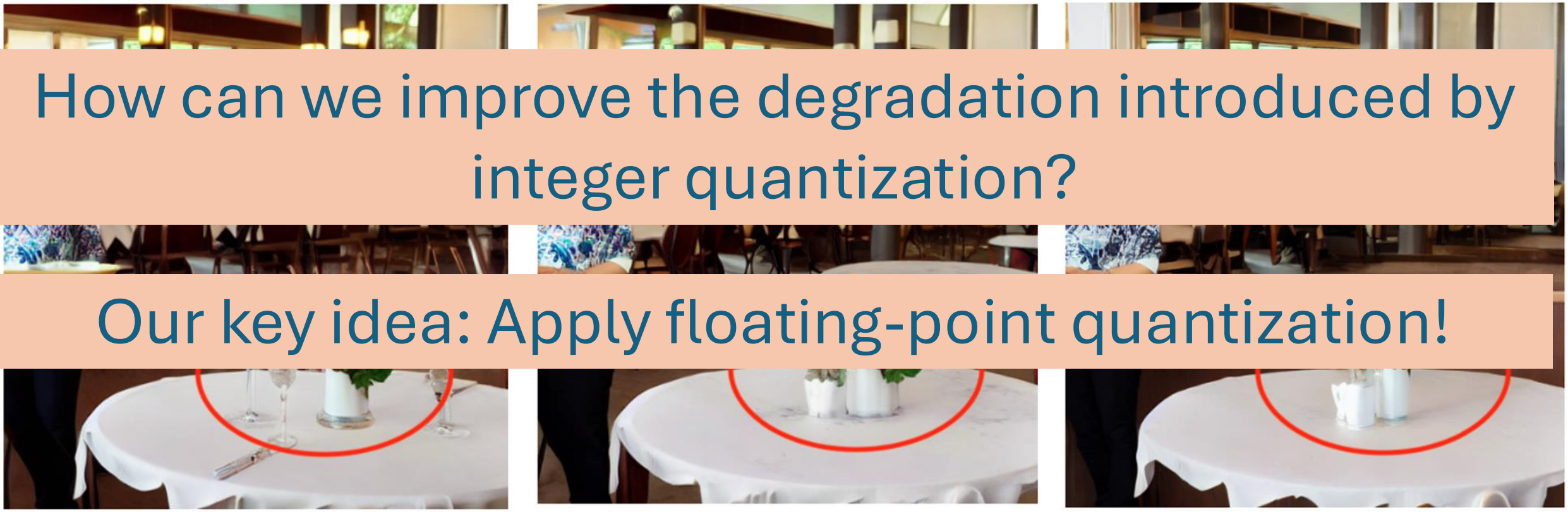
demands significant computing resources

Integer quantized model generated images

Full precision

INT8/INT8

INT4/INT8



How can we improve the degradation introduced by integer quantization?

Our key idea: Apply floating-point quantization!

INT vs. FP Performance

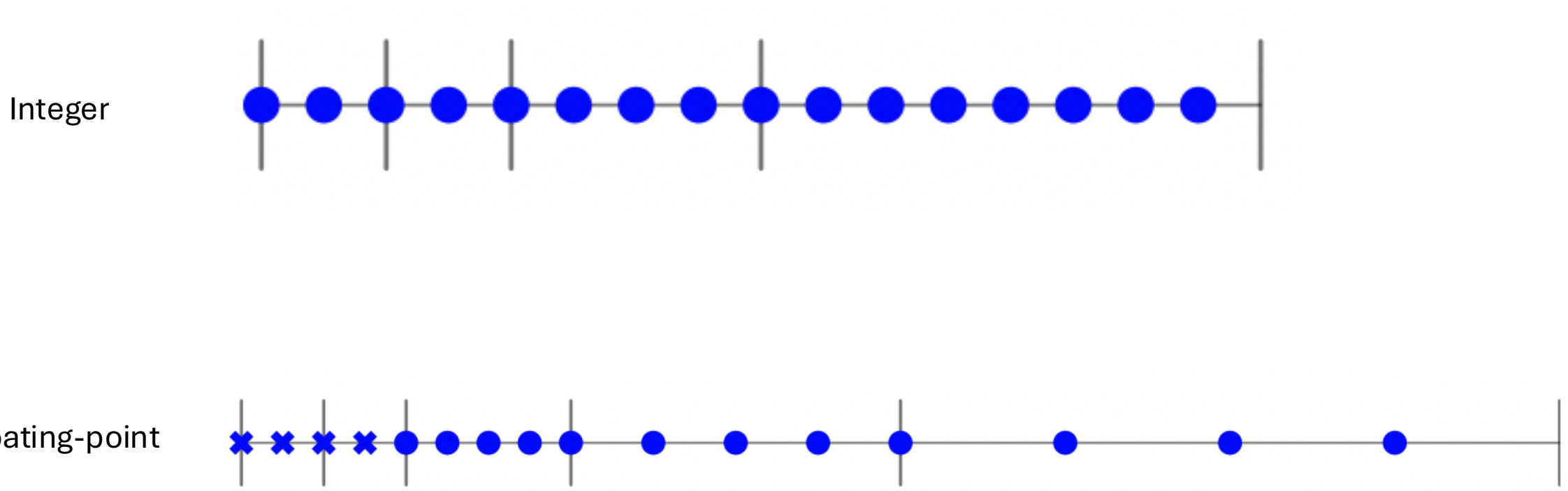
	NVIDIA H100 SXM5[1]	NVIDIA H100 PCIe[1]
Peak FP64[1]	30 TFLOPS	24 TFLOPS
Peak FP64 Tensor Core[1]	60 TFLOPS	48 TFLOPS
Peak FP32[1]	60 TFLOPS	48 TFLOPS
Peak FP16[1]	120 TFLOPS	96 TFLOPS
Peak BF16[1]	120 TFLOPS	96 TFLOPS
Peak TF32 Tensor	500 TFLOPS | 1000 TFLOPS[2]	400 TFLOPS | 800 TFLOPS[2]
Peak FP8 Tensor Core[1]	2000 TFLOPS | 4000 TFLOPS[2]	1600 TFLOPS | 3200 TFLOPS[2]
Peak INT8 Tensor Core[1]	2000 TOPS | 4000 TOPS[2]	1600 TOPS | 3200 TOPS[2]

Table 1. NVIDIA H100 Tensor Core GPU preliminary performance specs

FP8 and INT8 have the same compute throughput and memory footprint

https://developer.nvidia.com/blog/nvidia-hopper-architecture-in-depth/

INT vs. FP



Floating-point representation offers higher precision and a wider range compared to integer representation

Our Contributions

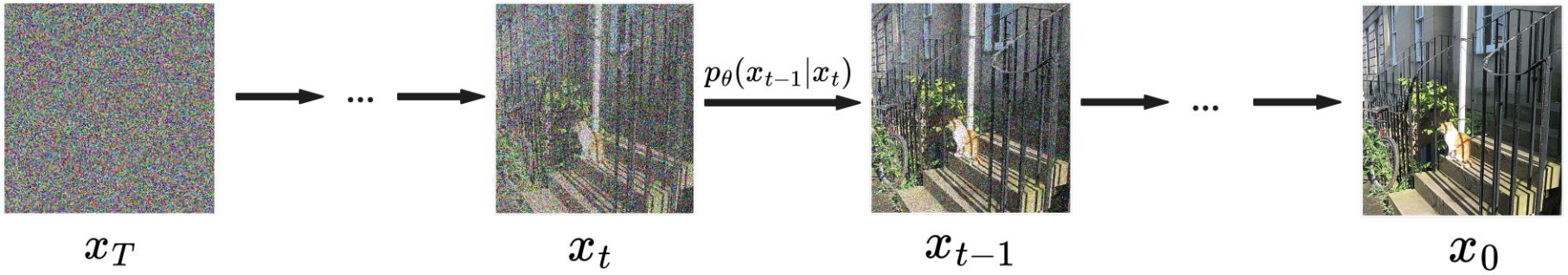
- Apply floating-point quantization on diffusion models(weights to FP4 and activations to FP8)
- Adapt rounding learning from low-bitwidth integer quantization to enable FP4 quantization
- Improve evaluation methodology
  - Avoid contradicting reality

Result highlight: quality

- FP8/FP8 VS. INT8/INT8 1.56x better
- FP4/FP8 VS. INT4/INT8 1.10x better
- Stable Diffusion:
  - FP4/FP8 better than INT8/INT8

How does diffusion model generate new images?

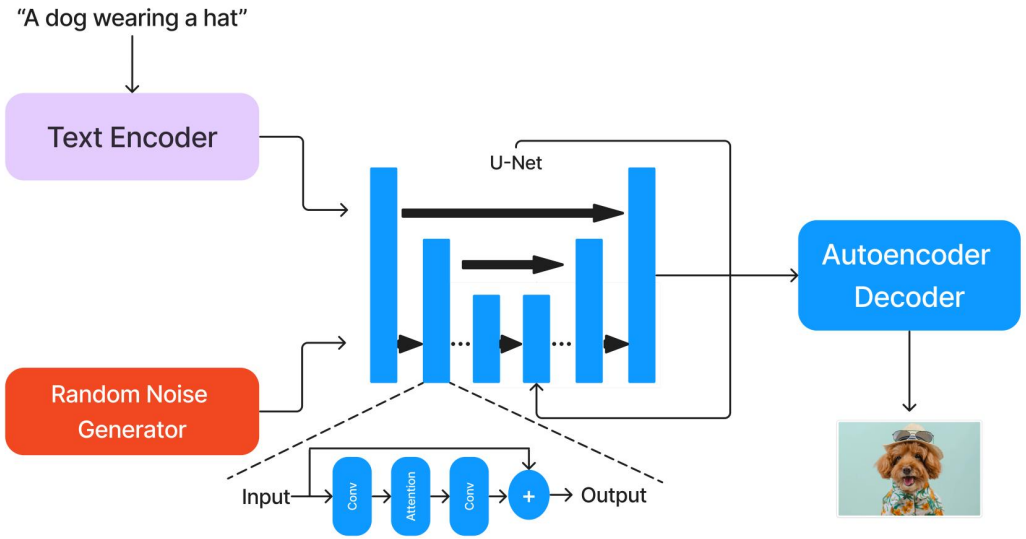
The denoising process: denoise from noisy images



Diffusion models are great, but expensive

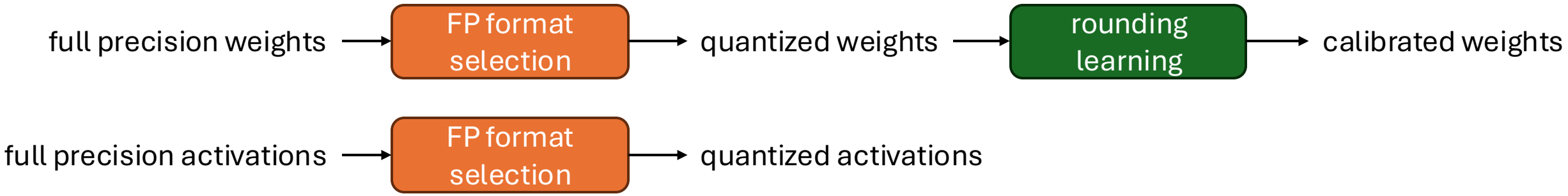
- Stable Diffusion (SD)
  - Total parameters: 1.06B
  - Structure:
    - U-Net: 860M
    - Text Encoder
    - Autoencoder Decoder

- Stable Diffusion XL (SDXL)
  - Total parameters: 3.5B
    - U-Net: 2.6B



Our Floating-Point Quantization Method

Quantization pipeline

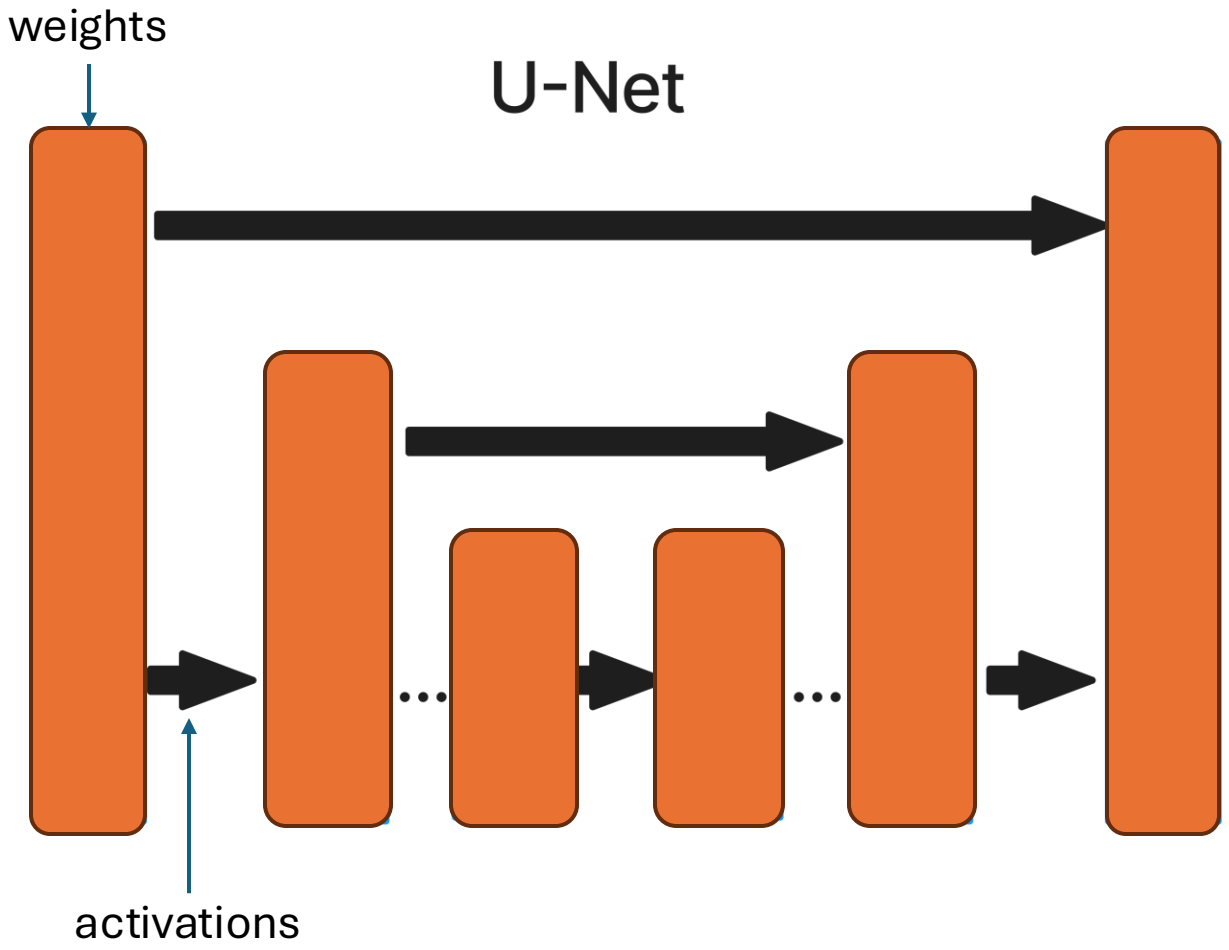


- Step1: search FP formats and bias
- Step2: apply rounding learning to weights to reduce degradation
- Quantization process takes ~20 hours

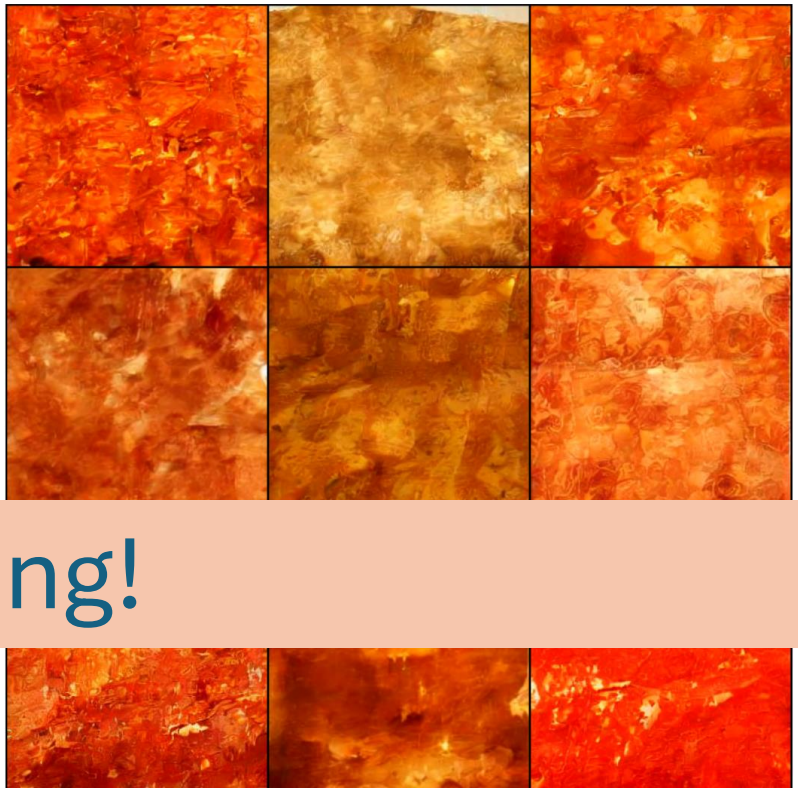
Search space for each floating-point format

- **Encoding candidates**
  - FP8: E2M5, E3M4,E4M3,E5M2
  - FP4: E1M2, E2M1
- **Bias candidates**
  - generate ~100 evenly spaced values between the minimum and maximum of the tensor and calculate the bias for each value
- **Total search space**
  - ~400 for FP8 and ~200 for FP4, for each tensor
  - ~200-600 tensors

Step 1 – greedy search



Randomly sampled images



Need rounding learning!

Full precision

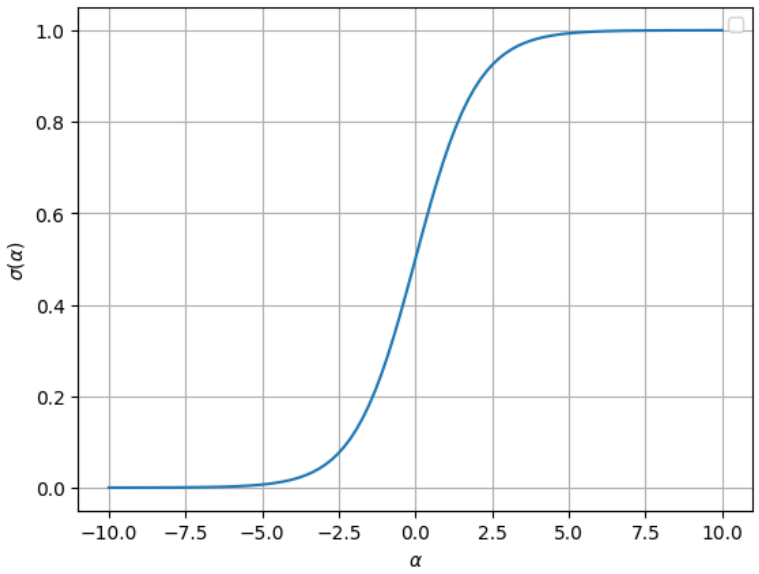
FP8/FP8

FP4/FP8

replace round-to-nearest with learned rounding

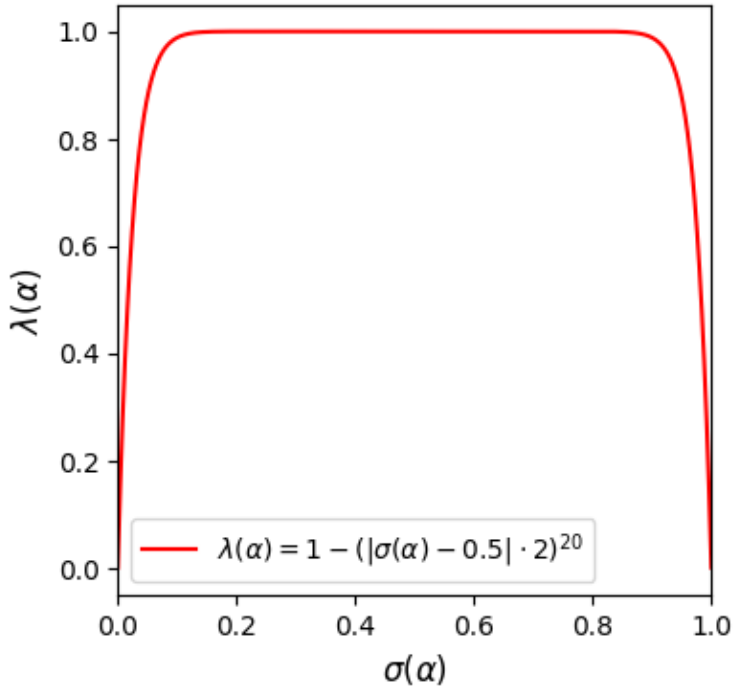
$$W_i^q = s_i \left\lfloor \frac{W_i}{s_i} \right\rceil \longrightarrow W_i^q(\alpha_i) = s_i \cdot \left(\left\lfloor \frac{W_i}{s_i} \right\rfloor + \sigma(\alpha_i) \right)$$

Objective: $\underset{\alpha}{argmin}\ MSE(W(\alpha)^q A, WA)$



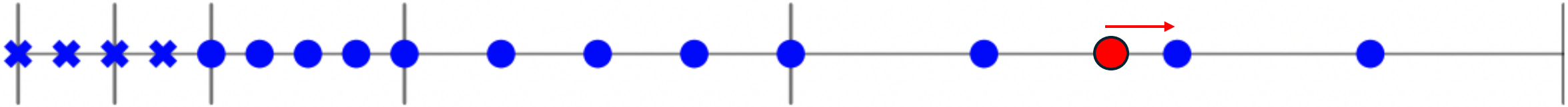
push the sigmoid to the boundary of [0,1]

$$\lambda(\alpha) = 1 - (|\sigma(\alpha) - 0.5| * 2)^{20}$$

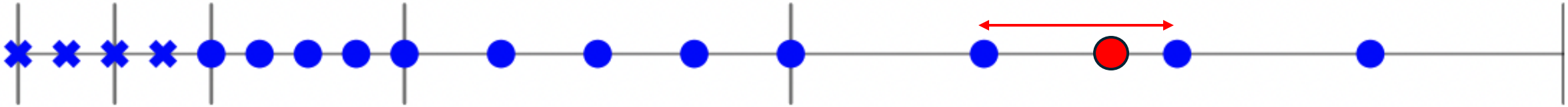


Objective: $\underset{\alpha}{argmin}\ MSE(W(\alpha)^q A, WA) + \lambda(\alpha)$

Round-to-nearest



Rounding learning

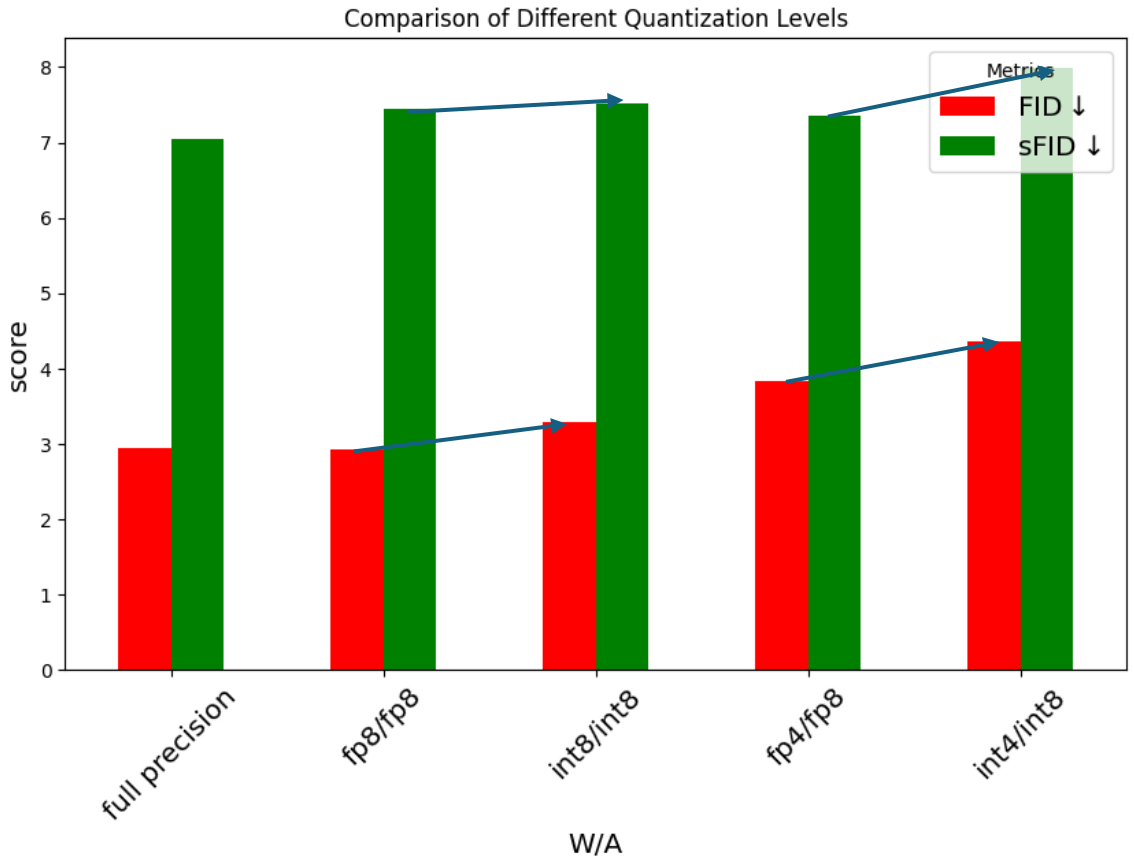


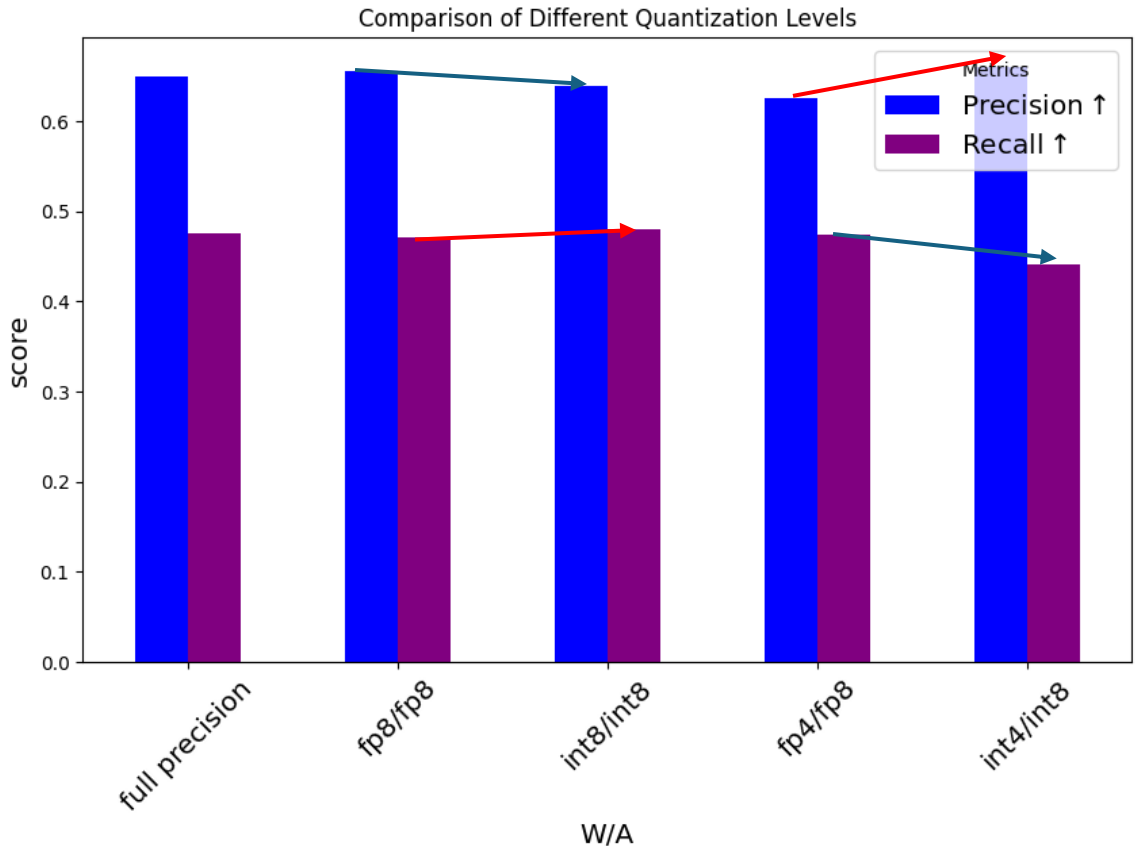
Results

Evaluation Methodology

- **Unconditional generation**
  - FP models outperform INT @ same bitwidth
  - FP4 needs rounding learning
- **Text-to-image generation**
  - FP models outperform INT @ same bitwidth
  - need to improve evaluation methodology
    - Metrics do not reflect reality
- **Metrics**
  - FID, sFID, Precision, Recall

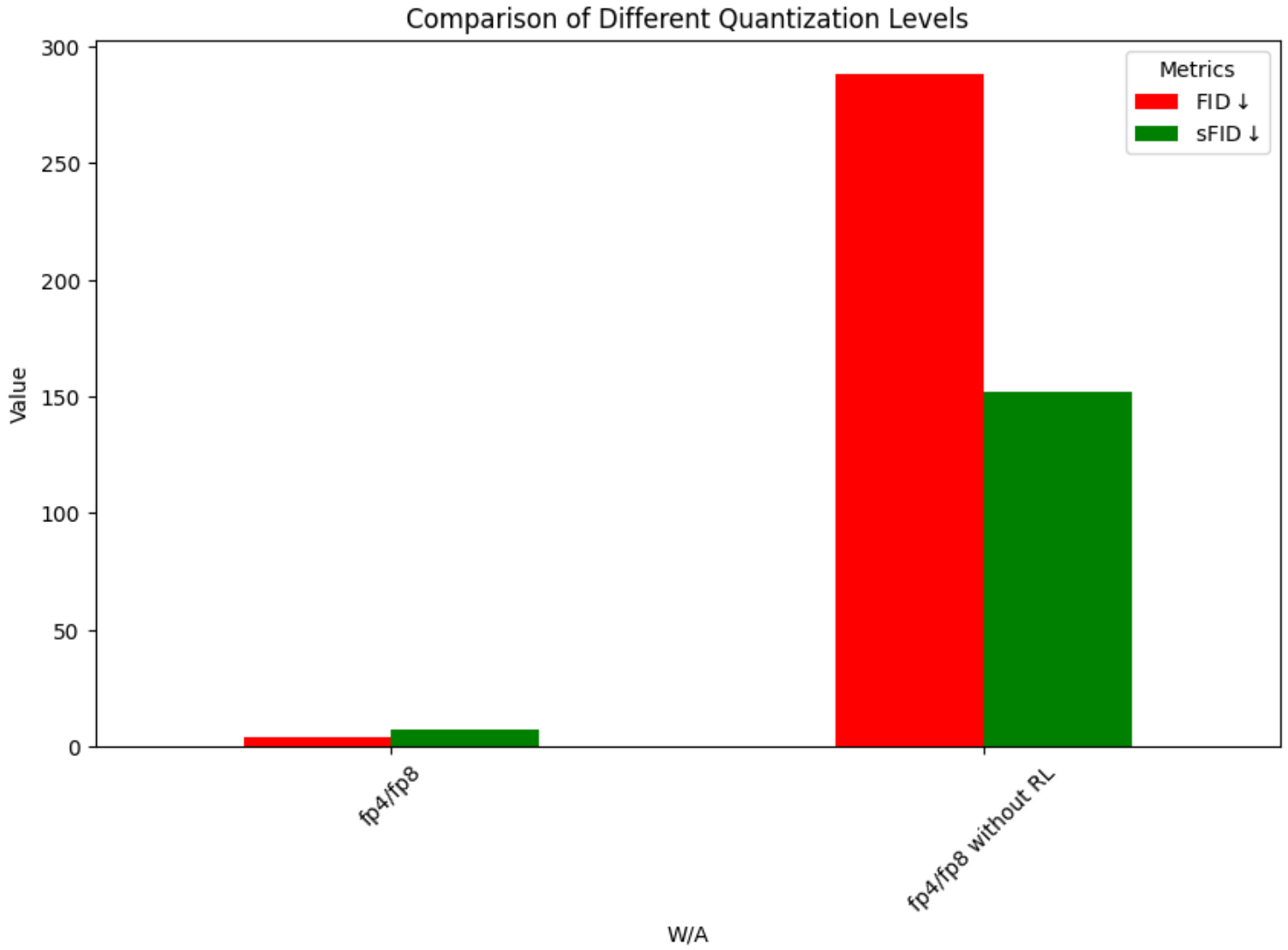
Unconditional Generation





FP quantized models outperform INT at the same bitwidth

Rounding learning significantly reduces degradation at low-bitwidth



Unconditional Generation

(a) full-precision

(b) FP8/FP8

Unconditional Generation

Full precision

FP8/FP8

Hard to tell the difference

Unconditional Generation

(a) full-precision

(d) FP4/FP8 without rounding learning

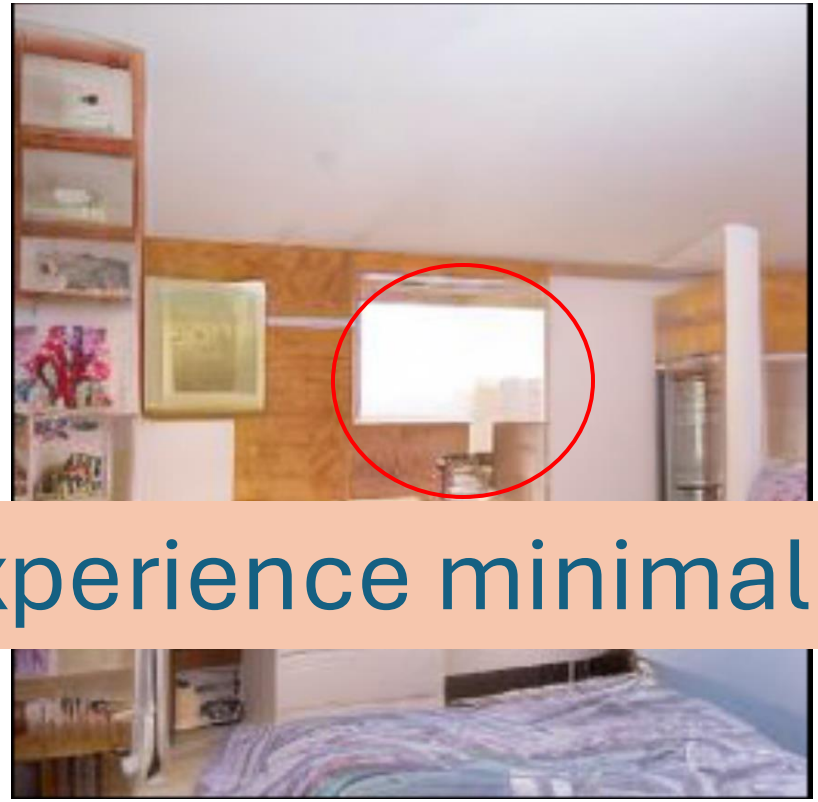
FP4/FP8 generate close to random noise without rounding learning

Unconditional Generation

(a) full-precision

(c) FP4/FP8

Unconditional Generation



Full precision

FP4/FP8

Experience minimal degradation for FP4/FP8

Text-to-Image Generation

Full precision

FP8/FP8

INT8/INT8

Text-to-Image Generation

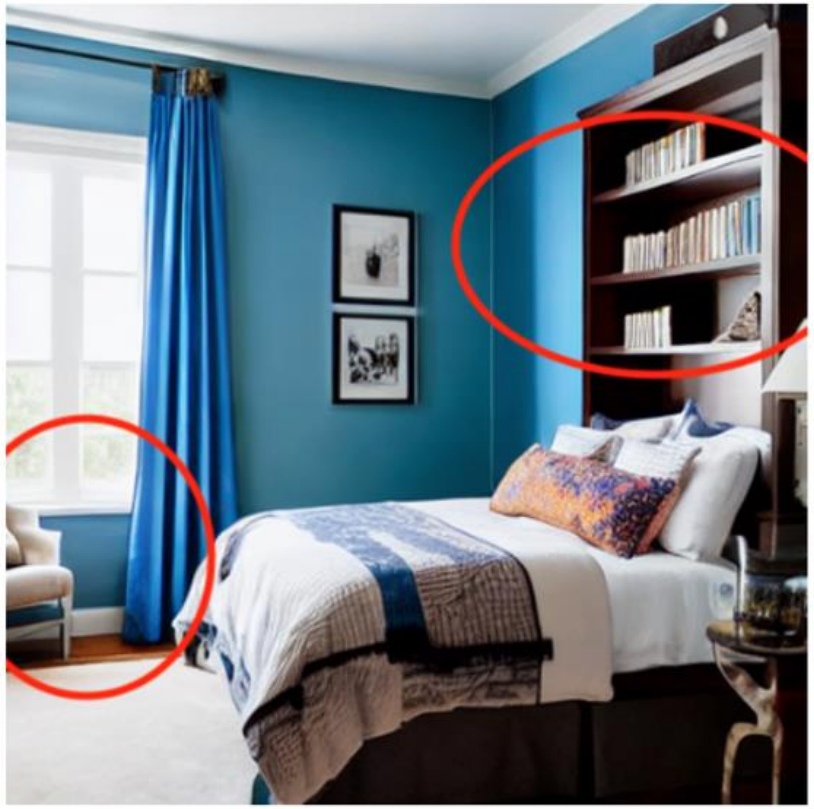
Full precision

FP4/FP8

INT4/INT8

Text-to-Image Generation

Full precision

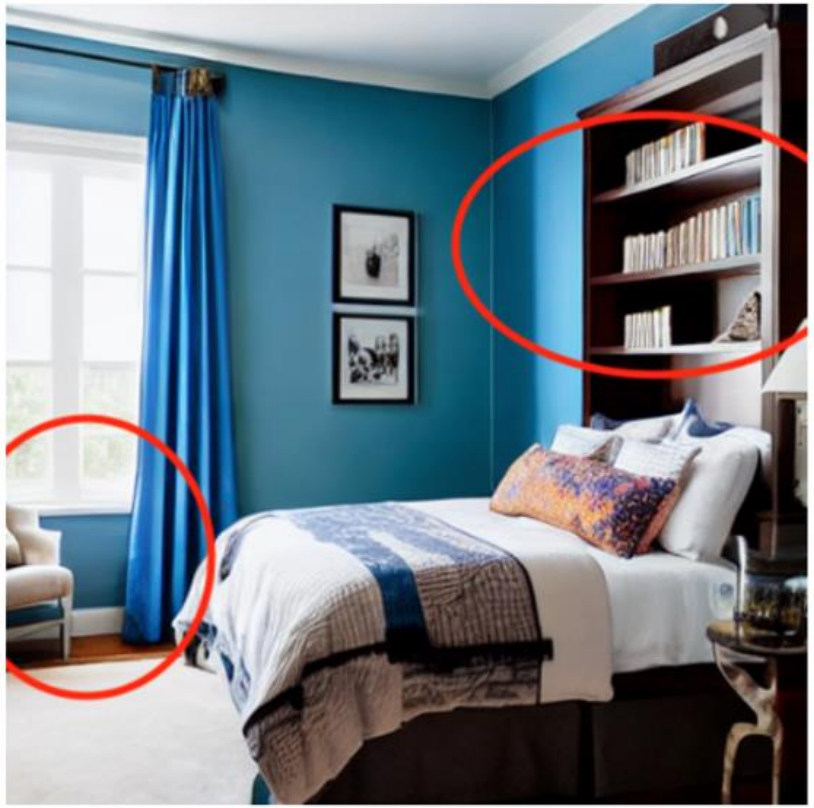


FP8/FP8

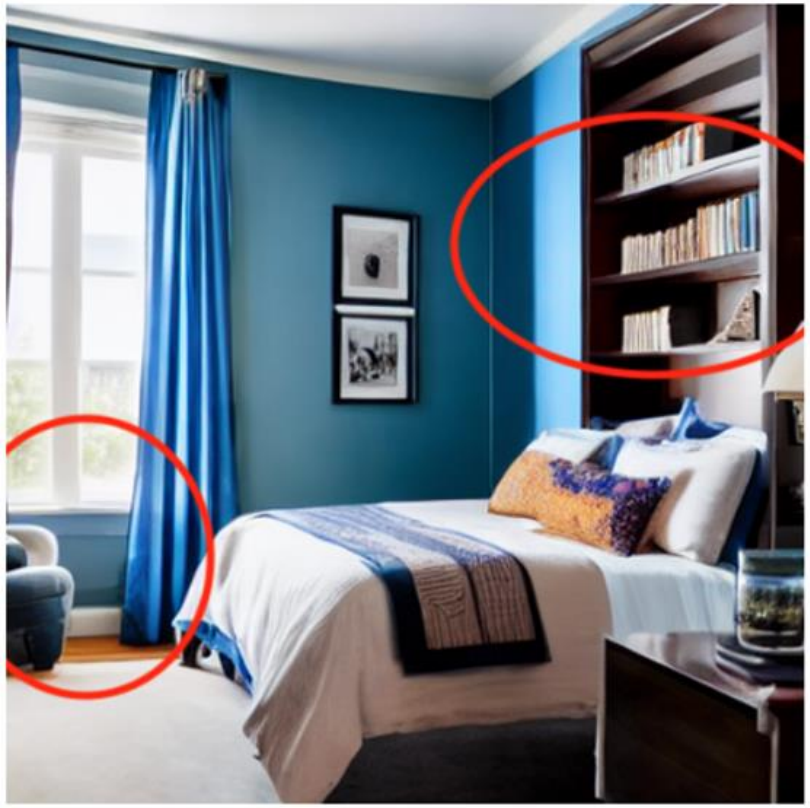
INT8/INT8

Text-to-Image Generation

Full precision



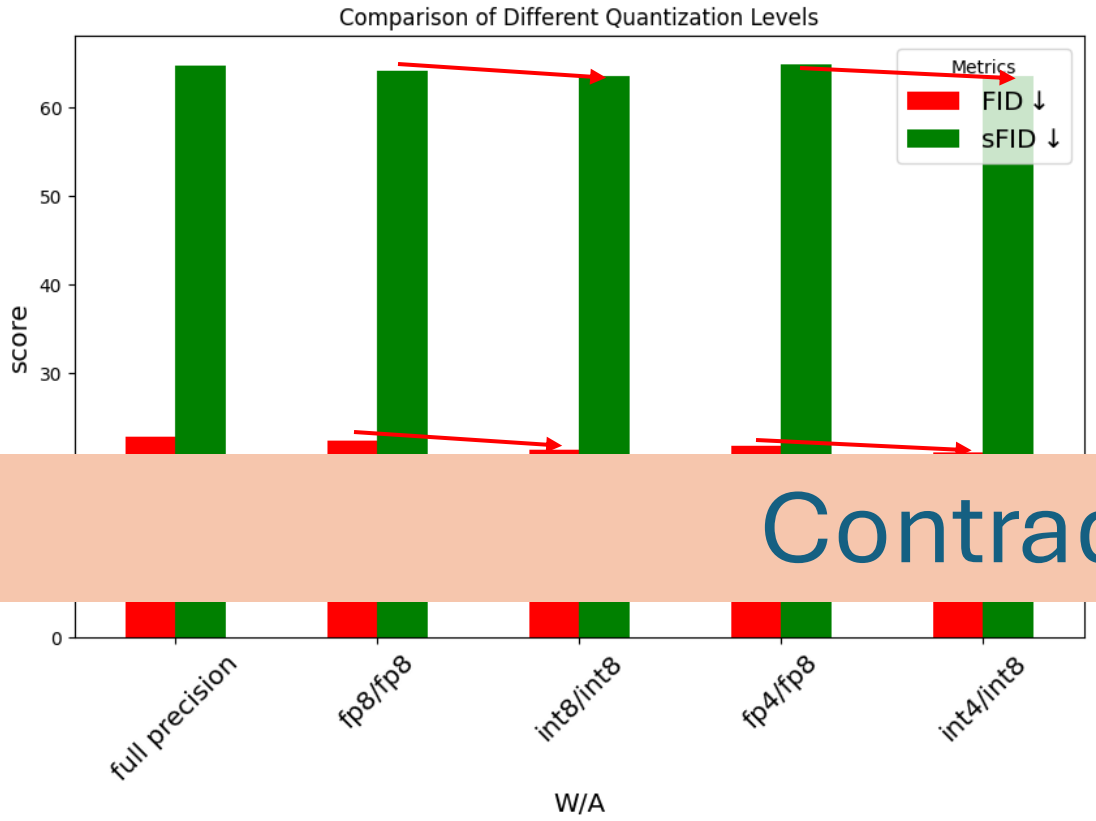
FP4/FP8

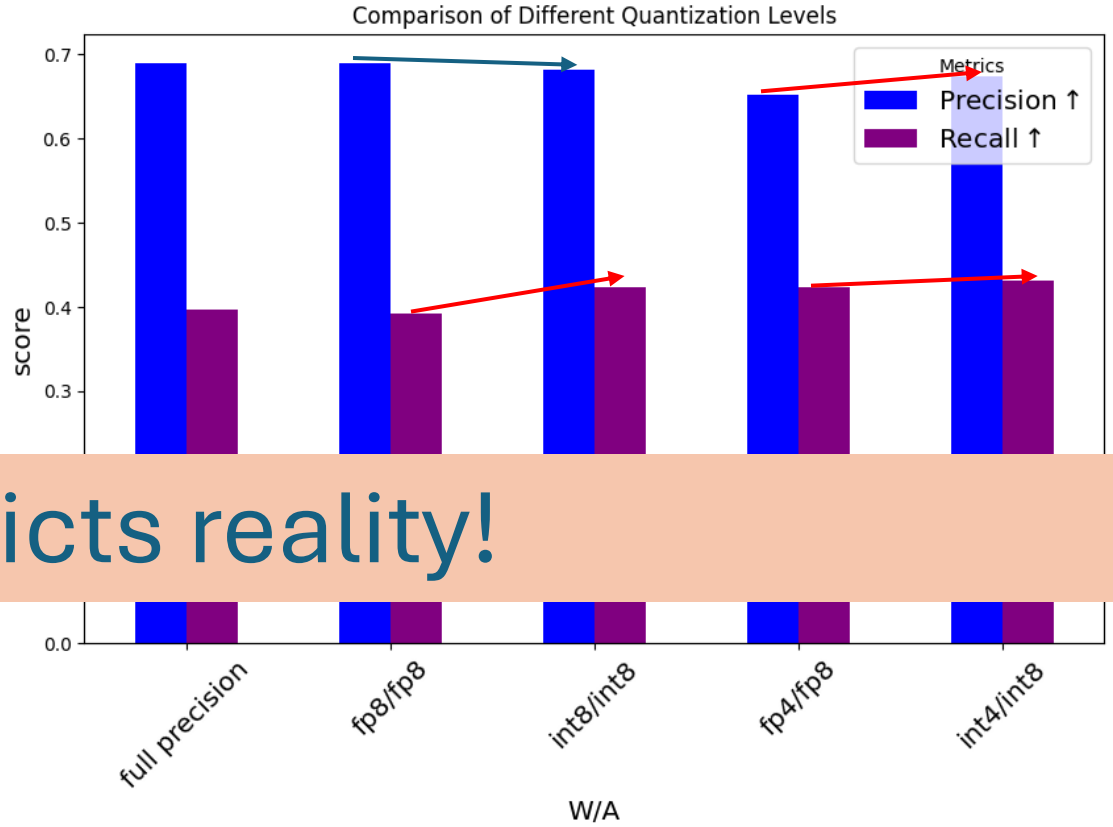


INT4/INT8

Text-to-Image Generation

Reference: MS-COCO



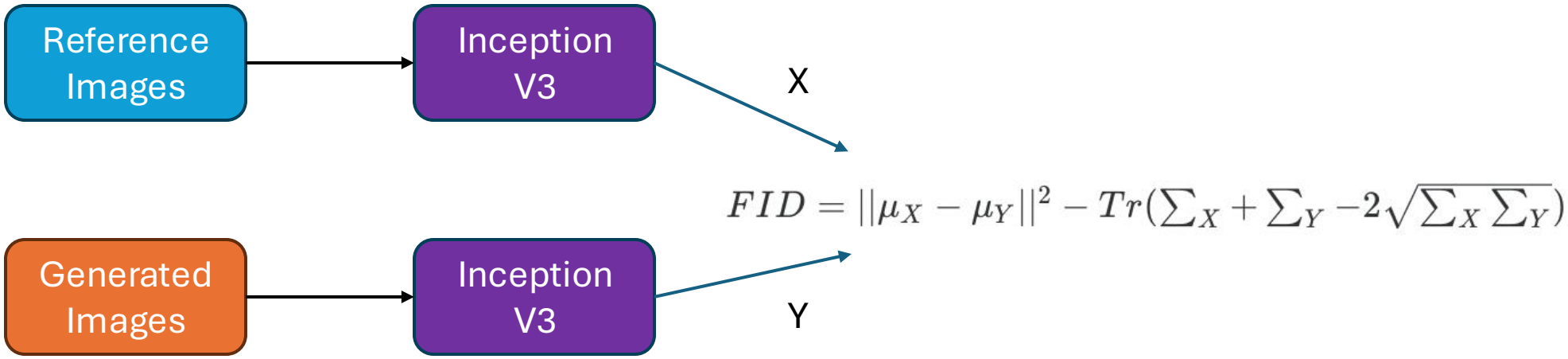


Contradicts reality!

- INT models outperform FP @ same bitwidth
- Quality improves as models get quantized to lower bitwidth

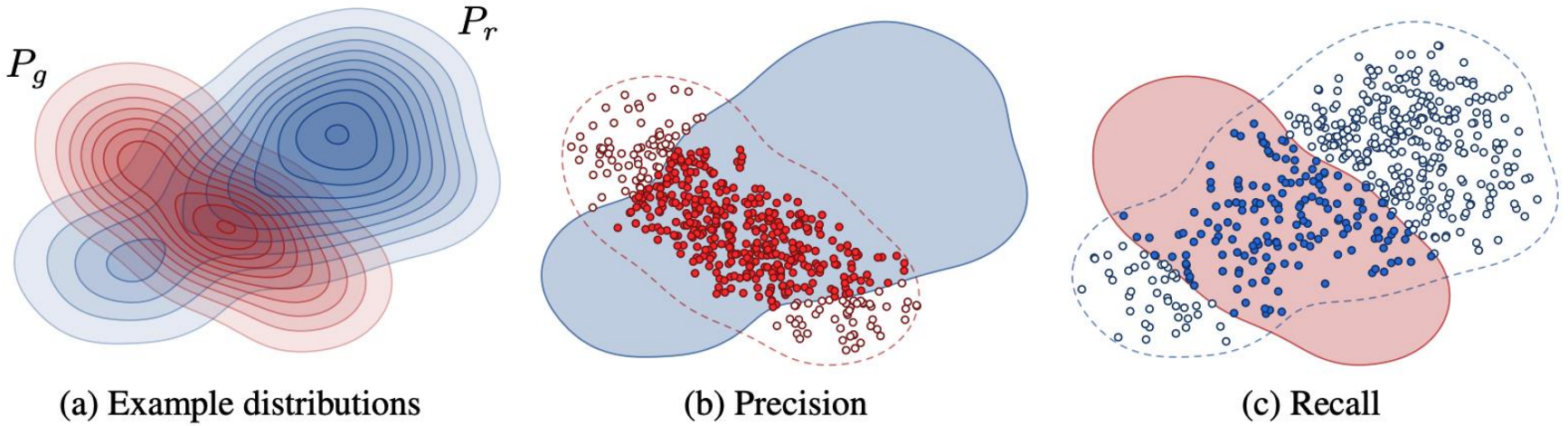
Evaluation Metrics

- FID, sFID



Evaluation Metrics

- Recall, Precision



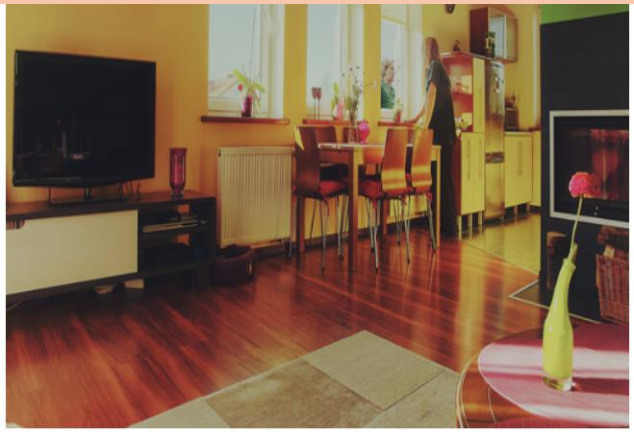
(a) Example distributions (b) Precision (c) Recall

https://arxiv.org/pdf/1904.06991

Discrepancy between quantitative eval and qualitative eval

- Standard methodology: use real-world collected images as reference
- Metrics measure similarity between reference images and generated images

Use the full precision model generated images as reference

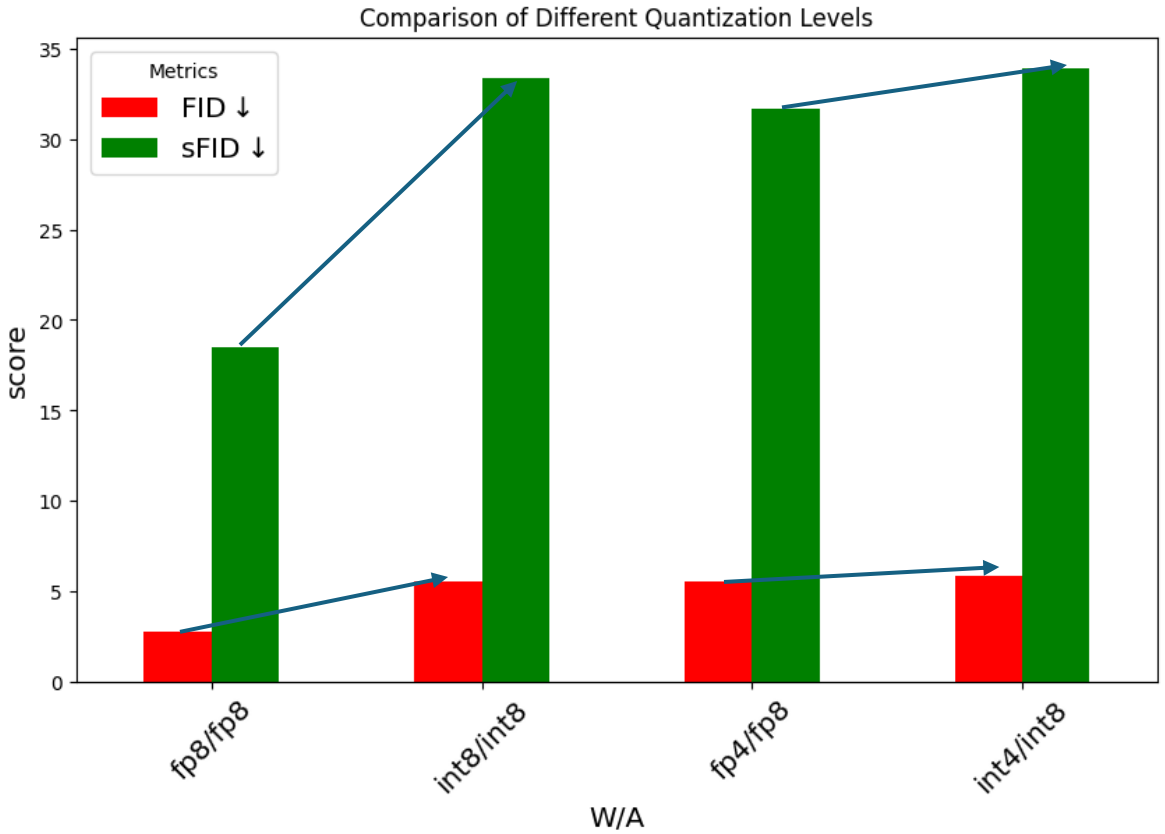


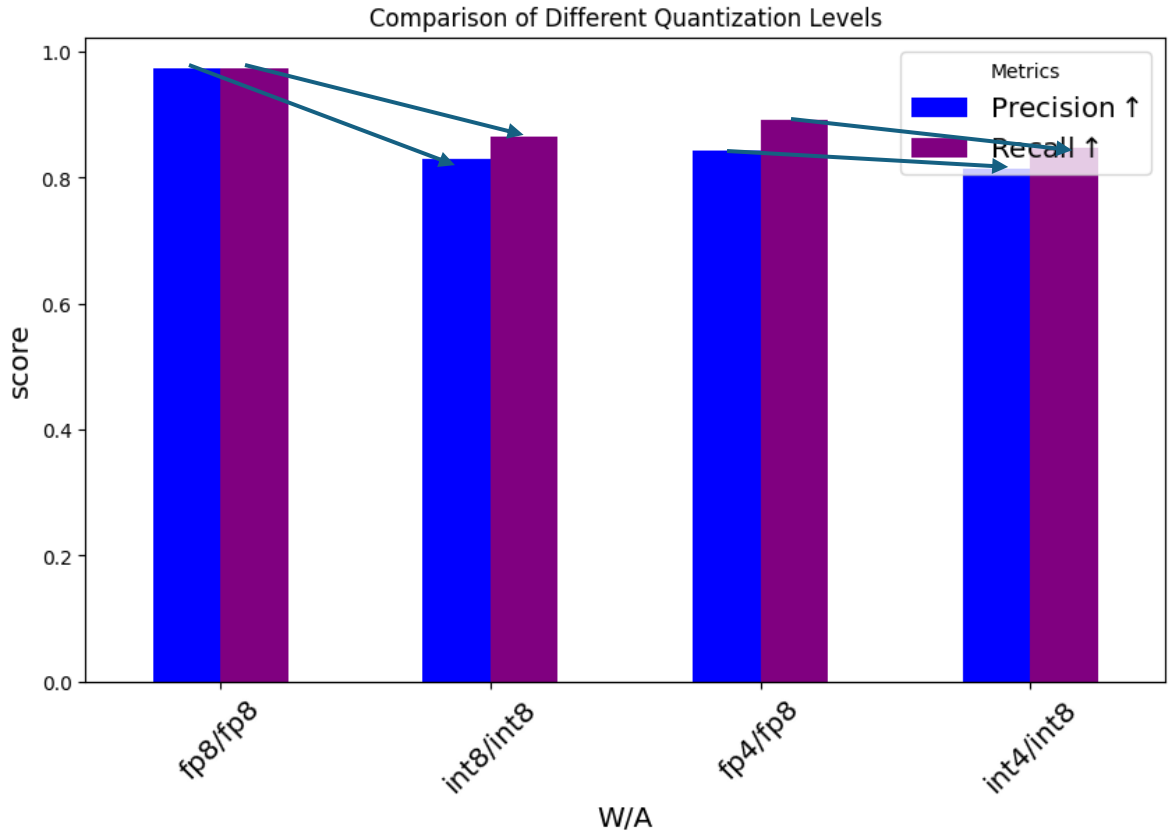
vs.

→ Quality

Text-to-Image Generation

Reference: Full-precision model generated images





- Metrics now represent reality
  - FP model outperform INT @ the same bitwidth
  - FP4/FP8 generates higher-quality images as INT8/INT8

Summary

- **Contributions**
  - Apply floating-point quantization on diffusion models(weights to FP4 and activations to FP8)
  - Adapt rounding learning for FP quantization
  - Improve evaluation methodology
- **Results**
  - FP8/FP8 VS. INT8/INT8 1.56x better
  - FP4/FP8 VS. INT4/INT8 1.10x better
  - FP4/FP8 better than INT8/INT8 in Stable Diffusion

Questions?