清華大學电子工程系

Department of Electronic Engineering, Tsinghua University

ICLR 2025

ViDiT-Q: Efficient and Accurate Quantization of Diffusion Transformers for Image and Video Generation

Tianchen Zhao[1,2], Tongcheng Fang[1,2], Enshu Liu[1], Rui Wan[1], Widyadewi Soedarmadji[1], Shiyao Li[12], Zinan Lin[3], Guohao Dai[24], Shengen Yan[2], Huazhong Yang[1], Xuefei Ning[1†‡], Yu Wang[1†],

[1]Tsinghua University, [2]Infinigence AI [3]Microsoft [4]Shanghai Jiao Tong University

[†]**Corresponding authors** [‡]**Project Advisor**

https://a-suozhang.xyz/viditq.github.io/

NICS EFC

Backgrounds: Visual Generation

- Recent advances of Visual Generation:

Midjourney

Midjourney generated
Image won art award

OpenAI SORA generates
Realistic videos

Backgrounds: Diffusion Model

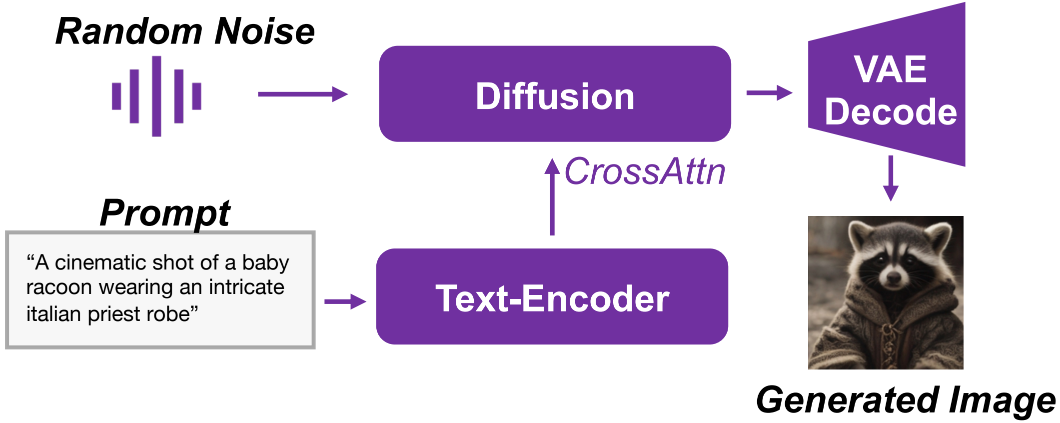
➢ Diffusion Model:

- Forward Process: Gradually add gaussian noise of different levels
- Backward Process: Gradually denoise the gaussian noise
- **Intuition:** the NN learns to predict the "noise" at each timestep.
  - *"Learning Data Distribution" -> "Denoising at different noise levels (timesteps)"*

[1] Ho, Jonathan et al. "Denoising Diffusion Probabilistic Models." *ArXiv* abs/2006.11239 (2020): n. pag.

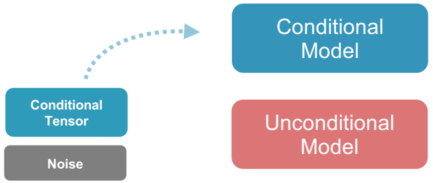
Backgrounds: Text-Conditioned Generation Flow

- **Task:** Text-Conditioned Generation
- **Components:**
  - **Diffusion Network:** U-Net / Diffusion Transformer
  - **Text Encoder:** CLIP / T5 / Language Model (ChatGLM)
  - **VAE Decoder:** CNN-based (8x Upsample)



- **CFG (Classifier-free Guidance)**
  - Along Batch-dimension
  - Inference **2 Times to generate 1 image**, with and without Inference

$$\tilde{\epsilon}_\theta(x_t, t, y) = \underbrace{(1+w)\,\epsilon_\theta(x_t, t, y)}_{\text{条件模型}} - \underbrace{w\,\epsilon_\theta(x_t, t)}_{\text{非条件模型}}$$



Classifier-free Guidance

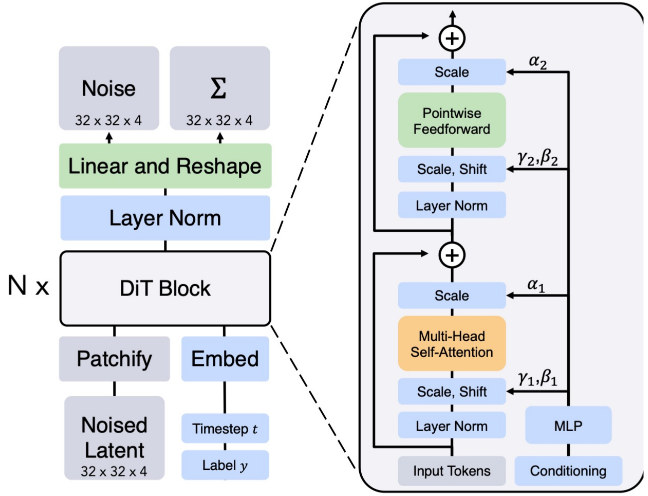
Backgrounds: Diffusion Transformer

Model Architecture:

- **Transformer Blocks**
- **AdaLN Modulation:** Use Linear layer to generate affine transform parameters for x
  - to incorporate the **timestep and condition** signal ($C = TimeEmb + Cond$)

```python
class DiTBlock(nn.Module):
    """
    A DiT block with adaptive layer norm zero (adaLN-Zero) conditioning.
    """
    def __init__(self, hidden_size, num_heads, mlp_ratio=4.0, **block_kwargs):
        super().__init__()
        self.norm1 = nn.LayerNorm(hidden_size, elementwise_affine=False, eps=1e-6)
        self.attn = Attention(hidden_size, num_heads=num_heads, qkv_bias=True, **block_kwargs)
        self.norm2 = nn.LayerNorm(hidden_size, elementwise_affine=False, eps=1e-6)
        mlp_hidden_dim = int(hidden_size * mlp_ratio)
        approx_gelu = lambda: nn.GELU(approximate="tanh")
        self.mlp = Mlp(in_features=hidden_size, hidden_features=mlp_hidden_dim, act_layer=approx_gelu, drop=0)
        self.adaLN_modulation = nn.Sequential(
            nn.SiLU(),
            nn.Linear(hidden_size, 6 * hidden_size, bias=True)
        )

    def forward(self, x, c):
        shift_msa, scale_msa, gate_msa, shift_mlp, scale_mlp, gate_mlp = self.adaLN_modulation(c).chunk(6, dim=1)
        x = x + gate_msa.unsqueeze(1) * self.attn(modulate(self.norm1(x), shift_msa, scale_msa))
        x = x + gate_mlp.unsqueeze(1) * self.mlp(modulate(self.norm2(x), shift_mlp, scale_mlp))
        return x
```



Latent Diffusion Transformer

DiT Block with adaLN-Zero

Backgrounds: Efficiency Problem

➢ The Diffusion Generative Model faces **severe "Efficiency Challenge"**

Latency Challenge:

Cannot Satisfy

SDXL (50 steps) generate 1024x1024 image on RTX3090: **30** s

Image Editing
Needs **Fast (<1s)** Feedback

OPEN SORA

OpenSORA (100 steps) generate 2s (512x512x16 Frames) image on RTX3090: **1-2 min**

Content Creation
Too Long **Waiting Time**

Memory Challenge:

Cannot Fit In

SDXL model
9.7GB GPU Memory

Desktop GPU: RTX4070
8GB GPU Memory

Cannot Fit In

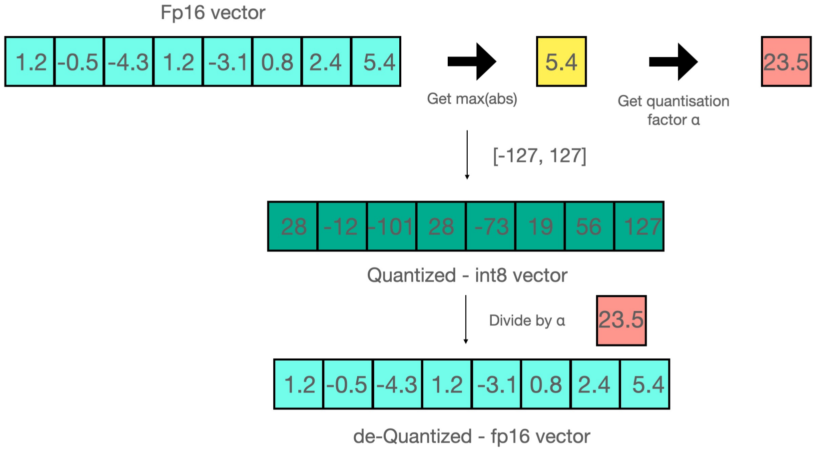
OPEN SORA

OpenSORA model
~12 GB GPU Memory

Backgrounds: Model Quantization

➢ The **model quantization** is an effective technique for reducing memory cost

- Reduce the Data **bit-width**
  - Reduce memory: could store 4x more param size (Compared with FP32)
  - Reduce computational complexity: more computing power for low bit-width operands
- A Few **Concepts**:
  - Quant/DeQuant Scheme
  - Quant Params:
    - Scale & ZeroPoint
    - **Granularity (Per-group)**
  - Static/Dynamic Quant



https://huggingface.co/blog/merve/quantization

Backgrounds: Model Quantization

- The **model quantization** is an effective technique for reducing memory cost
  - Quantization Process:

$$x_{\text{int}} = Q(x; s, z, b) = \operatorname{clamp}\left(\left\lfloor \frac{x}{s} \right\rceil + z, 0, 2^b - 1\right).$$

$$s = (\max(x) - \min(x))/(2^b - 1)$$

  - Objective:

$$\min \mathcal{L}_{\text{task}}(f_{FP}, f_q) \quad \Rightarrow \quad \min_{W_q, X_q} \sum_{l}^{L} \left(\|W^{(l)} - Q(W^{(l)})\|_2^2 + \|X^{(l)} - Q(X^{(l)})\|_2^2 \right),$$

Minimize Prediction between FP and Quantized model.

Minimize Quantization Error For Each Layer.

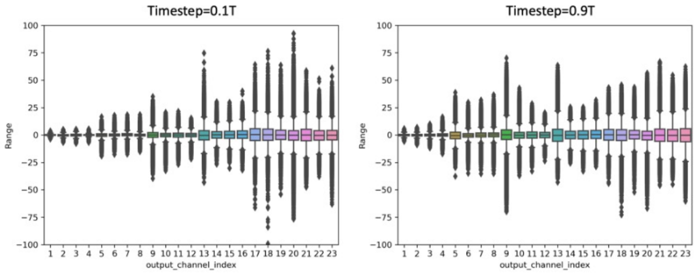
https://huggingface.co/blog/merve/quantization

Goals & Novelty

- Apply **Quantization** to Diffusion Transformers for Video & Image Generation.
  - What's **NEW**?
    - Pioneer in **Diffusion Transformer** Quantization
      - -> Existing Diffusion Quantization focuses on SD-like **U-Net (CNN) based** Model
      - -> Existing Transformer Quantization (LLM) does not include **unique "timestep"**
    - Pioneer in Quantization for **Video Generation Task**
      - -> Video Generation have unique challenges

	Granularity	Scheme
CNN	Per-tensor	Static
TR	Per-token	Dynamic

Transformer's Unique Challenge (Compared with CNNs): Dynamic Act Quantize & Per-token Quantization



Diffusion's Unique Challenge: Varying Activation Across Timesteps

Goals & Novelty

- Existing Quantization Methods faces challenges when quantizing DiTs

FP16

ViDiT-Q W8A8

Baseline W8A8

Preliminary Analysis: Quantization Error

- The **model quantization** is an effective technique for reducing memory cost
  - Quantization Process:

$$x_{\text{int}} = Q(x; s, z, b) = \text{clamp}\left(\left\lfloor \frac{x}{s} \right\rceil + z, 0, 2^b - 1\right).$$

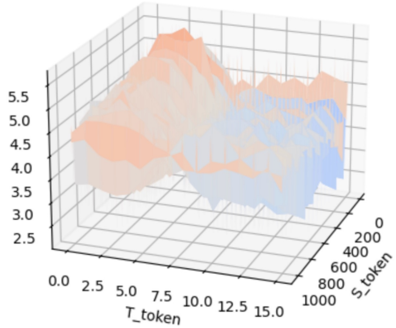
$$s = (\max(x) - \min(x))/(2^b - 1)$$

  - Quantization Error Analysis:
    - **Clipping Error:** When using Minmax Scaling, is **0**
    - **Rounding Error:** Within range $[-\frac{s}{2}, \frac{s}{2}]$
  - The Major Source of Quantization Failure: large **data variation**, some **outliers** causes **large *s***, not suitable for most elements. Measured by **"Incoherence":** $\frac{Max(X)}{Avg(X)}$ $X = [x_i, \dots, x_g]$
  - Adopting finer granularity (smaller group size **g**) reduces the incoherence.

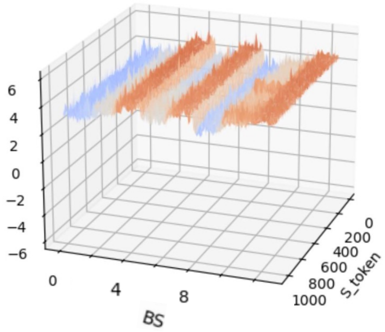
Findings: Unique Challenges for DiTs

➢ We conclude the unique challenges for DiT Quantization

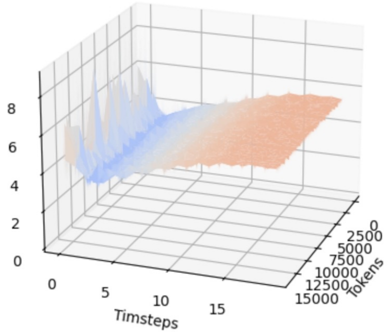
- **Large Data Variation** Across Different Dimensions:



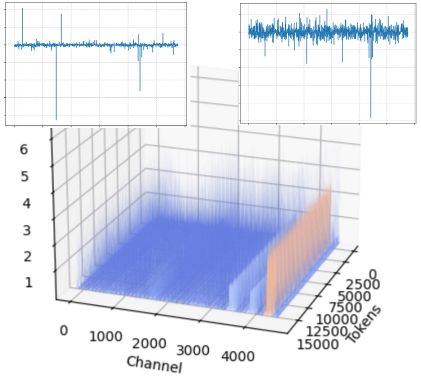
Token-wise Variation



CFG-wise Variation



Timestep-wise Variation



Time-varying Channel-wise Variation

Findings: Unique Challenges for Video Gen

- Video Generation Quality should be evaluated from multiple perspectives.

$$\min \mathcal{L}_{\text{task}}(f_{FP}, f_q) \quad \Rightarrow \quad \min_{W_q, X_q} \sum_{l}^{L} \left(\|W^{(l)} - Q(W^{(l)})\|_2^2 + \|X^{(l)} - Q(X^{(l)})\|_2^2 \right),$$

The MSE-based proxy task may not be enough

Text Alignment

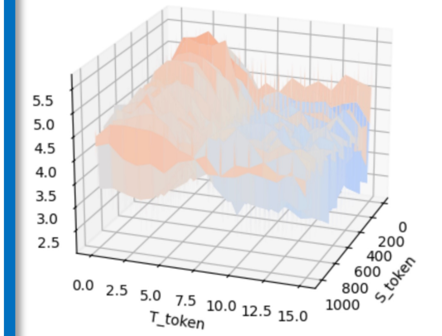
Visual Quality (Fidelity)

Time Consistency

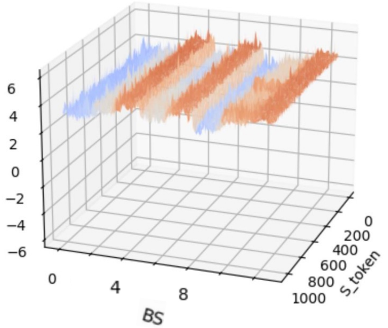
Method 1: Fine-grained & Dynamic Quantization

➢ Why Existing Methods Fails?

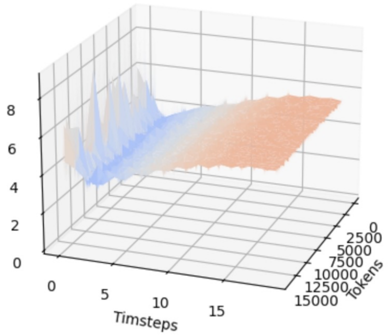
- Current CNN-target Quantization Scheme:
  - **Tensor-wise** Activation Quant Scheme



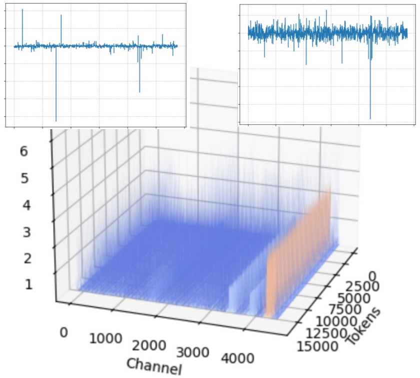
Token-wise Variation



CFG-wise Variation



Timestep-wise Variation

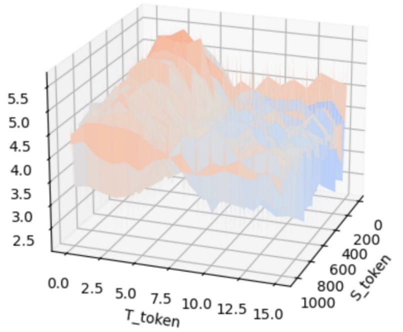


Time-varying Channel-wise Variation

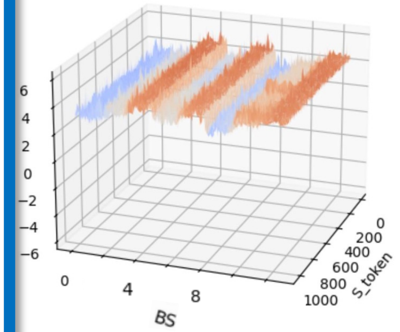
Method 1: Fine-grained & Dynamic Quantization

- Why Existing Methods Fails?
  - Current CNN-target Quantization Scheme:
    - **Static** Activation Quant Scheme
    - Timestep-wise Calibration & Adjustment for Quant Params

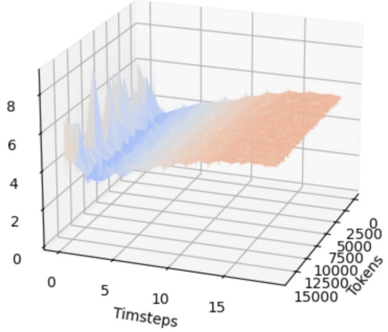
Dynamic Quant Intrinsically Solves This



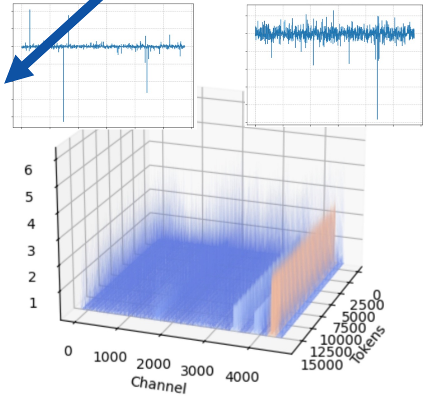
Token-wise Variation



CFG-wise Variation



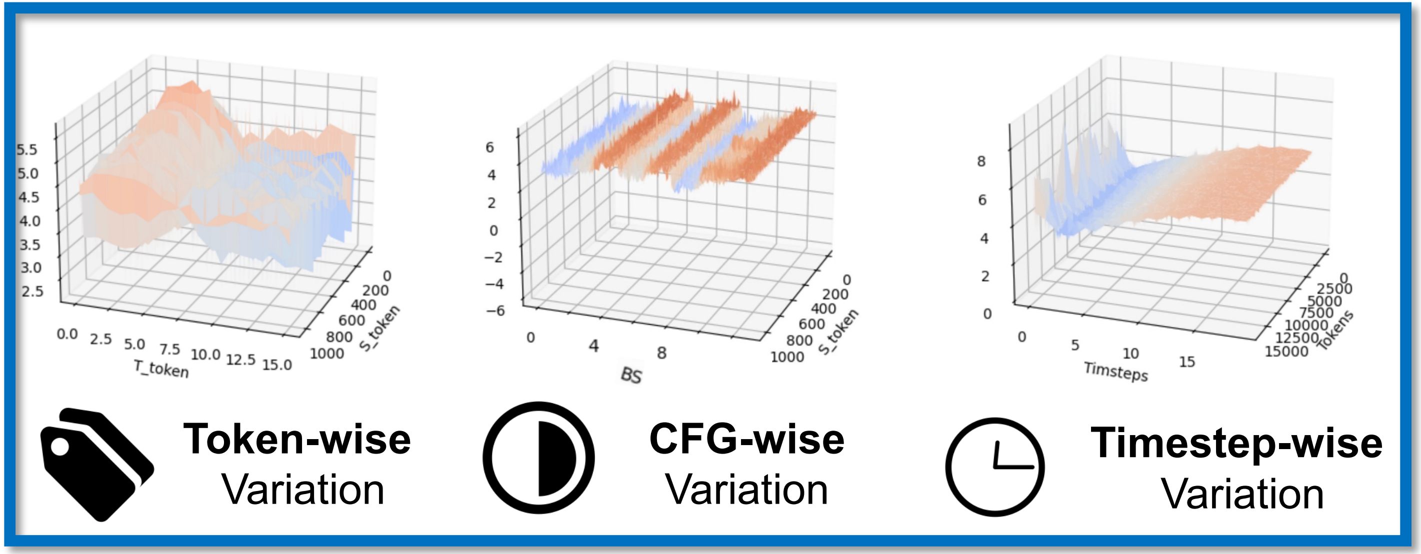
Timestep-wise Variation

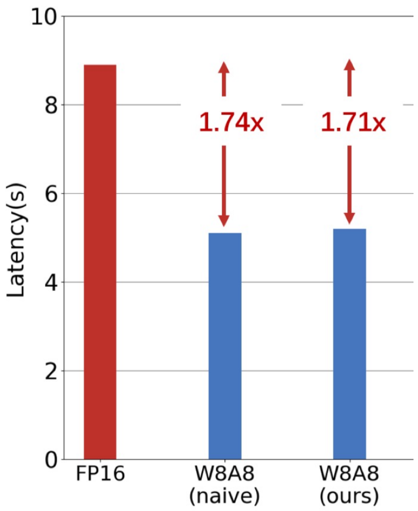


Time-varying Channel-wise Variation

Method 1: Fine-grained & Dynamic Quantization

- Solution: Adopting Fine-grained and Dynamic Activation Quant
  - (Which is the Standard Practice in LLM Quantization)
  - We highlight its importance and prove that it has **Negligible Overhead** during CUDA implementation

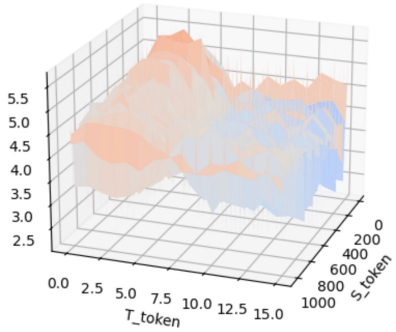




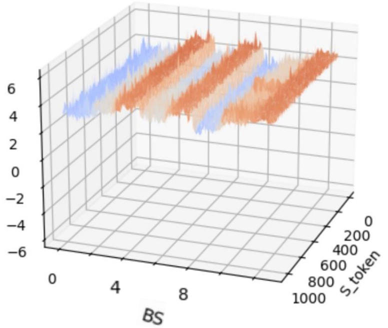
Method 2: Static-Dynamic Channel Balancing

➢ The Remaining Challenge: **Channel Imbalance**

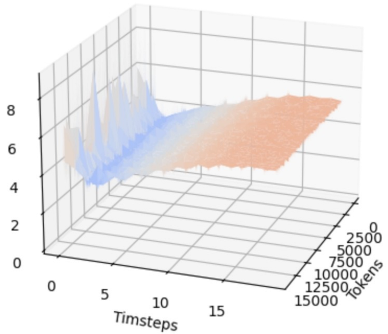
- By adopting fine-grained per-token quantization, the group consists of only [C] elements, However, variation still exists across channels.



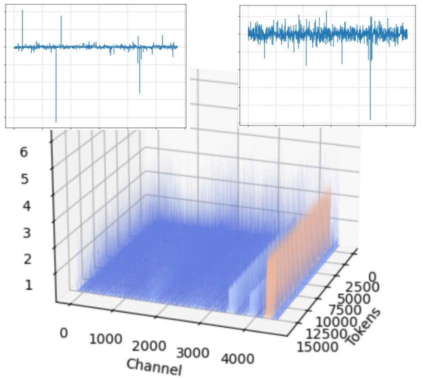
Token-wise Variation



CFG-wise Variation



Timestep-wise Variation



Time-varying Channel-wise Variation

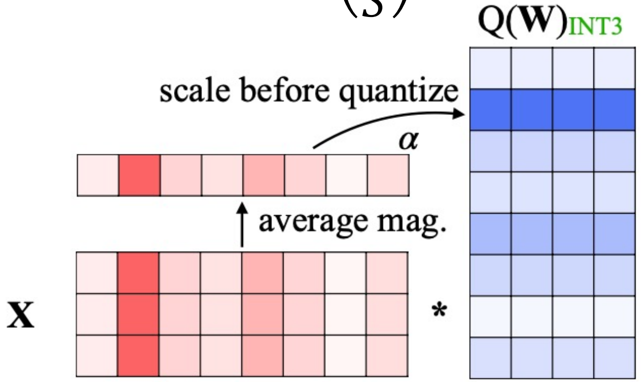
Method 2: Static-Dynamic Channel Balancing

➢ The Remaining Challenge: **Channel Imbalance**

- It's well studied in LLM Quantization, two schemes

Scaling-based

$$Y = X \cdot W^T = \left(\frac{X}{S}\right) * (W * S)$$



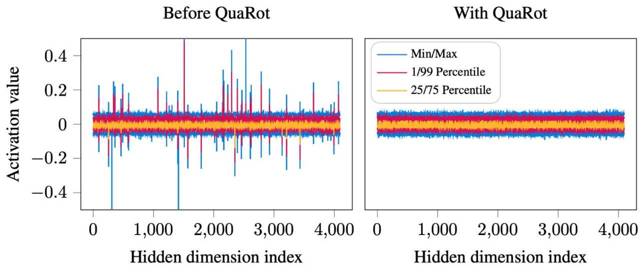
(c) Scale the weights before quantization (**PPL** 13.0)

AWQ [MLSYS 2024]

Rotation-based

$$Y = X \cdot W^T = (X \cdot H)(W \cdot H)^T$$

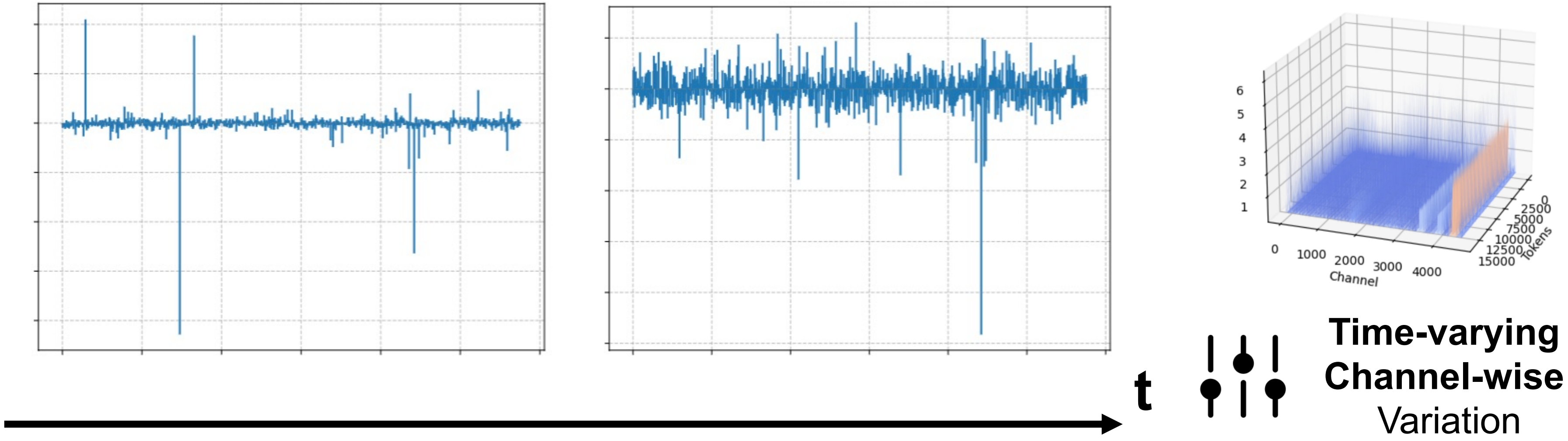
Orthogonal Matrix $H \cdot H^T = I$



Quarot [NeurIPS 2024]

Method 2: Static-Dynamic Channel Balancing

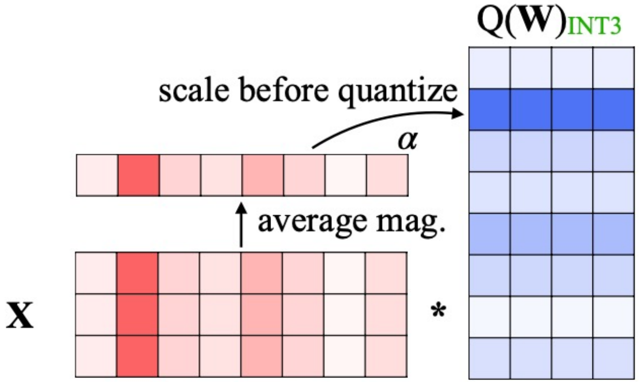
- Unique Challenge: **Time-Varying Channel Imbalance**
  - By adopting fine-grained per-token quantization, the group consists of only [C] elements, However, variation still exists across channels.



Method 2: Static-Dynamic Channel Balancing

➢ Existing Method’s Challenge for **Timestep-wise Channel Imbalance**

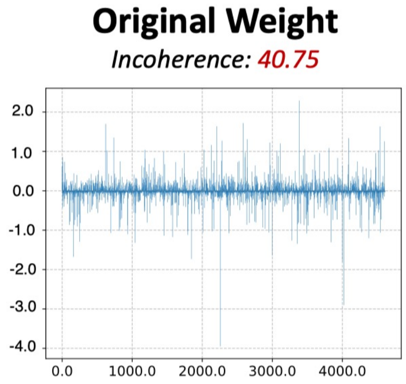
Scaling-based

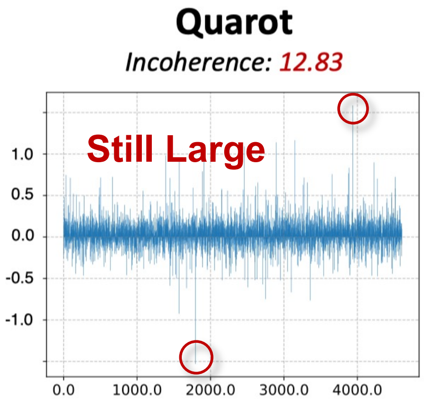


(c) Scale the weights before quantization (**PPL 13.0**)

- Should use different α for different timesteps
- Need to store **Multiple Weights**

Rotation-based





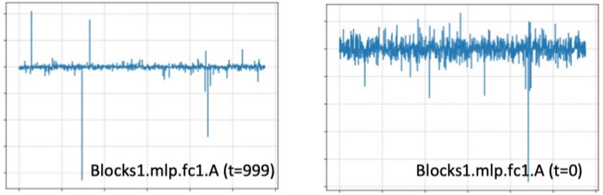
- **Could not address very large and distributed Outliers**

Method 2: Static-Dynamic Channel Balancing

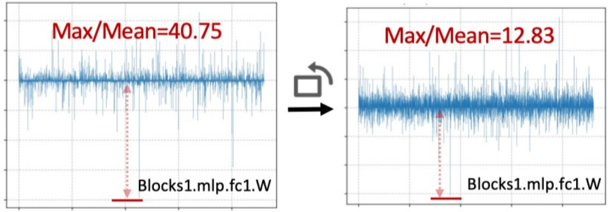
- ➢Exploring **where** Timestep-wise Channel Imbalance **comes from**

Existing Channel Balance's Challenges

Very Significant Imbalance | Relatively smaller imbalance

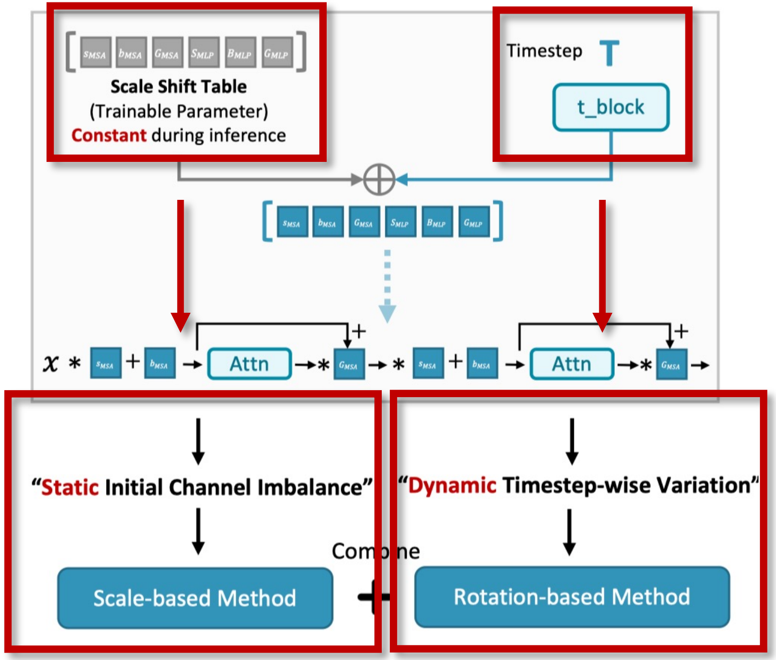


Scaling-based: Single α could not fit time-varying distribution



Rotation-based: Outlier still exists after rotation, 12.8x is hard for 4-bit(16 levels)

Static-Dynamic Channel Balance (Sec. 4.2)



Method 3: Metric Decoupled Mixed Precison

➤ Video Generation Quality should be evaluated from multiple perspectives. How to **preserve Quantization's effect on these perspectives?**

Method 3: Metric Decoupled Mixed Precison

- Motivation: -> Mixed Precision
  - Quantization under lower bitwidth (W4) is bottlenecked by some layer
  - Quantization for different layer types have unique correlation with evaluation

Baseline W4A8: "Blank Frames"

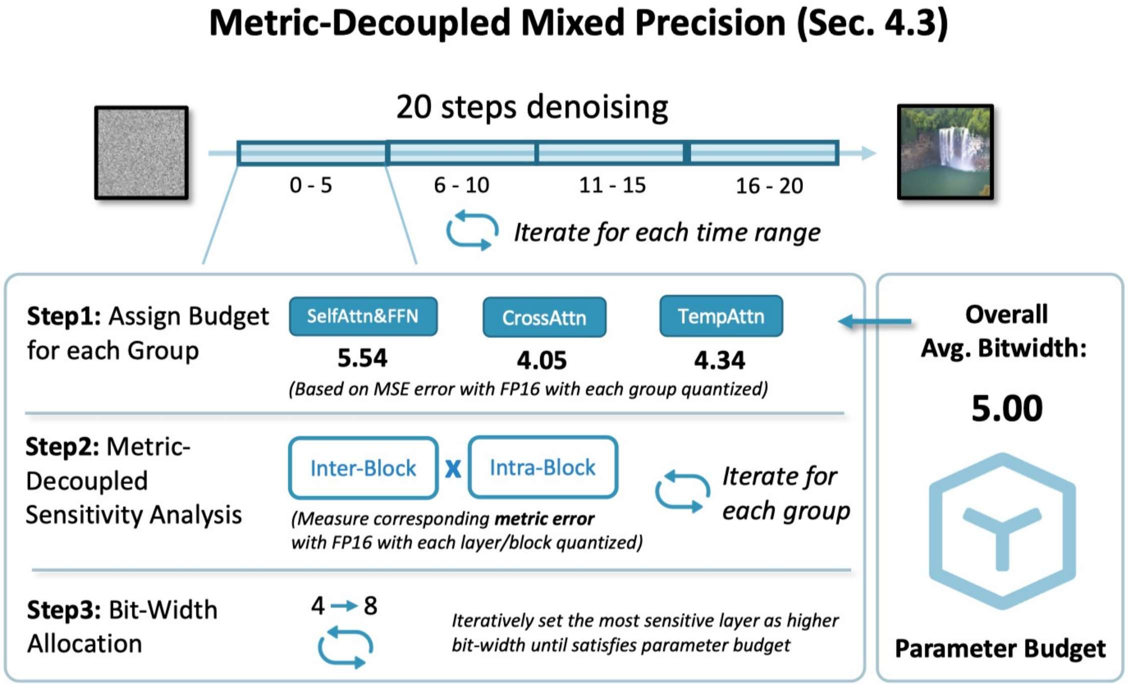
Failure under W4

Layers have **diverse quantization sensitivity**

Quantization effects are **highly correlated with layer types**

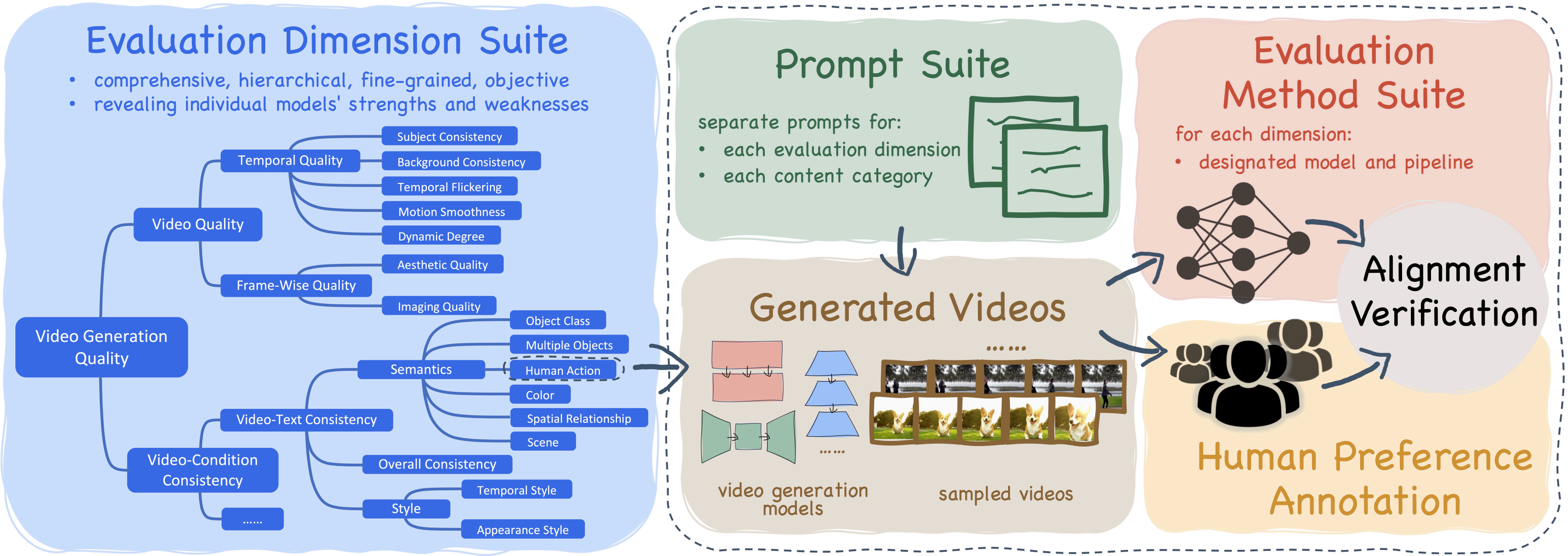
Method 3: Metric Decoupled Mixed Precison

- How to consider different aspects for Mixed Precision?
- **Metric-decoupled** Mixed Precision



Main Results: T2V VBench

- Vbench: Comprehensive Evaluation Suite from various perspectives

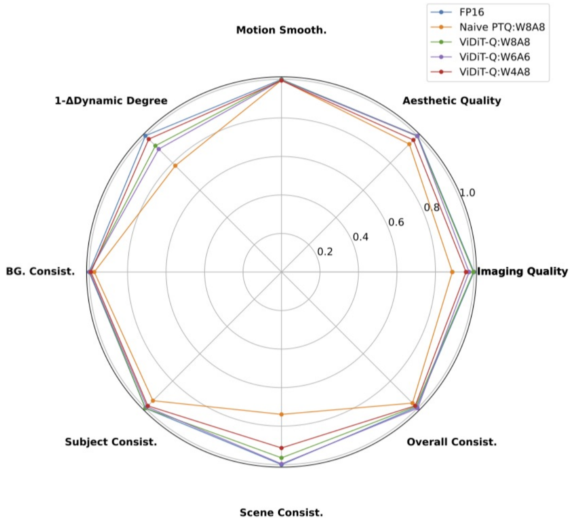


Main Results: T2V VBench

➢ Vbench: Comprehensive Evaluation Suite from various perspectives

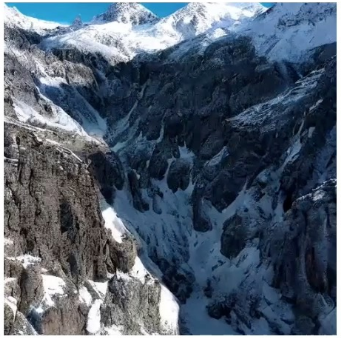
Table 1: **Performance of ViDiT-Q text-to-video generation on VBench evaluation benchmark suite.** The bit-width "16" represents FP16 without quantization. We omit some baselines that fails to produce readable content under W4A8. The mixed precision are applied for ViDiT-Q W4A8.

Method	**Bit-width** (W/A)	Imaging Quality	Aesthetic Quality	Motion Smooth.	Dynamic Degree	BG. Consist.	Subject Consist.	Scene Consist.	Overall Consist.
-	16/16	63.68	57.12	96.28	56.94	96.13	90.28	39.61	26.21
Q-Diffusion	8/8	60.38	55.15	94.44	68.05	94.17	87.74	36.62	25.66
Q-DiT	8/8	60.35	55.80	93.64	68.05	94.70	86.94	32.34	26.09
PTQ4DiT	8/8	56.88	55.53	95.89	63.88	96.02	91.26	34.52	25.32
SmoothQuant	8/8	62.22	55.90	95.96	68.05	94.17	87.71	36.66	25.66
Quarot	8/8	60.14	53.21	94.98	66.21	95.03	85.35	35.65	25.43
ViDiT-Q	8/8	63.48	56.95	96.14	61.11	95.84	90.24	38.22	26.06
Q-DiT	4/8	23.30	29.61	97.89	4.166	97.02	91.51	0.00	4.985
PTQ4DiT	4/8	37.97	31.15	92.56	9.722	98.18	93.59	3.561	11.46
SmoothQuant	4/8	46.98	44.38	94.59	21.67	94.36	82.79	26.41	18.25
Quarot	4/8	44.25	43.78	92.57	66.21	94.25	84.55	28.43	18.43
ViDiT-Q	4/8	61.07	55.37	95.69	58.33	95.23	88.72	36.19	25.94

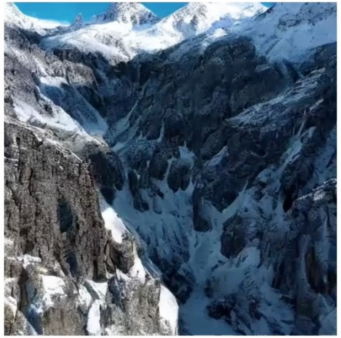


Main Results: T2V VBench

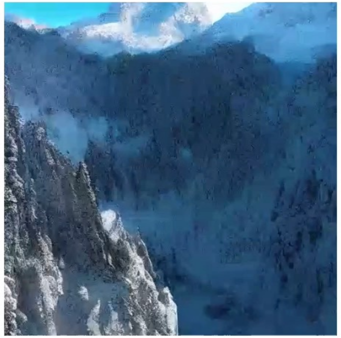
- Vbench: Comprehensive Evaluation Suite from various perspectives



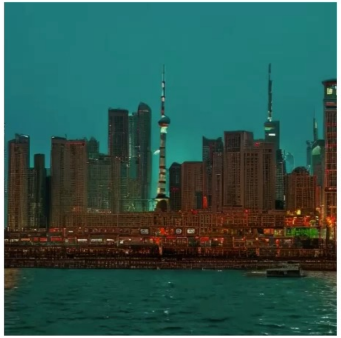
FP16



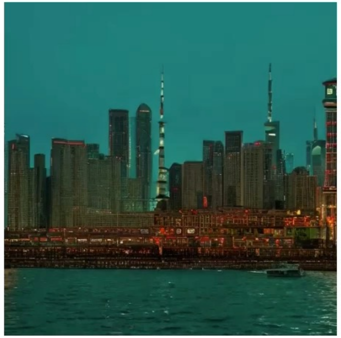
ViDiT-Q W8A8



Baseline W8A8: "Jitter and Color Shift"



FP16



ViDiT-Q W8A8

Baseline W8A8: "Content Changes"

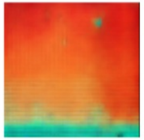
Main Results: T2V Metrics

- Metrics:
  - CLIPSIM/CLIP-Temp | VQA | FlowScore

Method	Bit-width (W/A)	CLIPSIM	CLIP-Temp	VQA-Aesthetic	VQA-Technical	Δ Flow Score. (↓)
-	16/16	0.1797	0.9988	63.40	50.46	-
Q-Diffusion	8/8	0.1781	0.9987	51.68	38.27	0.328
Q-DiT	8/8	0.1788	0.9977	61.03	34.97	0.473
PTQ4DiT	8/8	0.1836	0.9991	54.56	53.33	0.440
SmoothQuant	8/8	0.1951	0.9986	59.78	51.53	0.331
Quarot	8/8	0.1949	0.9976	58.73	52.28	0.215
ViDiT-Q	8/8	0.1950	0.9991	60.70	54.64	0.089
Q-DiT	6/6	0.1710	0.9943	11.04	1.869	41.10
PTQ4DiT	6/6	0.1799	0.9976	59.97	43.89	0.997
SmoothQuant	6/6	0.1807	0.9985	56.45	48.21	29.26
Quarot	6/6	0.1820	0.9975	61.47	53.06	0.146
ViDiT-Q	6/6	0.1791	0.9984	64.45	51.58	0.625
Q-DiT	4/8	0.1687	0.9833	0.007	0.018	3.013
PTQ4DiT	4/8	0.1735	0.9973	2.210	0.318	0.108
SmoothQuant	4/8	0.1832	0.9983	31.96	22.85	0.415
Quarot	4/8	0.1817	0.9965	47.36	33.13	0.326
ViDiT-Q	4/8	0.1809	0.9989	60.62	49.38	0.153

FP16


ViDiT-Q


Q-DiT


PTQ4DiT


Main Results: T2I Metrics

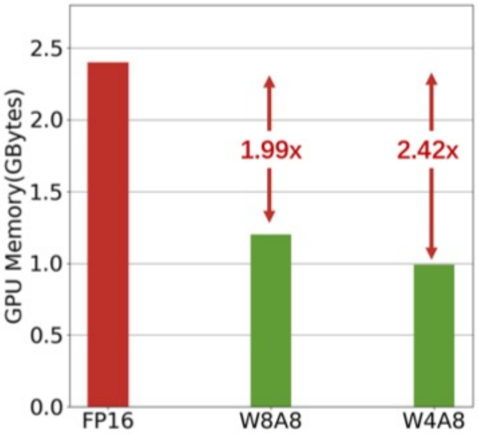
- Metrics:
  - FID | CLIP-score | ImageReward

Method	Bit-width (W/A)	FID(↓)	CLIP(↑)	IR(↑)
-	16/16	73.34	0.258	0.901
Q-Diffusion	8/8	96.54	0.239	0.186
	4/8	91.95	0.228	-0.224
Q-DiT	8/8	73.60	0.256	0.854
	4/8	475.8	0.127	-2.277
PTQ4DiT	8/8	127.9	0.217	-1.216
	4/8	171.9	0.177	-2.064
ViDiT-Q	8/8	75.61	0.259	0.917
	4/8	74.33	0.257	0.887

Main Results: Hardware Efficiency

- We implement the Efficient CUDA Kernel for actual hardware resource measurement on Nvidia A100.
  - (Fused Kernel Implemented)

Bit-width (W/A)	Memory Opt.	Latency Opt.
16/16	1.00×	1.00×
8/8 (naive)	1.99×	1.74×
8/8 (ours)	1.99×	1.71×
4/8 (ours)	2.42×	1.38×



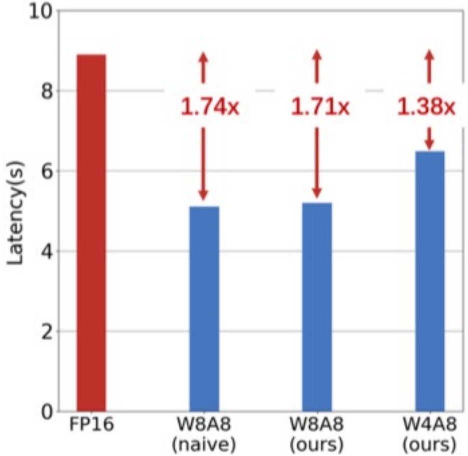


Figure 7: **The illustration of ViDiT-Q's hardware resource savings.** The table and figures present memory savings and end-to-end latency speedup of ViDiT-Q and naive quantization scheme.

Analysis: Ablation Studies

➢ Ablation Studies

Methods			Bit-width	CLIPSIM	CLIP-Temp	VQA-	VQA-	Δ Flow
Quant Params	Channel Balance	Mixed Precision	(W/A)			Aesthetic	Technical	Score.
-	-	-	16/16	0.180	0.998	64.198	51.904	-
Static & Tensor-wise	-	-	4/8	0.201	0.997	0.178	0.086	0.603
Dynamic & Token-wise	-	-	4/8	0.196	0.998	32.217	10.994	0.109
Dynamic & Token-wise	Scaling-based	-	4/8	0.191	0.999	31.963	22.847	0.415
Dynamic & Token-wise	Rotation-based	-	4/8	0.181	0.999	47.356	33.128	0.326
Dynamic & Token-wise	Static-Dynamic	-	4/8	0.181	0.999	60.216	42.257	0.151
Dynamic & Token-wise	Static-Dynamic	MSE-based	4/8	0.179	0.999	53.335	38.729	0.258
Dynamic & Token-wise	Static-Dynamic	Metric Decoupled	4/8	0.199	0.999	60.616	49.383	0.334

Generated Videos Example of Ablation Studies: **STDiT W4A8**

"A serene underwater scene featuring a sea turtle swimming through a coral reef. The turtle, with its greenish-brown shell, is the main focus of the video, swimming gracefully towards the right side of the frame. The coral reef, teeming with life, is visible in the background, providing a vibrant and colorful backdrop to the turtle's journey. Several small fish, darting around the turtle, add a sense of movement and dynamism to the scene. The video is shot from a slightly elevated angle, providing a comprehensive view of the turtle's surroundings. The overall style of the video is calm and peaceful, capturing the beauty and tranquility of the underwater world."

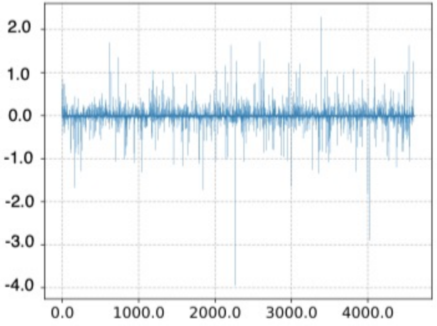
Baseline | + Dynamic Quantization | + Channel Balancing | + Static-Dynamic Quantization | + Mixed Precision | **FP16**

Analysis: Ablation Studies

➢ Visualization of Channel Balancing

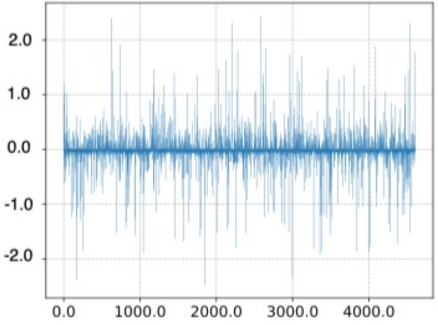
Original Weight

Incoherence: 40.75



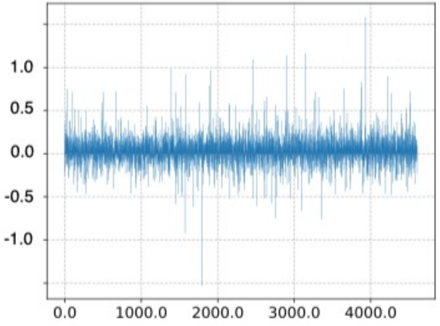
Static- Dynamic Channel Balanced (static scaling only)

Incoherence: 17.77



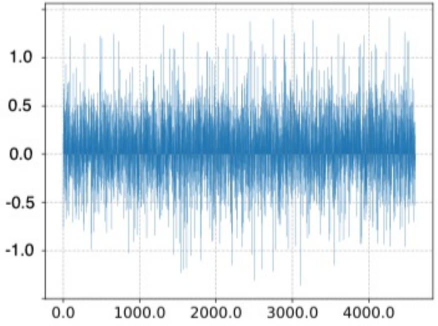
Quarot

Incoherence: 12.83



Static- Dynamic Channel Balanced

Incoherence: 5.02



Thank you!

Tianchen Zhao

suozhang1998@gmail.com

Project Page:
Open-sourced Code
& CUDA Kernels
(Update **Soon**)