Error Propagation Mechanisms and Compensation Strategies for Quantized Diffusion Models

ICML 2026 Oral

Authors: Songwei Liu*[1], Chao Zeng*[1], Chenqian Yan[1], Xurui Peng[1], Xing Wang[1], Fanmin Chen[1], Xing Mei[1]

Affiliation: [1]ByteDance Inc.

Contact: liusongwei.zju@bytedance.com

Catalog

Contact: 21831068@zju.edu.cn

Motivation & Contributions

The Problem

- Diffusion inference is expensive because denoising is iterativeantized.
- Quantized Diffusion Models (QDMs) achieve fast inference but suffer from severe quality degradation.
- Existing PTQ methods mainly reduce single-step error.

Our Solution

TCEC explicitly models this accumulated error and corrects it online with very small overhead.

Our Contributions

Identify the key bottleneck in low-precision DMs

Cumulative error is the main reason for performance degradation.

Build a principled and efficient correction framework

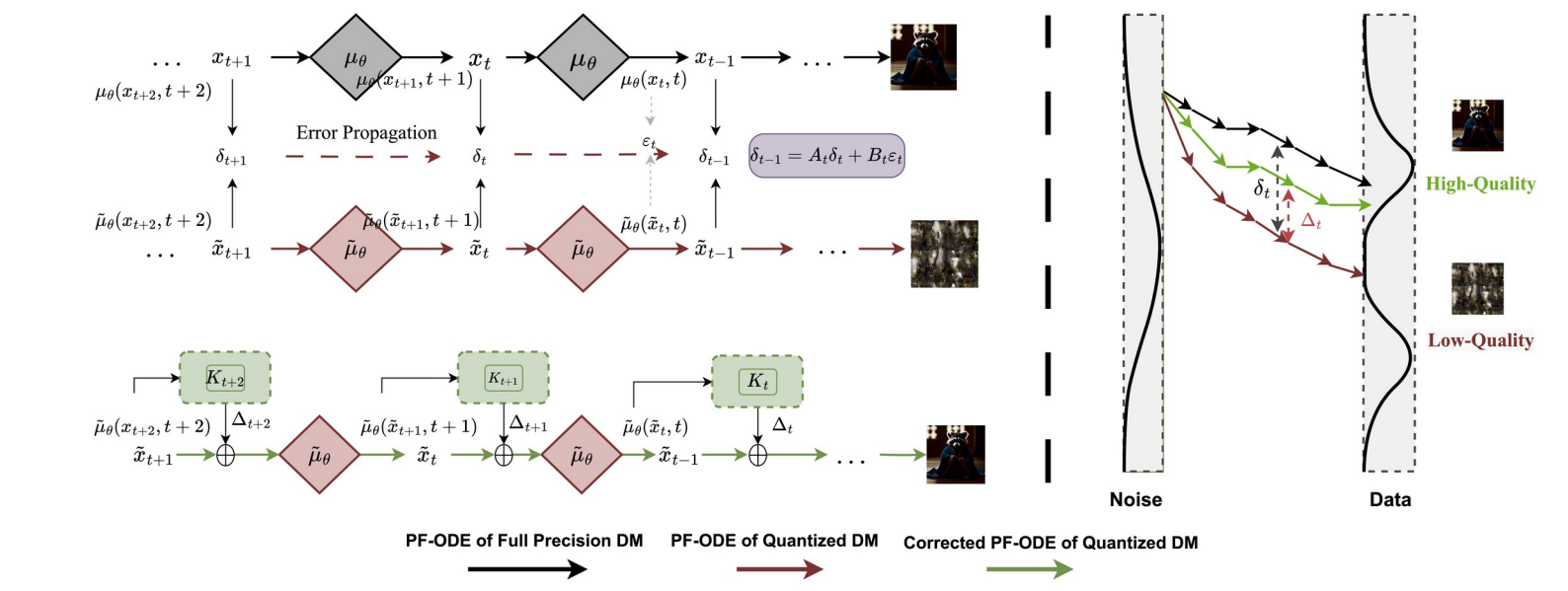
Formulate cumulative error through per-step error, accumulation, and propagation, derive a closed-form solution.

Achieve strong efficiency–quality trade-offs in practice

Significantly improves image quality and diversity while delivering 3.5× memory reduction and 3× inference speedup under low-precision settings..

Contact: 21831068@zju.edu.cn

Methodology (1/2) - The Problem



- **Full-precision trajectory follows the desired denoising path.**
- **Quantized trajectory drifts due to accumulated error.**
- **TCEC adds online correction to pull it back.**

Contact: 21831068@zju.edu.cn

Methodology (2/2) - Our Correction

Three Key Quantities

- **(ε_t)**
  The output distortion introduced by the quantized model at

- The deviation between the quantized latent state and the full-precision latent state after multiple denoising steps

- **Correction term (Δ_t)**
  The compensation injected online to reduce accumulated trajectory drift .

DDIM Sample

- **Full-presicion update**

$$x_{t-1} = \mathrm{DDIM}(x_t, \mu_\theta(x_t, t))$$

- **Quantized update**

$$\widetilde{x_{t-1}} = \mathrm{DDIM}(\tilde{x}_t, \widetilde{\mu_\theta}(\tilde{x}_t, t))$$

- **Quantized model output**

$$\widetilde{\mu_\theta}(\tilde{x}_t, t) = \mu_\theta(\tilde{x}_t, t)) + \varepsilon_t$$

- **State deviation**

$$\tilde{x}_t = x_t + \delta_t$$

Error Propagation Equation

$$\delta_{t-1} = A_t\delta_t + B_t\varepsilon_t$$

$A_t\delta_t$: propagated error from later timesteps.

$B_t\varepsilon_t$: newly injected quantization error at the current timestep.

Key idea: Quantization error does not stay local. It perturbs the current prediction, shifts the next latent input, and keeps propagating through the entire denoising trajectory.

Contact: 21831068@zju.edu.cn

Methodology (2/2) - Our Correction

From local error to cumulative drift

A small per-step quantization error changes the latent state, so the following denoising steps are evaluated at shifted inputs.

Why the next slide?
The exact closed form is accurate, but too expensive for fast quantized sampling.

1 Per-step error
The quantized model injects a local prediction error at timestep t.

→

2 State drift
The next latent input is no longer exactly on the full-precision trajectory.

→

3 Accumulation
The drift is transported and amplified across later reverse steps.

Propagation view: $\delta_{t-1} = A_t\delta_t + B_t\varepsilon_t$

Correction target: estimate δ_t and apply $\Delta_t = -\delta_t$

Contact: 21831068@zju.edu.cn

Methodology (2/2) - Our Correction

Directly computing the exact closed-form cumulative error involves:

- recursive matrix products
- Jacobian-related terms
- long-horizon accumulation from timestep

$$\delta_t = \sum_{k=t}^{T} \left(\prod_{j=t}^{k-1} A_j^{-1} \right) B_k \varepsilon_k$$

Accurate, but not practical for real-time inference.

Approximation 1

For a well-trained diffusion model, it is insensitive to local changes in the input, which implies that we can ignore the Jacobian term: $J_{x_t} \approx 0$.

$$\prod_{j=t}^{k-1} A_j^{-1} = \prod_{j=t}^{k-1} \frac{\sqrt{\alpha_j}}{\sqrt{\alpha_{j-1}}} = \frac{\sqrt{\alpha_t}}{\sqrt{\alpha_{t-1}}} \cdot \frac{\sqrt{\alpha_{t+1}}}{\sqrt{\alpha_t}} \cdots = \frac{\sqrt{\alpha_{k-1}}}{\sqrt{\alpha_{t-1}}}$$

$$\Delta_t = -\sum_{k=t}^{T} \left(\frac{\sqrt{\alpha_{k-1}}}{\sqrt{\alpha_{t-1}}} \right) B_k \varepsilon_k$$

Approximation 2

The correction term only takes into account the subsequent *m* steps. Since the denoising process unfolds in reverse, proceeding from *T* to *0*, at the *t-th* step, only the quantization noises at steps $t + m$, $t + m - 1$, ..., $t + 1$ are factored in. We refer to this as the temporal locality approximation.

$$\Delta_t \approx -\frac{1}{\sqrt{\alpha_{t-1}}} \sum_{k=t}^{\min(t+m,T)} \sqrt{\alpha_{k-1}} \mathbf{B}_k \varepsilon_k$$

Lightweight solution

Per-step error estimation

$$\varepsilon_t = \mathbf{K}_t \odot \tilde{\mu}_\theta(\tilde{x}_t, t)$$

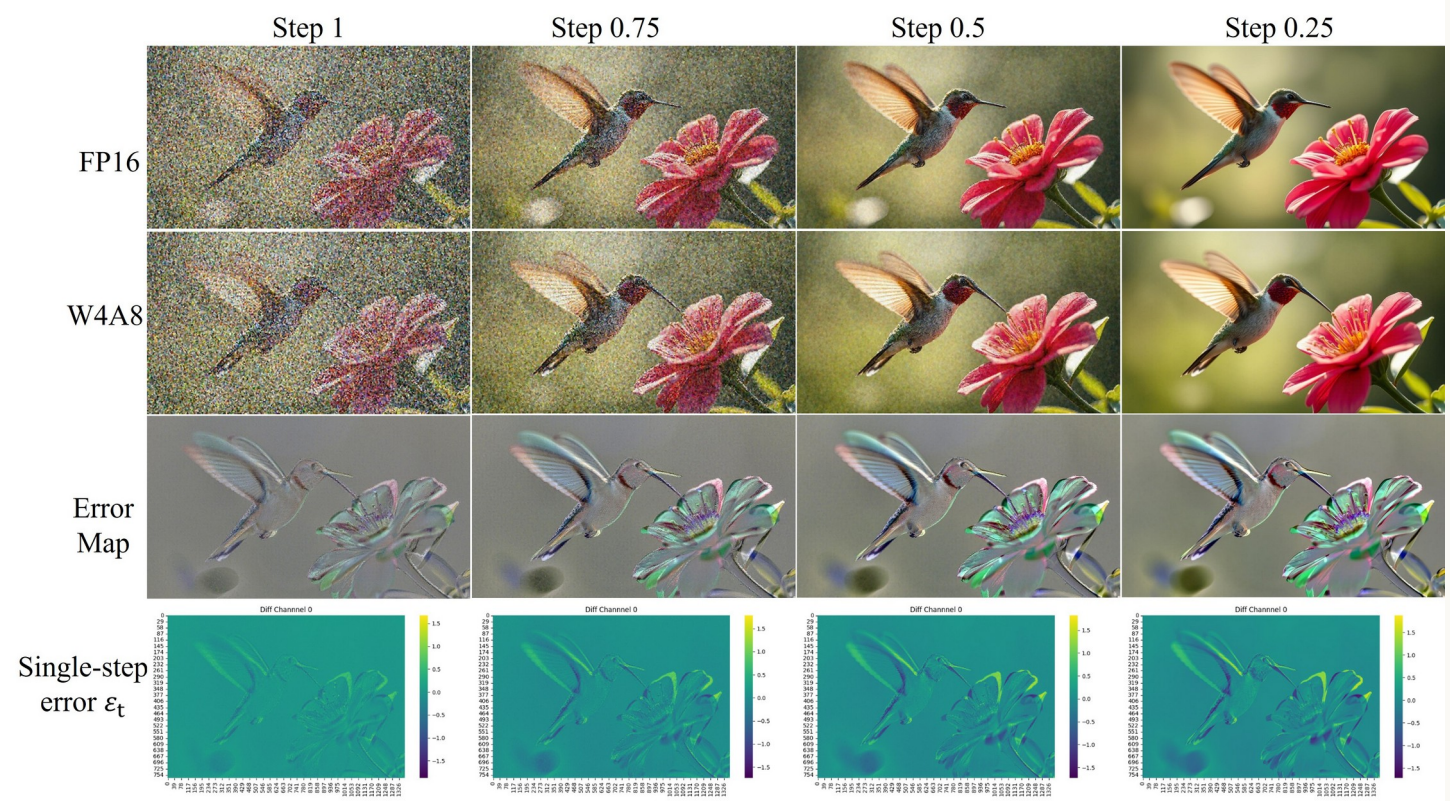
Final online correction

$$\Delta_t \approx -\frac{1}{\sqrt{\alpha_{t-1}}} \sum_{k=t}^{\min(t+1,T)} \sqrt{\alpha_{k-1}} \mathbf{B}_k \varepsilon_k$$

- only uses nearby timesteps.
- only requires cached channel-wise coefficients.
- adds negligible computation beyond the quantized backbone.

Contact: 21831068@zju.edu.cn

Empirical evidence



Errors accumulate across denoising steps

Errors correlate with generated structure

High-frequency regions are most affected

Visualization

Quantitative Results

Model	Precision	Method	MJHQ				sDCI			
			Quality		Similarity		Quality		Similarity	
			FID↓	IR↑	LPIPS↓	PSNR↑	FID↓	IR↑	LPIPS↓	PSNR↑
SDXL	FP16	-	16.6	0.729	-	-	22.5	0.573	-	-
	W8A8	TensorRT	20.2	0.591	0.247	22.0	25.4	0.453	0.265	21.7
	W8A8	SVDQuant	16.6	0.718	0.119	26.4	22.4	0.574	0.129	25.9
	W8A8	SVDQuant + TCEC	**16.0**	**0.728**	**0.092**	**27.3**	**22.0**	**0.580**	**0.103**	**26.7**
	W4A4	SVDQuant	20.6	0.601	0.288	21.0	26.3	0.477	0.307	20.7
	W4A4	SVDQuant + TCEC	**18.1**	**0.652**	**0.249**	**21.9**	**23.4**	**0.513**	**0.259**	**21.9**
SDXL-Turbo	FP16	-	24.3	0.845	-	-	24.7	0.705	-	-
	W8A8	MixDQ	**24.1**	0.834	0.147	21.7	25.0	0.690	0.157	21.6
	W8A8	SVDQuant	24.3	0.845	0.100	24.0	24.8	0.701	0.110	23.7
	W8A8	SVDQuant + TCEC	24.5	**0.849**	**0.083**	**24.9**	**23.9**	**0.720**	**0.098**	**24.5**
	W4A8	MixDQ	27.7	0.708	0.402	15.7	25.9	0.610	0.415	15.7
	W4A4	MixDQ	353	-2.26	0.685	11.0	373	-2.28	0.686	11.3
	W4A4	SVDQuant	24.6	0.816	0.262	18.1	26.0	0.671	0.272	18.0
	W4A4	SVDQuant + TCEC	**23.9**	**0.833**	**0.230**	**19.0**	**25.1**	**0.691**	**0.232**	**19.3**
PixArt-$\sum$	FP16	-	16.6	0.944	-	-	24.8	0.966	-	-
	W8A8	ViDiT-Q	**15.7**	0.944	0.137	22.5	23.5	**0.974**	0.163	20.4
	W8A8	SVDQuant	16.3	0.955	0.109	23.7	24.2	0.969	0.129	21.8
	W8A8	SVDQuant + TCEC	16.2	**0.964**	**0.098**	**24.5**	**23.4**	0.952	**0.118**	**22.6**
	W4A4	ViDiT-Q	412	-2.27	0.854	6.44	425	-2.28	0.838	6.70
	W4A4	SVDQuant	19.2	0.878	0.323	17.6	25.9	0.918	0.352	16.5
	W4A4	SVDQuant + TCEC	**18.1**	**0.903**	**0.285**	**18.3**	**25.3**	**0.934**	**0.304**	**16.9**

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
ViDiT-Q + TCEC	8/8	**65.56**	**57.12**	**96.27**	61.09	**96.23**	**91.34**	**39.58**	**26.20**
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
ViDiT-Q + TCEC	4/8	**64.97**	**56.90**	96.01	59.42	97.01	90.05	**37.20**	**26.20**

Model	Type	FP16 (s)	Quantized Inference (s)	
			w/o TCEC	w/ TCEC
Opensora1.2 (51 frames, 480P)	Video (W8A8)	44.56	26.211	**26.316**
CogVideoX (48 frames, 480P)	Video (W8A8)	78.48	49.67	**49.894**
Wan2.1-1.3B (81 frames, 480P)	Video (W8A8)	199	118.45	**119.029**
Flux-dev 1.0 (T = 30)	Img2Vid (W4A4)	26.14	9.947	**9.996**

Key Takeaways

SDXL W4A4: PSNR 21→ 21.9

Video W8A8: Imaging Quality +2.08

Video W4A8: Imaging Quality +3.90

Extra latency <0.5%

Contact: 21831068@zju.edu.cn

Conclusion

Key Takeaways

- Cumulative error is the missing piece in low-bit diffusion quantization 。
- TCEC provides a closed-form and practical online correction framework 。
- It improves image and video generation quality with negligible overhead 。

Contact: 21831068@zju.edu.cn

Thank You

Contact: 21831068@zju.edu.cn