QUANTIZATION-AWARE KALMAN ESTIMATION FOR DIFFUSION SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Quantization offers a practical path to deploying diffusion models with reduced memory and computation, but aggressive compression can cause quantized outputs to deviate substantially from their full-precision counterparts. Sampling-stage correction methods seek to compensate for such deviations during sampling, but existing approaches rely primarily on local information and underexploit trajectory history, limiting their ability to correct errors that propagate across timesteps. In this work, we formulate sampling with a quantized denoiser as an online estimation problem, using the history of quantized denoiser outputs to recover the underlying full-precision outputs required by the sampler. We propose *QuAKE*, a Quantization-Aware Kalman Estimator that combines a smooth trajectory prior with a conditional Gaussian observation model. At each sampling step, QuAKE recursively updates the posterior over the output window in closed form and feeds its posterior mean to the sampler. QuAKE is a lightweight plug-and-play corrector that requires no modification to the quantized network and naturally supports arbitrary high-order multistep ODE samplers. Experiments across W4A4-quantized text-to-image diffusion models show that QuAKE consistently outperforms existing methods in reducing the distributional discrepancy from full-precision sampling.

1 INTRODUCTION

Diffusion models have achieved strong performance in image synthesis, text-to-image generation, and other generative tasks (Ho et al., 2020; Dhariwal & Nichol, 2021; Rombach et al., 2022; Saharia et al., 2022; Liu et al., 2023). However, deploying these models on resource-constrained hardware remains challenging because of their large memory consumption and the repeated denoiser evaluations required during iterative sampling (Li et al., 2023; He et al., 2023). Post-training quantization (PTQ) offers a practical path to deploying pretrained diffusion models by converting them to low precision using only a small calibration set, thereby reducing memory and inference costs without additional training (Li et al., 2023; Shang et al., 2023; Li et al., 2024b).

Despite these benefits, aggressive low-bit PTQ can still substantially perturb denoiser outputs and alter the resulting sampling trajectory. Because the denoiser output at each step determines the next sample state, quantization errors are fed back into subsequent model evaluations and can accumulate throughout sampling, shifting the generated distribution away from full-precision sampling (He et al., 2023; Yao et al., 2024; Liu et al., 2025). This has motivated sampling-stage correction methods, which use estimated quantization-error statistics to steer the quantized sampling process toward its full-precision counterpart without modifying the quantized network (He et al., 2023; Yao et al., 2024; Zeng et al., 2025; Liu et al., 2025; Zhong et al., 2025; Ryu et al., 2026).

Yet existing sampling-stage correction methods are typically local in time, constructing the correction at each step primarily from the current quantized evaluation while overlooking useful temporal structure in the denoiser outputs. Because these outputs empirically evolve smoothly along the sampling trajectory, past quantized evaluations remain informative about the underlying full-precision denoiser outputs at subsequent timesteps.

Motivated by this observation, we formulate quantized diffusion sampling as an online state-estimation problem, where the latent state to be estimated is the full-precision denoiser-output window required by the numerical sampler, while the quantized outputs serve as noisy observations.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



Prompt: *Awesome dog samurai warrior with katana, Anthropomorphic, Ornate Armor, Cinematic lighting, Volumetric lighting, Epic composition, Photorealism, Bokeh blur, Very high detail, Sony Alpha 7, Character design, Unreal Engine, Octane render, HDR, Subsurface scattering*

Figure 1: As a lightweight plug-and-play method, QuAKE corrects quantization-induced output shifts with negligible additional overhead.

We propose *QuAKE*, a Quantization-Aware Kalman Estimator that combines a smooth trajectory prior with a conditional Gaussian observation model. At each sampling step, QuAKE first uses the trajectory prior to predict the current full-precision output window, then corrects this prediction using the current quantized output. The corrected output-window estimate is then supplied to the original sampler to advance the diffusion sampling trajectory to the next sample state. Through this recursive prediction–correction process, QuAKE continuously integrates information from the observation history into the sampling trajectory. Because its state directly represents the output window consumed by the sampler, QuAKE naturally supports arbitrary high-order multistep ODE samplers. As a plug-and-play sampler-side corrector, QuAKE requires no modification to the quantized network and adds only lightweight filtering operations, resulting in negligible overhead relative to a denoiser evaluation, as illustrated in Figure 1.

Our contributions are summarized as follows:

- We formulate sampling-stage correction for quantized diffusion models as a history-conditioned online state-estimation problem, where the full-precision denoiser-output window required by the numerical sampler is sequentially inferred from quantized observations along the sampling trajectory.
- We propose QuAKE, a closed-form Kalman estimator that combines a smooth trajectory prior with a conditional Gaussian observation model to fuse information across sampling steps. As a lightweight plug-and-play corrector, QuAKE requires no modification to the quantized network and naturally supports arbitrary high-order multistep ODE samplers.
- We evaluate QuAKE across different W4A4-quantized text-to-image diffusion models, demonstrating consistent improvements over existing sampling-stage correction methods in recovering the full-precision sampling distribution.

2 Preliminaries

2.1 Diffusion ODE Sampling

Deterministic diffusion sampling can be formulated through the probability-flow ODE (Song et al., 2020; Lu et al., 2022)

$$\frac{\mathrm{d}x_t}{\mathrm{d}t} = \alpha_t x_t - \frac{1}{2}\beta_t^2 \nabla_{x_t} \log p_t(x_t \mid c), \tag{1}$$

where x_t is the sample state at time t, c denotes optional conditioning information, and α_t and β_t are the drift and diffusion coefficients, respectively. Sampling proceeds backward in diffusion

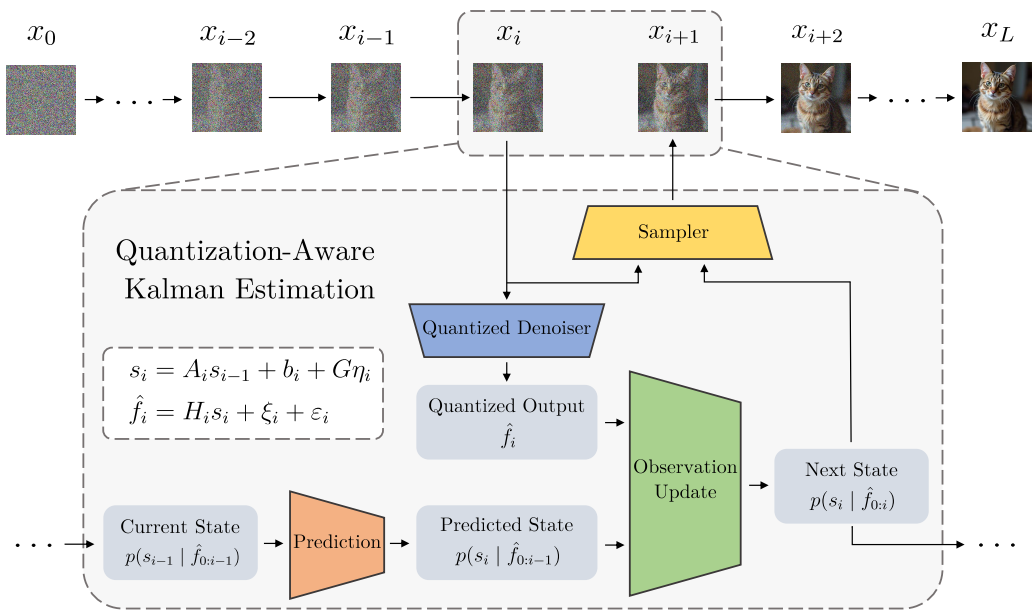


Figure 2: Overview of QuAKE. At each sampling step, QuAKE first predicts the full-precision denoiser-output window from the previous posterior, then updates the prediction using the current quantized output, and finally supplies the corrected estimate to the sampler.

time from a noise sample, with the score function $\nabla_{x_t} \log p_t(x_t \mid c)$ evaluated using a pretrained denoising model f_θ.

2.2 High-Order Multistep Samplers

Numerical samplers solve equation 1 on a decreasing time grid $\{t_i\}_{i=0}^{L}$, where t_0 corresponds to the noise endpoint and t_L to the data endpoint. Let $x_i := x_{t_i}$ denote the sample state at timestep t_i. The corresponding denoiser output is

$$f_i = f_\theta(x_i, t_i, c). \tag{2}$$

Modern fast samplers often employ high-order multistep ODE solvers that compute the next sample state using a window of N recent denoiser outputs (Lu et al., 2022; Zhao et al., 2023; Lu et al., 2025). We denote this output window at step i by

$$s_i = [f_i, f_{i-1}, \ldots, f_{i-N+1}]^\top, \tag{3}$$

and express the corresponding sampling update as

$$x_{i+1} = \Psi_i(x_i; s_i), \tag{4}$$

where Ψ_i denotes the solver-specific update rule.

3 Quantization-Aware Kalman Estimation

In this section, we formulate QuAKE as a sequential framework for estimating the full-precision denoiser-output window from quantized outputs. We first define the estimation objective in Sec. 3.1, and then establish the state-space model through the smooth trajectory prior in Sec. 3.2 and conditional Gaussian observation model in Sec. 3.3. Finally, Sec. 3.4 derives the recursive update, and Sec. 3.5 describes how the estimated outputs are supplied to the sampler. An overview of QuAKE is illustrated in Figure 2.

3.1 Online Estimation Formulation

Let f_θ denote the full-precision denoiser and $f_{\hat{\theta}}$ its corresponding quantized counterpart. Evaluated at the same sample state x_i, their outputs at step i are

$$f_i = f_\theta(x_i, t_i, c), \qquad \hat{f}_i = f_{\hat{\theta}}(x_i, t_i, c). \tag{5}$$

Here, f_i and $\hat{f}_i$ denote scalar output elements, following the element-wise treatment adopted in prior work (He et al., 2023; Yao et al., 2024; Liu et al., 2025; Ryu et al., 2026). During quantized sampling, only the quantized output $\hat{f}_i$ is observable, whereas the underlying full-precision output f_i is unavailable. At each sampling step i, the estimation target is the current full-precision denoiser-output window s_i defined in equation 3. Conditioned on all quantized outputs observed up to the current step, we formulate its estimation as

$$s_i^\star = \arg\max_{s_i} p(s_i \mid \hat{f}_{0:i}), \tag{6}$$

where $\hat{f}_{0:i} := [\hat{f}_0, \ldots, \hat{f}_i]^\top$ collects the complete sequence of quantized denoiser outputs available to the estimator up to step i. The estimated full-precision output window $s_i^\star$ is then used by the sampler to compute the next sample state

$$x_{i+1} = \Psi_i(x_i; s_i^\star)\,. \tag{7}$$

Therefore, for sampling-stage correction, given the quantized denoiser $f_{\hat{\theta}}$ and the numerical sampler updates $\{\Psi_i\}_{i=0}^{L-1}$, the task reduces to modeling and solving the estimation problem in equation 6.

3.2 Smooth Trajectory Prior

Denoiser outputs along a sampling trajectory often exhibit local smoothness (Liu et al., 2022; Zhang & Chen, 2022; Lu et al., 2022; Zhao et al., 2023; Lu et al., 2025). Motivated by this property, we assume that the current full-precision denoiser output can be linearly predicted from its recent values, with the remaining discrepancy modeled as a residual term:

$$f_i = \sum_{k=1}^{N} a_{i,k} f_{i-k} + w_i, \qquad w_i \sim \mathcal{N}(\mu_i, Q_i), \tag{8}$$

where $\{a_{i,k}\}_{k=1}^{N}$ are predefined extrapolation coefficients, and μ_i and Q_i denote the mean and variance of the prediction residual w_i.

Defining the zero-mean process noise

$$\eta_i = w_i - \mu_i, \qquad \eta_i \sim \mathcal{N}(0, Q_i), \tag{9}$$

we can rewrite the linear extrapolation in equation 8 as a first-order state transition of s_i:

$$s_i = A_i s_{i-1} + b_i + G\eta_i, \tag{10}$$

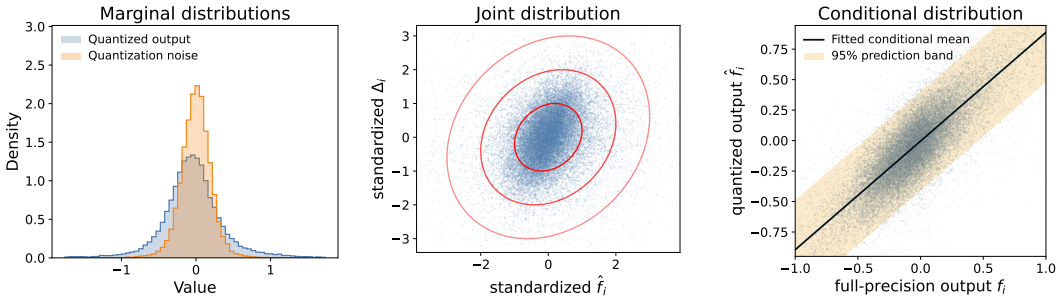
where

$$A_i = \begin{bmatrix} a_{i,1} & a_{i,2} & \cdots & a_{i,N-1} & a_{i,N} \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \qquad b_i = \mu_i e_1, \qquad G = e_1, \tag{11}$$

and $e_1 = [1, 0, \ldots, 0]^\top$. The first row of A_i extrapolates the current full-precision output from the preceding output window, while the remaining rows shift the output history forward by one step. This state transition model serves as the smooth trajectory prior for sequential estimation.

3.3 Conditional Gaussian Observation

We define the quantization noise at sampling step i as $\Delta_i = \hat{f}_i - f_i$. Empirically, the quantized output $\hat{f}_i$ and the corresponding quantization noise Δ_i exhibit approximately Gaussian marginals



(a) Marginal distributions (b) Joint distribution (c) Conditional distribution

Figure 3: Empirical statistics of quantization noise. (a) Marginal distributions of the quantized output $\hat{f}_i$ and the quantization noise Δ_i. (b) Joint distribution of $(\hat{f}_i, \Delta_i)$ with fitted 1σ, 2σ, and 3σ covariance ellipses. (c) Conditional distribution of $\hat{f}_i$ given f_i, with the fitted conditional mean and the 95% prediction interval. The plots use 32,768 samples from step index 5 of Sana-0.6B and its quantized counterpart.

and an elliptical joint distribution, motivating a joint Gaussian model for $(\hat{f}_i, \Delta_i)$ (Zeng et al., 2025; Ryu et al., 2026), as illustrated in Figure 3. Since f_i is an affine transformation of $(\hat{f}_i, \Delta_i)$, this joint model induces the following conditional Gaussian model for $\hat{f}_i$ given f_i:

$$\hat{f}_i \mid f_i \sim \mathcal{N}\left(\gamma_i f_i + \xi_i, R_i\right), \tag{12}$$

where γ_i, ξ_i, and R_i are timestep-dependent parameters.

Equivalently, equation 12 can be expressed as the linear observation equation

$$\hat{f}_i = \gamma_i f_i + \xi_i + \varepsilon_i, \qquad \varepsilon_i \sim \mathcal{N}(0, R_i). \tag{13}$$

Since the current full-precision output f_i is the first component of the output-window state s_i, the observation equation can be written as

$$\hat{f}_i = H_i s_i + \xi_i + \varepsilon_i, \qquad H_i = \gamma_i e_1^\top. \tag{14}$$

Thus, the current quantized denoiser output provides a noisy observation of the current full-precision output within the output-window state.

3.4 Recursive Posterior Estimation

Combining the state transition equation in equation 10 with the conditional Gaussian observation equation in equation 14 gives the linear Gaussian state-space system

$$\begin{aligned} s_i &= A_i s_{i-1} + b_i + G\eta_i, & \eta_i &\sim \mathcal{N}(0, Q_i), \\ \hat{f}_i &= H_i s_i + \xi_i + \varepsilon_i, & \varepsilon_i &\sim \mathcal{N}(0, R_i). \end{aligned} \tag{15}$$

We assume that the process and observation noises are mutually and temporally independent. Given a Gaussian initial prior for s_0, the posterior $p(s_i \mid \hat{f}_{0:i})$ remains Gaussian at every step and can be updated recursively.

Prediction. Suppose the posterior at the preceding step is

$$p(s_{i-1} \mid \hat{f}_{0:i-1}) = \mathcal{N}\left(m_{i-1|i-1}, P_{i-1|i-1}\right). \tag{16}$$

Before observing $\hat{f}_i$, the predictive distribution is obtained by propagating this posterior through the state transition equation:

$$p(s_i \mid \hat{f}_{0:i-1}) = \int p(s_i \mid s_{i-1}) p(s_{i-1} \mid \hat{f}_{0:i-1}) \, \mathrm{d}s_{i-1}. \tag{17}$$

Because the transition is affine and both distributions are Gaussian, this integration gives

$$p(s_i \mid \hat{f}_{0:i-1}) = \mathcal{N}\left(m_{i|i-1}, P_{i|i-1}\right), \tag{18}$$

where

$$m_{i|i-1} = A_i m_{i-1|i-1} + b_i, \qquad P_{i|i-1} = A_i P_{i-1|i-1} A_i^\top + G Q_i G^\top. \tag{19}$$

Algorithm 1 Quantized diffusion sampling with QuAKE

Require: Initial sample x_0, optional conditioning c, time grid $\{t_i\}_{i=0}^{L}$, quantized denoiser $f_{\hat{\theta}}$, solver updates $\{\Psi_i\}_{i=0}^{L-1}$, calibrated model parameters, and initial predictive statistics $m_{0|-1}$ and $P_{0|-1}$
Ensure: Generated sample x_L

for $i = 0, \dots, L-1$ **do**
  # PREDICTION
  if $i = 0$ **then**
    $m_{i|i-1} \leftarrow m_{0|-1}, \quad P_{i|i-1} \leftarrow P_{0|-1}$
  else
    $m_{i|i-1} \leftarrow A_i m_{i-1|i-1} + b_i, \quad P_{i|i-1} \leftarrow A_i P_{i-1|i-1} A_i^\top + G Q_i G^\top$
  end if
  # OBSERVATION UPDATE
  $\hat{f}_i \leftarrow f_{\hat{\theta}}(x_i, t_i, c)$
  $r_i \leftarrow \hat{f}_i - H_i m_{i|i-1} - \xi_i, \quad S_i \leftarrow H_i P_{i|i-1} H_i^\top + R_i, \quad K_i \leftarrow P_{i|i-1} H_i^\top S_i^{-1}$
  $m_{i|i} \leftarrow m_{i|i-1} + K_i r_i, \quad P_{i|i} \leftarrow P_{i|i-1} - K_i S_i K_i^\top$
  # SAMPLING
  $x_{i+1} \leftarrow \Psi_i\big(x_i; m_{i|i}\big)$
end for

Observation update. Once the current quantized output $\hat{f}_i$ is observed, Bayes' rule gives

$$p(s_i \mid \hat{f}_{0:i}) \propto p(\hat{f}_i \mid s_i)\, p(s_i \mid \hat{f}_{0:i-1}). \tag{20}$$

Because the predictive distribution and observation likelihood are both Gaussian, applying Bayes' rule yields the following closed-form Kalman update. For compactness, we define the innovation r_i, its covariance S_i, and the Kalman gain K_i as

$$r_i = \hat{f}_i - H_i m_{i|i-1} - \xi_i, \qquad S_i = H_i P_{i|i-1} H_i^\top + R_i, \qquad K_i = P_{i|i-1} H_i^\top S_i^{-1}. \tag{21}$$

The resulting posterior is

$$p(s_i \mid \hat{f}_{0:i}) = \mathcal{N}\left(m_{i|i}, P_{i|i}\right), \tag{22}$$

where

$$m_{i|i} = m_{i|i-1} + K_i r_i, \qquad P_{i|i} = P_{i|i-1} - K_i S_i K_i^\top. \tag{23}$$

This recursive update aggregates the complete observation history into a compact posterior, without explicitly revisiting preceding observations. Moreover, its covariance-aware weighting adaptively balances the trajectory prediction against the current quantized observation, preserving reliable historical information while suppressing uncertain observation noise. All the required model statistics can be estimated offline through calibration. Detailed derivations are provided in Appendix B, and the calibration procedure is described in Appendix C.

3.5 POSTERIOR READOUT FOR SAMPLING

Since the posterior in equation 22 is Gaussian, its mean coincides with the MAP estimate in equation 6. We therefore read out the corrected denoiser-output window as $s_i^\star = m_{i|i}$ and supply it directly to the original numerical solver:

$$x_{i+1} = \Psi_i\big(x_i; m_{i|i}\big)\,. \tag{24}$$

Because the posterior is defined over the complete output window, each new quantized observation updates not only the current denoiser output but also the retained historical outputs through their posterior correlations. QuAKE thus provides the high-order sampler with a temporally consistent, jointly corrected output history. Moreover, it modifies only the output window supplied to the sampler while leaving the solver-specific update rule unchanged, making it directly compatible with different high-order multistep solvers. The complete QuAKE sampling procedure is summarized in Algorithm 1.

Table 1: Quantitative results on MJHQ-30K and sDCI using the DPM-Solver++ sampler. KID values are reported after multiplication by 10^3. IR stands for ImageReward. Boldface marks the best result among sampling-stage correction methods for each model.

Model	Method	MJHQ-30K					sDCI				
		Similarity (↓)		Quality (↑)			Similarity (↓)		Quality (↑)		
		FID	KID	CLIPScore	CLIP-IQA	IR	FID	KID	CLIPScore	CLIP-IQA	IR
FLUX.1-dev	Full-Precision	–	–	0.3194	0.8989	0.9017	–	–	0.3130	0.8449	0.8589
	Quantized	7.30	0.257	0.3179	0.9057	0.8887	7.22	0.278	0.3121	0.8540	0.8543
	Bias Correction	7.24	0.264	0.3183	0.9045	**0.8891**	7.21	0.304	0.3127	0.8513	0.8509
	TAC-Diffusion	7.31	0.304	0.3179	0.9049	0.8890	7.21	0.315	**0.3131**	**0.8544**	**0.8562**
	Q-Drift	7.16	0.209	0.3179	**0.9051**	0.8860	7.03	0.213	0.3117	0.8539	0.8501
	QuAKE (Ours)	**7.02**	**0.158**	**0.3184**	0.9042	0.8822	**6.91**	**0.148**	0.3127	0.8534	0.8445
Sana-0.6B	Full-Precision	–	–	0.3375	0.9003	1.0990	–	–	0.3310	0.9182	0.9420
	Quantized	10.23	1.394	0.3360	0.8936	1.0481	10.14	1.465	0.3285	0.9078	0.8868
	Bias Correction	22.50	7.755	0.3321	0.8764	0.9522	23.51	9.246	0.3234	0.8802	0.7707
	TAC-Diffusion	22.36	7.639	0.3320	0.8766	0.9576	23.04	8.970	0.3232	0.8768	0.7687
	Q-Drift	9.73	1.072	0.3359	0.8890	1.0356	9.72	1.280	0.3285	0.9062	0.8699
	QuAKE (Ours)	**7.90**	**0.289**	**0.3363**	**0.8951**	**1.0496**	**7.95**	**0.317**	**0.3298**	**0.9126**	**0.8753**
PixArt-α	Full-Precision	–	–	0.3302	0.8910	1.0004	–	–	0.3140	0.9141	0.5967
	Quantized	15.87	3.177	0.3282	0.8955	0.9131	15.70	3.558	0.3118	0.9122	0.4851
	Bias Correction	28.32	9.511	0.3275	**0.9037**	0.8969	29.56	11.733	0.3146	**0.9222**	0.4475
	TAC-Diffusion	28.16	9.573	0.3275	0.9014	**0.9004**	29.17	11.639	**0.3149**	0.9154	0.4467
	Q-Drift	14.17	2.351	0.3278	0.8873	0.8940	13.93	2.646	0.3115	0.9032	**0.4676**
	QuAKE (Ours)	**13.71**	**1.987**	**0.3282**	0.8909	0.8934	**13.84**	**2.401**	0.3118	0.9113	0.4552
PixArt-Σ	Full-Precision	–	–	0.3318	0.9066	0.9588	–	–	0.3238	0.9143	0.8597
	Quantized	17.22	4.062	0.3296	0.8978	0.8661	16.97	4.687	0.3287	0.8863	0.7943
	Bias Correction	24.38	7.528	0.3294	0.8891	**0.8757**	24.12	8.723	0.3288	0.8727	0.7755
	TAC-Diffusion	24.07	7.281	0.3297	0.8889	0.8637	23.77	8.564	**0.3304**	0.8751	0.7808
	Q-Drift	17.22	4.052	0.3298	0.8934	0.8546	16.87	4.592	0.3291	0.8790	0.7813
	QuAKE (Ours)	**15.99**	**3.401**	**0.3299**	**0.8980**	0.8705	**15.50**	**3.859**	0.3285	**0.8863**	**0.7913**

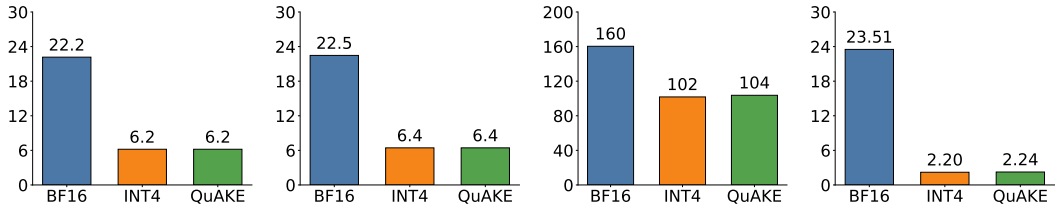
4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models and datasets. We conduct experiments on four text-to-image diffusion models: FLUX.1-dev (Black Forest Labs, 2024), Sana-0.6B (Xie et al., 2024), PixArt-α (Chen et al., 2024b), and PixArt-Σ (Chen et al., 2024a). The full-precision models use 16-bit precision, and all models are quantized with SVDQuant (Li et al., 2024b) under the W4A4 setting. Following prior work (Liu et al., 2025; Ryu et al., 2026), we evaluate on MJHQ-30K (Li et al., 2024a) and sDCI (Urbanek et al., 2024), using $1{,}024$ samples from each dataset to calibrate timestep-wise statistics and a disjoint set of $5{,}000$ samples for evaluation. For sampling, we use DPM-Solver++ (Lu et al., 2025) in the main experiments and further evaluate UniPC (Zhao et al., 2023) in the supplementary experiments. Both samplers use second-order multistep updates with 20 sampling steps.

Compared methods. We use the full-precision model and the uncorrected quantized model as two primary baselines. We additionally compare QuAKE with representative sampling-time correction methods, including Bias Correction (PTQD (He et al., 2023) and D^2-DPM (Zeng et al., 2025), which reduce to the same conditional-mean correction under the deterministic ODE sampling setting), TAC-Diffusion (Yao et al., 2024), and Q-Drift (Ryu et al., 2026). As no official implementations of TAC-Diffusion or Q-Drift are publicly available, we reimplement both methods following their original papers. For a fair comparison, all correction methods use the same quantized backbone, calibration and evaluation sets, sampler, initial noise samples, and number of sampling steps.

Metrics. We use FID (Heusel et al., 2017) and KID (Bińkowski et al., 2018) to quantify the distributional gap between quantized generation and full-precision generation, treating the latter as the



(a) DiT Size (GiB) (b) DiT Inference Memory (GiB) (c) Per-Step Latency (ms) (d) E2E Latency (s)

Figure 4: Memory and latency measurements for generating a single image with FLUX.1-dev. Compared with standard quantized sampling, QuAKE introduces negligible additional memory usage and latency. All measurements are conducted on a single NVIDIA A100-SXM4-40GB GPU, with CPU offloading enabled for BF16 sampling.

reference distribution. We further report CLIPScore (Hessel et al., 2021) to evaluate text–image semantic alignment, CLIP-IQA (Wang et al., 2023) to assess perceptual image quality, and ImageReward (Xu et al., 2023) to measure human-preference-aligned generation quality. Additional experimental details are provided in Appendix E.

4.2 Main Results

Table 1 reports the main quantitative results with DPM-Solver++ across four text-to-image diffusion models and two evaluation datasets. QuAKE consistently achieves the lowest FID and KID among all sampling-stage correction methods for every model–dataset combination, demonstrating its effectiveness in recovering the output distribution of the full-precision model. Meanwhile, QuAKE remains competitive across semantic alignment, perceptual quality, and human preference. These results indicate that **QuAKE substantially reduces the distributional discrepancy induced by quantization** without compromising overall generation quality, thereby better preserving the behavior of the full-precision model. Appendix F further evaluates QuAKE with the UniPC sampler and observes a consistent overall trend, supporting its generalization across samplers and its applicability to different high-order multistep ODE solvers without solver-specific redesign.

The results also highlight the importance of maintaining temporal consistency when correcting quantized outputs for high-order multistep samplers. Bias Correction and TAC-Diffusion primarily correct the current denoiser evaluation, without explicitly accounting for its consistency with the historical outputs retained by the sampler, as detailed in Appendix A.2. Since high-order multistep updates jointly rely on the current and previous denoiser outputs, independently correcting the current output can introduce inconsistencies within the output history, which may in turn degrade the resulting sampling trajectory. This behavior is reflected by the increased FID and KID of these methods on several models. In contrast, QuAKE treats the complete output window as the estimation target and recursively combines a smooth trajectory prior with successive quantized observations, producing a temporally consistent corrected history. This enables our method to consistently reduce both FID and KID over the uncorrected quantized baseline across all model–dataset combinations.

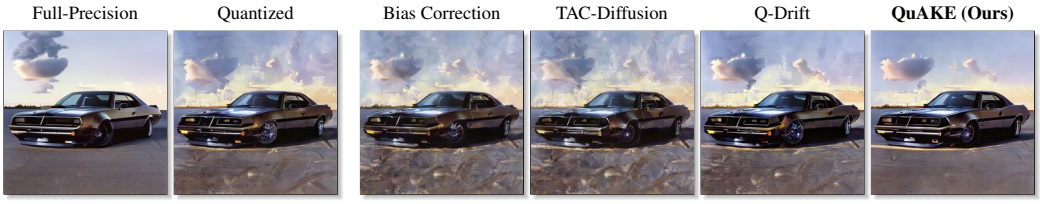
Figure 4 evaluates the memory and latency overhead introduced by QuAKE. Compared with standard quantized sampling, the calibrated statistics required by our method occupy only 0.25 MiB of additional memory for FLUX.1-dev, while its additional latency is negligible compared with the cost of a denoising network evaluation. This confirms that our method can serve as a lightweight plug-and-play correction module, providing substantial improvements in distributional recovery while largely preserving the memory and computational advantages of low-precision inference.

Figure 5 presents qualitative comparisons across different text-to-image diffusion models. Quantization can noticeably perturb the generated outputs from their full-precision counterparts, leading to potential deviations in visual content and structure. In contrast, QuAKE effectively compensates for these quantization-induced deviations and produces images that more closely resemble the corresponding full-precision outputs. Compared with existing sampling-stage correction methods, our method better preserves the visual characteristics of the full-precision generation, consistent with the reduced distributional discrepancy observed in FID and KID.



Prompt: *a cute boy with a plane and sunglasses is reaching for the sky, in the style of shaped canvas, retro filters, poster, photo taken with provia, conceptual sculptures*

(a) FLUX.1-dev



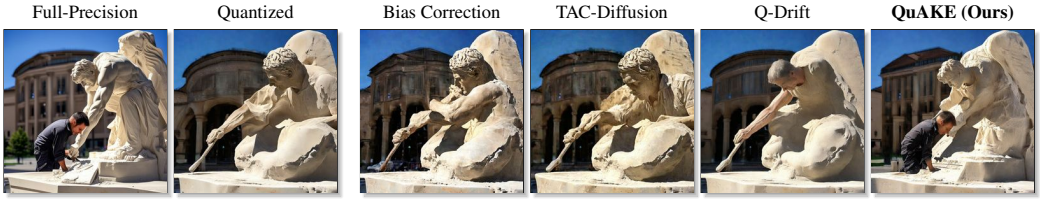
Prompt: *1998 Dodge Charger, full view*

(b) Sana-0.6B



Prompt: *Beautiful 16 years old girl riding a horse at the beach, fast motion, sunset, professional color grading, soft shadows, no contrast, clean sharp focus, national geographic photography.*

(c) PixArt-α



Prompt: *A man is seen leaning over a sculpture while using a chisel to carve it. There is a large building in the background.*

(d) PixArt-Σ

Figure 5: Qualitative comparison across different text-to-image diffusion models using DPM-Solver++. QuAKE effectively compensates for the generation deviations introduced by quantization, producing results closer to the full-precision outputs than existing sampling-stage correction methods.

5 Conclusion

In this work, we propose QuAKE, a Quantization-Aware Kalman Estimator for sampling-stage correction of quantized diffusion models. QuAKE formulates quantized diffusion sampling as an online estimation problem, enabling correction to exploit the temporal structure and historical information along the sampling trajectory. Building on this formulation, QuAKE combines a smooth trajectory prior with a conditional Gaussian observation model to recursively refine quantized outputs, while remaining lightweight, plug-and-play, and compatible with different high-order multistep ODE samplers. Experiments across multiple W4A4-quantized text-to-image diffusion models show that QuAKE consistently reduces the distributional discrepancy from full-precision generation compared with existing correction methods, with negligible memory and latency overhead.

AI USE STATEMENT

Generative AI tools were used to improve the readability and language of the manuscript and to assist with the creation and editing of software code. All AI-assisted text was manually reviewed and revised by the authors, and all AI-assisted code was inspected and tested for correctness. The authors take full responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, the main text provides the complete formulation and algorithmic framework of QuAKE, while Appendix B further presents detailed derivations of the proposed estimation procedure. The calibration procedure for the statistics used by QuAKE is described in Appendix C. Additional implementation details of QuAKE are provided in Appendix D. Detailed model configurations, quantization settings, evaluation protocols, and implementation environment are documented in Appendix E. During the review period, we will provide anonymous code files to support reproducibility. The complete codebase and associated resources will be made publicly available upon acceptance.

REFERENCES

Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. *arXiv preprint arXiv:1801.01401*, 2018.

Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-Σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In *European Conference on Computer Vision*, pp. 74–91. Springer, 2024a.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *International conference on learning representations*, volume 2024, pp. 57611–57640, 2024b.

Yi-Chung Chen, Zhi-Kai Huang, and Jing-Ren Chen. Stepbaq: Stepping backward as correction for quantized diffusion models. *Advances in Neural Information Processing Systems*, 37:54054–54078, 2024c.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Yefei He, Luping Liu, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. PTQD: Accurate post-training quantization for diffusion models. *Advances in Neural Information Processing Systems*, 36:13237–13249, 2023.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 2021.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Daiqing Li, Aleks Kamko, Ehsan Akhgari, Ali Sabet, Linmiao Xu, and Suhail Doshi. Playground v2.5: Three insights towards enhancing aesthetic quality in text-to-image generation. *arXiv preprint arXiv:2402.17245*, 2024a.

Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. Svdquant: Absorbing outliers by low-rank components for 4-bit diffusion models. *arXiv preprint arXiv:2411.05007*, 2024b.

Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17535–17545, 2023.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022.

Songwei Liu, Chao Zeng, Chenqian Yan, Xurui Peng, Xing Wang, Fangmin Chen, and Xing Mei. Error propagation mechanisms and compensation strategies for quantized diffusion. *arXiv preprint arXiv:2508.12094*, 2025.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22 (4):730–751, 2025.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in GAN evaluation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11410–11420, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Sooyoung Ryu, Mathieu Salzmann, and Saqib Javed. Q-drift: Quantization-aware drift correction for diffusion model sampling. *arXiv preprint arXiv:2603.18095*, 2026.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1972–1981, 2023.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Jack Urbanek, Florian Bordes, Pietro Astolfi, Mary Williamson, Vasu Sharma, and Adriana Romero-Soriano. A picture is worth more than 77 text tokens: Evaluating CLIP-style models on dense captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26700–26709, 2024.

Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring CLIP for assessing the look and feel of images. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 2555–2563, 2023.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

Yuzhe Yao, Feng Tian, Jun Chen, Haonan Lin, Guang Dai, Yong Liu, and Jingdong Wang. Timestep-aware correction for quantized diffusion models. In *European Conference on Computer Vision*, pp. 215–232. Springer, 2024.

Qian Zeng, Jie Song, Han Zheng, Hao Jiang, and Mingli Song. D2-DPM: Dual denoising for quantized diffusion probabilistic models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9806–9814, 2025.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. UniPC: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:49842–49869, 2023.

Yunshan Zhong, Weiqi Yan, and Yuxin Zhang. Test-time iterative error correction for efficient diffusion models. *arXiv preprint arXiv:2511.06250*, 2025.

A Related Work

This appendix discusses related work on high-order ODE samplers and sampling-stage correction methods for quantized diffusion models.

A.1 High-Order Multistep ODE Samplers for Diffusion Models

High-order multistep ODE samplers reuse recent denoiser evaluations to construct higher-order approximations to the diffusion ODE solution, thereby reducing discretization error and enabling accurate sampling with fewer steps (Liu et al., 2022; Zhang & Chen, 2022; Lu et al., 2022; Zhao et al., 2023; Lu et al., 2025).

Take DPM-Solver++ (Lu et al., 2025) as an example. Under the data-prediction parameterization, the reverse-time ODE can be written as

$$\mathrm{d}x_t = x_t \,\mathrm{d}\log\sigma_t + \sigma_t e^{\lambda_t} f_\theta(x_t, t, c)\,\mathrm{d}\lambda_t, \tag{25}$$

where σ_t and λ_t denote the marginal noise scale and log-SNR coordinate, respectively. Define

$$\sigma_i := \sigma_{t_i}, \qquad \lambda_i := \lambda_{t_i}, \qquad f(\lambda) := f_\theta\big(x_{t(\lambda)}, t(\lambda), c\big), \qquad f_i := f(\lambda_i). \tag{26}$$

Integrating equation 25 from t_i to t_{i+1} yields

$$x_{i+1} = \frac{\sigma_{i+1}}{\sigma_i} x_i + \sigma_{i+1} \int_{\lambda_i}^{\lambda_{i+1}} e^{\lambda} f(\lambda)\,\mathrm{d}\lambda. \tag{27}$$

To construct an N-th order update, DPM-Solver++ applies an $(N-1)$-th order Taylor expansion around λ_i:

$$f(\lambda) = \sum_{n=0}^{N-1} \frac{(\lambda-\lambda_i)^n}{n!} f_i^{(n)} + \mathcal{O}\big((\lambda-\lambda_i)^N\big), \tag{28}$$

where

$$f_i^{(n)} := \left.\frac{\mathrm{d}^n f(\lambda)}{\mathrm{d}\lambda^n}\right|_{\lambda=\lambda_i} \tag{29}$$

denotes the n-th total derivative of the data prediction along the sampling trajectory. Substituting equation 28 into equation 27 gives

$$x_{i+1} = \frac{\sigma_{i+1}}{\sigma_i} x_i + \sigma_{i+1} \sum_{n=0}^{N-1} f_i^{(n)} \int_{\lambda_i}^{\lambda_{i+1}} e^{\lambda} \frac{(\lambda-\lambda_i)^n}{n!}\,\mathrm{d}\lambda + \mathcal{O}\big((\lambda_{i+1}-\lambda_i)^{N+1}\big). \tag{30}$$

The exponentially weighted integrals can be evaluated analytically, while the derivatives are approximated by finite differences of the N recent denoiser evaluations. Absorbing the resulting integration and finite-difference coefficients into $\{\omega_{i,j}\}_{j=0}^{N-1}$ yields the multistep DPM-Solver++ update

$$x_{i+1} = \frac{\sigma_{i+1}}{\sigma_i} x_i + \sum_{j=0}^{N-1} \omega_{i,j} f_{i-j}. \tag{31}$$

More generally, other high-order multistep ODE samplers can also be formulated in this form using the denoiser-output window $s_i = [f_i, f_{i-1}, \ldots, f_{i-N+1}]^\top$. We uniformly abstract them throughout this work as

$$x_{i+1} = \Psi_i(x_i; s_i). \tag{32}$$

Connection to QuAKE. The connection to QuAKE is twofold. First, because high-order multistep samplers require the entire output window s_i to advance the sampling trajectory, we take s_i rather than only the current denoiser output f_i as the estimation target. Second, their Taylor-based constructions implicitly exploit the local smoothness of denoiser outputs along the sampling trajectory, which motivates the smooth trajectory prior introduced in Section 3.2.

A.2 Sampling-Stage Correction for Quantized Diffusion Models

Sampling-stage correction methods compensate for quantization-induced deviations by modifying the sampling process while leaving the quantized denoiser unchanged. Representative methods are described below using the formulation adopted in this work.

Bias correction. PTQD (He et al., 2023) and D^2-DPM (Zeng et al., 2025) model the conditional statistics of quantization noise and use the estimated conditional mean to correct the quantized denoiser output. Their complete correction rules are designed for stochastic DDPM-style sampling and additionally adjust the stochastic noise term in the sampling equation. Under the deterministic ODE sampling setting considered in this work, the conditional-mean correction can be written as

$$\tilde{f}_i = \hat{f}_i - \mathbb{E}[\Delta_i \mid \hat{f}_i]. \tag{33}$$

The corrected output $\tilde{f}_i$ is stored in the sampler output cache in place of the raw quantized output.

TAC-Diffusion. TAC-Diffusion (Yao et al., 2024) decomposes sampling-stage correction into Noise Estimation Reconstruction (NER) and Input Bias Correction (IBC). NER rescales the quantized denoiser output to reduce its discrepancy from the full-precision output, whereas IBC shifts the denoiser input to compensate for accumulated input bias. Under the present formulation, the resulting corrected denoiser evaluation can be written as

$$\tilde{f}_i = \kappa_i f_{\hat{\theta}}\left(x_i - B_i, t_i, c\right), \tag{34}$$

where κ_i is the timestep-dependent reconstruction coefficient and B_i is the timestep-dependent input bias. The corrected evaluation $\tilde{f}_i$ is then supplied to the sampler.

Q-Drift. Q-Drift (Ryu et al., 2026) treats quantization error as an implicit stochastic perturbation of the sampling dynamics and compensates for the resulting distribution shift through a marginal-preserving drift rescaling. For the N-output window used by a multistep sampler, the correction can be written as

$$\left[\tilde{f}_i, \tilde{f}_{i-1}, \ldots, \tilde{f}_{i-N+1}\right]^\top = (1+\rho_i)\left[\hat{f}_i, \hat{f}_{i-1}, \ldots, \hat{f}_{i-N+1}\right]^\top, \tag{35}$$

where ρ_i is a timestep-dependent drift-rescaling coefficient determined by the conditional variance statistic and the coefficients of the numerical sampler. The rescaled outputs are subsequently used in the sampler output cache. Because ρ_i depends on the discretization coefficients of the sampling update, applying Q-Drift to a different sampler requires a corresponding solver-specific derivation.

DDIM-specific corrections. Several other sampling-stage correction methods derive their correction rules directly from the first-order DDIM recursion. TCEC (Liu et al., 2025) and IEC (Zhong et al., 2025) estimate and compensate for the accumulated sample-state error induced by quantization, while StepbaQ (Chen et al., 2024c) adjusts the sampling steps according to a DDIM-based quantization-error analysis. Since these correction rules explicitly depend on the coefficients and error-propagation structure of the DDIM update, they do not directly provide correction mechanisms for general high-order multistep ODE samplers.

B Detailed Derivations

This appendix first derives the conditional Gaussian observation model in Sec. 3.3, followed by the recursive posterior update in Sec. 3.4. In addition, we prove that our method remains LMMSE-optimal when the process and observation noises are not strictly Gaussian.

B.1 Conditional Gaussian Observation

We first derive how the joint Gaussian model of the quantized output and quantization noise induces the conditional Gaussian observation model in equation 12. Recall that the quantization noise is defined as $\Delta_i = \hat{f}_i - f_i$. We model the quantized output and quantization noise as jointly Gaussian:

$$\begin{bmatrix} \hat{f}_i \\ \Delta_i \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbb{E}[\hat{f}_i] \\ \mathbb{E}[\Delta_i] \end{bmatrix}, \begin{bmatrix} \mathrm{Var}(\hat{f}_i) & \mathrm{Cov}(\hat{f}_i, \Delta_i) \\ \mathrm{Cov}(\hat{f}_i, \Delta_i) & \mathrm{Var}(\Delta_i) \end{bmatrix} \right). \tag{36}$$

Since $f_i = \hat{f}_i - \Delta_i$, the pair $(f_i, \hat{f}_i)$ is obtained through the linear transformation

$$\begin{bmatrix} f_i \\ \hat{f}_i \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{f}_i \\ \Delta_i \end{bmatrix}, \tag{37}$$

and is therefore also jointly Gaussian. The corresponding moments are

$$\mathbb{E}[f_i] = \mathbb{E}[\hat{f}_i] - \mathbb{E}[\Delta_i], \tag{38}$$

$$\mathrm{Var}(f_i) = \mathrm{Var}(\hat{f}_i) + \mathrm{Var}(\Delta_i) - 2\,\mathrm{Cov}(\hat{f}_i, \Delta_i), \tag{39}$$

$$\mathrm{Cov}(f_i, \hat{f}_i) = \mathrm{Var}(\hat{f}_i) - \mathrm{Cov}(\hat{f}_i, \Delta_i). \tag{40}$$

Applying the conditional Gaussian formula gives

$$\hat{f}_i \mid f_i \sim \mathcal{N}\left(\mathbb{E}[\hat{f}_i] + \frac{\mathrm{Cov}(f_i, \hat{f}_i)}{\mathrm{Var}(f_i)}\big(f_i - \mathbb{E}[f_i]\big), \mathrm{Var}(\hat{f}_i) - \frac{\mathrm{Cov}(f_i, \hat{f}_i)^2}{\mathrm{Var}(f_i)}\right). \tag{41}$$

Defining

$$\gamma_i = \frac{\mathrm{Cov}(f_i, \hat{f}_i)}{\mathrm{Var}(f_i)}, \qquad \xi_i = \mathbb{E}[\hat{f}_i] - \gamma_i \mathbb{E}[f_i], \qquad R_i = \mathrm{Var}(\hat{f}_i) - \frac{\mathrm{Cov}(f_i, \hat{f}_i)^2}{\mathrm{Var}(f_i)}, \tag{42}$$

we obtain

$$\hat{f}_i \mid f_i \sim \mathcal{N}\left(\gamma_i f_i + \xi_i, R_i\right). \tag{43}$$

B.2 Recursive Posterior Estimation

This subsection provides a detailed derivation of the recursive posterior update. Recall the linear Gaussian state-space model

$$\begin{aligned} s_i &= A_i s_{i-1} + b_i + G\eta_i, & \eta_i &\sim \mathcal{N}(0, Q_i), \\ \hat{f}_i &= H_i s_i + \xi_i + \varepsilon_i, & \varepsilon_i &\sim \mathcal{N}(0, R_i). \end{aligned} \tag{44}$$

The process and observation noises are assumed to be mutually and temporally independent. In particular, η_i is independent of s_{i-1} and the preceding observations, while ε_i is independent of s_i and the preceding observations. The recursion starts from the Gaussian prior

$$s_0 \sim \mathcal{N}(m_{0|-1}, P_{0|-1}), \tag{45}$$

which serves directly as the predictive distribution at $i = 0$.

Prediction. For $i \geq 1$, suppose the posterior at the preceding step is

$$s_{i-1} \mid \hat{f}_{0:i-1} \sim \mathcal{N}\left(m_{i-1|i-1}, P_{i-1|i-1}\right). \tag{46}$$

The predictive distribution is obtained by propagating this posterior through the state transition:

$$p(s_i \mid \hat{f}_{0:i-1}) = \int p(s_i \mid s_{i-1}) p(s_{i-1} \mid \hat{f}_{0:i-1})\,\mathrm{d}s_{i-1}. \tag{47}$$

Because the transition is affine Gaussian, the predictive distribution remains Gaussian. Its mean is

$$\begin{aligned} m_{i|i-1} &= \mathbb{E}[s_i \mid \hat{f}_{0:i-1}] \\ &= \mathbb{E}\big[A_i s_{i-1} + b_i + G\eta_i \mid \hat{f}_{0:i-1}\big] \\ &= A_i m_{i-1|i-1} + b_i, \end{aligned} \tag{48}$$

where $\mathbb{E}[\eta_i] = 0$. Its covariance is

$$\begin{aligned} P_{i|i-1} &= \mathrm{Cov}(s_i \mid \hat{f}_{0:i-1}) \\ &= \mathrm{Cov}\big(A_i s_{i-1} + G\eta_i \mid \hat{f}_{0:i-1}\big) \\ &= A_i P_{i-1|i-1} A_i^\top + G Q_i G^\top, \end{aligned} \tag{49}$$

where the cross-covariance between s_{i-1} and η_i vanishes by independence. Therefore,

$$s_i \mid \hat{f}_{0:i-1} \sim \mathcal{N}\left(m_{i|i-1}, P_{i|i-1}\right). \tag{50}$$

Observation update. After observing $\hat{f}_i$, Bayes' rule gives

$$p(s_i \mid \hat{f}_{0:i}) = \frac{p(\hat{f}_i \mid s_i)p(s_i \mid \hat{f}_{0:i-1})}{p(\hat{f}_i \mid \hat{f}_{0:i-1})}. \tag{51}$$

From the observation equation, the predictive observation mean is

$$\begin{aligned}\mathbb{E}[\hat{f}_i \mid \hat{f}_{0:i-1}] &= \mathbb{E}\big[H_i s_i + \xi_i + \varepsilon_i \mid \hat{f}_{0:i-1}\big] \\ &= H_i m_{i|i-1} + \xi_i.\end{aligned} \tag{52}$$

Its covariance is

$$\begin{aligned}\mathrm{Cov}(\hat{f}_i \mid \hat{f}_{0:i-1}) &= \mathrm{Cov}\big(H_i s_i + \varepsilon_i \mid \hat{f}_{0:i-1}\big) \\ &= H_i P_{i|i-1} H_i^\top + R_i,\end{aligned} \tag{53}$$

and the state–observation cross-covariance is

$$\begin{aligned}\mathrm{Cov}(s_i, \hat{f}_i \mid \hat{f}_{0:i-1}) &= \mathrm{Cov}\big(s_i, H_i s_i + \varepsilon_i \mid \hat{f}_{0:i-1}\big) \\ &= P_{i|i-1} H_i^\top.\end{aligned} \tag{54}$$

Define the innovation and its covariance as

$$r_i = \hat{f}_i - H_i m_{i|i-1} - \xi_i, \qquad S_i = H_i P_{i|i-1} H_i^\top + R_i, \tag{55}$$

and define the Kalman gain as

$$K_i = P_{i|i-1} H_i^\top S_i^{-1}. \tag{56}$$

Conditioned on the preceding observations, s_i and $\hat{f}_i$ are jointly Gaussian:

$$\begin{bmatrix} s_i \\ \hat{f}_i \end{bmatrix} \;\Big|\; \hat{f}_{0:i-1} \sim \mathcal{N}\left(\begin{bmatrix} m_{i|i-1} \\ H_i m_{i|i-1} + \xi_i \end{bmatrix}, \begin{bmatrix} P_{i|i-1} & P_{i|i-1} H_i^\top \\ H_i P_{i|i-1} & S_i \end{bmatrix} \right). \tag{57}$$

Applying the conditional Gaussian formula yields

$$s_i \mid \hat{f}_{0:i} \sim \mathcal{N}\left(m_{i|i}, P_{i|i}\right), \tag{58}$$

where

$$m_{i|i} = m_{i|i-1} + K_i r_i, \tag{59}$$

$$P_{i|i} = P_{i|i-1} - K_i S_i K_i^\top. \tag{60}$$

These are the recursive posterior-update equations used in Sec. 3.4.

B.3 LMMSE Optimality under Non-Gaussian Noise

This subsection proves that our method remains optimal in the LMMSE sense even when the process and observation noises in the model are not Gaussian. We only assume that the process and observation noises are zero mean, have finite second moments

$$Q_i = \mathrm{Cov}(\eta_i), \qquad R_i = \mathrm{Cov}(\varepsilon_i), \tag{61}$$

and are mutually and temporally uncorrelated and uncorrelated with the initial state. Recall the state-space model

$$\begin{aligned} s_i &= A_i s_{i-1} + b_i + G\eta_i, \\ \hat{f}_i &= H_i s_i + \xi_i + \varepsilon_i. \end{aligned} \tag{62}$$

Assume that $m_{i-1|i-1}$ is already the LMMSE estimator of s_{i-1} based on the observation history $\hat{f}_{0:i-1}$, with estimation error

$$e_{i-1|i-1} = s_{i-1} - m_{i-1|i-1} \tag{63}$$

and its covariance

$$P_{i-1|i-1} = \mathrm{Cov}\left(e_{i-1|i-1}\right). \tag{64}$$

By the orthogonality principle, the estimation error $e_{i-1|i-1}$ has zero mean and is uncorrelated with $\hat{f}_{0:i-1}$. Define the prediction as

$$m_{i|i-1} = A_i m_{i-1|i-1} + b_i, \tag{65}$$

and define the prediction error and innovation by

$$e_{i|i-1} = s_i - m_{i|i-1}, \qquad r_i = \hat{f}_i - \left(H_i m_{i|i-1} + \xi_i\right). \tag{66}$$

Since

$$e_{i|i-1} = A_i e_{i-1|i-1} + G\eta_i, \tag{67}$$

the assumptions imply that $e_{i|i-1}$ has zero mean and is uncorrelated with the past observations $\hat{f}_{0:i-1}$. Hence, $m_{i|i-1}$ is the LMMSE predictor of s_i based on $\hat{f}_{0:i-1}$. Its error covariance is

$$P_{i|i-1} = \mathrm{Cov}\left(e_{i|i-1}\right) = A_i P_{i-1|i-1} A_i^\top + G Q_i G^\top. \tag{68}$$

Additionally, using

$$r_i = H_i e_{i|i-1} + \varepsilon_i, \tag{69}$$

it follows that r_i also has zero mean and is uncorrelated with the past observations $\hat{f}_{0:i-1}$. To construct the LMMSE estimator of s_i, consider the affine estimator

$$m_{i|i} = m_{i|i-1} + K_i r_i. \tag{70}$$

Its estimation error is

$$s_i - m_{i|i} = e_{i|i-1} - K_i r_i. \tag{71}$$

This error has zero mean and is uncorrelated with the past observations $\hat{f}_{0:i-1}$. By the orthogonality principle, it remains to choose K_i such that this error is uncorrelated with the innovation r_i, resulting in

$$\mathrm{Cov}\left(e_{i|i-1} - K_i r_i,\, r_i\right) = 0, \tag{72}$$

or equivalently,

$$K_i S_i = \mathrm{Cov}\left(e_{i|i-1}, r_i\right) = P_{i|i-1} H_i^\top, \tag{73}$$

where

$$S_i = \mathrm{Cov}(r_i) = H_i P_{i|i-1} H_i^\top + R_i \tag{74}$$

denotes the innovation covariance. Thus, the optimal gain is

$$K_i = P_{i|i-1} H_i^\top S_i^{-1}, \tag{75}$$

which is exactly the Kalman gain used in the recursive update. The corresponding estimation error covariance is

$$\begin{aligned} P_{i|i} &= \mathrm{Cov}\left(e_{i|i-1} - K_i r_i\right) \\ &= P_{i|i-1} - K_i H_i P_{i|i-1} - P_{i|i-1} H_i^\top K_i^\top + K_i S_i K_i^\top \\ &= P_{i|i-1} - K_i S_i K_i^\top. \end{aligned} \tag{76}$$

Consequently, given the previous LMMSE estimate, the Kalman update is the LMMSE estimator based on the observation history $\hat{f}_{0:i}$. Applying this argument recursively over the denoising steps establishes the LMMSE optimality of the complete recursion.

C Calibration of Statistics

This appendix describes the calibration granularity and the estimation of the observation and process statistics used by QuAKE.

Granularity. Following the channel-wise calibration strategy commonly used in quantized diffusion correction methods (He et al., 2023; Yao et al., 2024; Liu et al., 2025; Ryu et al., 2026), we estimate all statistics separately for each sampling step and each latent channel. For each timestep-channel pair, all spatial positions are aggregated to estimate one set of calibrated statistics. This aggregation reduces the amount of calibration data required for stable estimation. It corresponds to assuming that elements at the same timestep and channel share the same prior, observation, and process-noise parameters. During sampling, these calibrated statistics are broadcast to all spatial positions, while the correction itself is applied independently to each element. This element-wise correction avoids estimating cross-channel or cross-spatial covariance terms and keeps the online filtering cost small. For a fair comparison, the previous sampling-stage methods use the same timestep-wise and channel-wise calibration granularity.

Observation statistics. We collect calibration statistics along full-precision sampling trajectories. At each calibration step, we run both the full-precision denoiser and the quantized denoiser on the same full-precision latent state, obtaining paired outputs f_i and $\hat{f}_i$. For each timestep and channel, all expectations, variances, and covariances are computed over the calibration samples and spatial positions. The valid current-output entry of the initial prior $m_{0|-1}$ and $P_{0|-1}$ is initialized from the marginal mean and variance of the first full-precision denoiser output. For later observation updates, QuAKE uses the conditional Gaussian model

$$\hat{f}_i \mid f_i \sim \mathcal{N}(\gamma_i f_i + \xi_i, R_i). \tag{77}$$

The observation gain and bias are estimated by linear regression from f_i to $\hat{f}_i$:

$$\gamma_i = \frac{\mathrm{Cov}(f_i, \hat{f}_i)}{\mathrm{Var}(f_i)}, \qquad \xi_i = \mathbb{E}[\hat{f}_i] - \gamma_i \mathbb{E}[f_i]. \tag{78}$$

The residual variance is estimated as

$$R_i = \mathrm{Var}\left(\hat{f}_i - \gamma_i f_i - \xi_i\right). \tag{79}$$

Process statistics. The process statistics are calibrated on the full-precision sampling trajectories. For each step with available history, we apply the transition predictor used by QuAKE to the historical full-precision denoiser outputs and compare the prediction with the actual full-precision output at the current step. The difference is treated as the process residual w_i. For each timestep and channel, we estimate

$$\mu_i = \mathbb{E}[w_i], \qquad Q_i = \mathrm{Var}(w_i). \tag{80}$$

These statistics provide the process bias and process-noise variance used in the Kalman prediction step.

D Implementation Details

This appendix provides additional implementation details for QuAKE, focusing on how the linear transition is constructed and how the output window is handled at the beginning of the sampling trajectory.

Transition coefficients. We construct the linear Gaussian transition through an x_0-domain extrapolation. Empirically, denoiser trajectories are smoother after converting model outputs to the predicted x_0 parameterization and using the log-SNR coordinate λ as the extrapolation variable (Lu et al., 2022; 2025). This leads to smaller process residuals in practice for QuAKE. Therefore, at each step, we first convert the available historical model outputs to the corresponding predicted x_0 values, perform Lagrange extrapolation in the λ coordinate, and then convert the extrapolated x_0 back to the model-output parameterization required by the sampler. This construction remains compatible with the linear Gaussian state-space model used by our method. For fixed sample states and noise levels, the conversion between the model-output parameterization and the predicted x_0 parameterization is affine. Since Lagrange extrapolation is linear in the available historical values, the whole procedure is an affine function of the previous model outputs. Therefore, it can be absorbed into the transition matrix and bias term, and the process model still takes the form

$$s_i = A_i s_{i-1} + b_i + G\eta_i, \tag{81}$$

where the first row of A_i and the corresponding entry of b_i are induced by the composed mapping from model outputs to the x_0-domain, the λ-domain Lagrange extrapolation, and the mapping from the extrapolated x_0 back to the model-output domain. The remaining rows of A_i implement the standard history-shift operation, so that the corrected current output becomes a historical entry in the next step.

Boundary handling. At the first sampling step, there is no previous denoiser output to propagate. We therefore skip the transition prediction and directly apply the observation update to the initial prior. The posterior state after this step contains only the corrected current output, and historical entries that do not correspond to valid denoiser outputs are not returned to the sampler. During the

Table 2: Quantitative results on MJHQ-30K and sDCI using the UniPC sampler. KID values are reported after multiplication by 10^3. IR stands for ImageReward. Boldface marks the best result among sampling-stage correction methods for each model.

Model	Method	MJHQ-30K					sDCI				
		Similarity (↓)		Quality (↑)			Similarity (↓)		Quality (↑)		
		FID	KID	CLIPScore	CLIP-IQA	IR	FID	KID	CLIPScore	CLIP-IQA	IR
FLUX.1-dev	Full-Precision	–	–	0.3186	0.9019	0.9029	–	–	0.3123	0.8473	0.8539
	Quantized	7.45	0.264	0.3172	0.9078	0.8874	7.38	0.316	0.3115	0.8544	0.8541
	Bias Correction	7.35	0.294	0.3172	0.9070	0.8843	7.34	0.346	0.3118	0.8529	0.8439
	TAC-Diffusion	7.39	0.285	0.3172	**0.9081**	0.8848	7.47	0.358	**0.3126**	**0.8555**	**0.8527**
	Q-Drift	7.17	0.197	**0.3175**	0.9077	**0.8853**	7.14	**0.168**	0.3112	0.8542	0.8473
	QuAKE (Ours)	**7.05**	**0.130**	0.3173	0.9068	0.8797	**7.06**	0.180	0.3120	0.8548	0.8452
Sana-0.6B	Full-Precision	–	–	0.3372	0.9019	1.0923	–	–	0.3296	0.9145	0.9042
	Quantized	10.63	1.575	0.3355	0.8950	1.0459	10.70	1.636	0.3275	0.9041	0.8604
	Bias Correction	23.10	8.128	0.3319	0.8761	0.9533	23.60	9.313	0.3228	0.8724	0.7448
	TAC-Diffusion	22.72	7.954	0.3318	0.8773	0.9577	23.39	9.051	0.3223	0.8703	0.7385
	Q-Drift	9.66	1.033	0.3357	0.8897	1.0351	10.00	1.283	0.3281	0.9018	0.8350
	QuAKE (Ours)	**8.12**	**0.364**	**0.3363**	**0.8964**	**1.0523**	**8.31**	**0.409**	**0.3287**	**0.9072**	**0.8419**
PixArt-α	Full-Precision	–	–	0.3301	0.8905	0.9962	–	–	0.3141	0.9141	0.5903
	Quantized	16.26	3.273	0.3279	0.8956	0.9145	16.07	3.739	0.3123	0.9127	0.4917
	Bias Correction	29.44	10.088	0.3279	**0.9029**	0.8977	30.87	12.498	0.3154	**0.9214**	0.4491
	TAC-Diffusion	29.09	10.016	0.3280	0.9008	**0.9037**	30.31	12.323	**0.3156**	0.9145	0.4455
	Q-Drift	13.77	2.183	0.3280	0.8851	0.8896	13.66	2.493	0.3114	0.9020	0.4583
	QuAKE (Ours)	**13.38**	**1.918**	**0.3282**	0.8952	0.9026	**13.49**	**2.337**	0.3126	0.9162	**0.4885**
PixArt-Σ	Full-Precision	–	–	0.3318	0.9065	0.9571	–	–	0.3238	0.9151	0.8587
	Quantized	17.65	4.252	0.3299	0.8988	0.8669	17.22	4.831	0.3289	0.8883	0.7970
	Bias Correction	25.08	7.769	0.3294	0.8872	0.8746	24.55	9.013	0.3293	0.8716	0.7771
	TAC-Diffusion	24.80	7.622	0.3300	0.8900	0.8628	24.22	8.788	**0.3313**	0.8763	0.7726
	Q-Drift	17.21	4.073	0.3296	0.8939	0.8518	16.69	4.459	0.3293	0.8781	0.7895
	QuAKE (Ours)	**15.82**	**3.270**	**0.3303**	**0.9021**	**0.8868**	**15.44**	**3.859**	0.3291	**0.8934**	**0.8030**

first few sampling steps, the number of available historical outputs can be smaller than the state window length. In this case, QuAKE uses the same x_0-domain extrapolation procedure with only the available history. Equivalently, the transition is formed with a reduced effective order, while the state is kept in the same augmented format for implementation consistency. After each update, the sampler reads only the valid posterior entries required at the current step. Once enough historical outputs are available, QuAKE uses the full-order Lagrange extrapolation in the λ coordinate.

E Detailed Experimental Setup

This appendix provides additional details on the model configurations, quantization settings, evaluation metrics, and implementation environment used in our experiments. Unless otherwise specified, all models are evaluated with their default inference configurations. For text-to-image sampling, we use a resolution of 1024×1024 and a classifier-free guidance scale of 4.5 for PixArt-α, PixArt-Σ, and Sana-0.6B, while FLUX.1-dev is evaluated at a resolution of 512×512 with a guidance scale of 3.5. For quantized models, we only quantize the main diffusion backbone, while keeping the VAE, text encoder, and other auxiliary components in their original precision. For the SVDQuant model-quantization hyperparameter r, we set $r = 32$ for PixArt-α and PixArt-Σ, $r = 16$ for FLUX.1-dev, and $r = 4$ for Sana-0.6B. For evaluation, FID and KID are computed using `clean-fid` (Parmar et al., 2022). CLIP-based evaluation metrics are computed with the CLIP-ViT-B/32 backbone (Radford et al., 2021). All experiments are conducted on NVIDIA A100-SXM4-40GB GPUs.



Prompt: *A captivating birdseye view of a massive rocket taking off from Earth, ascending towards space with a trail of smoke and fire, rendered with Unreal Engine 5 for unparalleled detail, vivid colors, and cinematic lighting.*

(a) FLUX.1-dev



Prompt: *high resolution rendering with depth of field of a minimalist future multi storeyed building in kochi city kerala*

(b) Sana-0.6B



Prompt: *Florida portrait by William Whitaker*

(c) PixArt-α



Prompt: *There can be 366 birthdays in a year, including February 29, and at least 367 people are needed to have the same birthday for two people.*

(d) PixArt-Σ

Figure 6: Additional qualitative comparison across different text-to-image diffusion models using UniPC.

F Additional Results

This appendix provides additional quantitative and qualitative results to further evaluate the generalization of QuAKE across different numerical samplers. Table 2 reports results using the UniPC sampler. QuAKE consistently provides the strongest overall distributional recovery across different models and datasets, substantially reducing the FID and KID gap to the full-precision generation while maintaining competitive generation quality. These results closely follow the trend observed with DPM-Solver++, demonstrating that the effectiveness of QuAKE is not tied to a specific numerical solver and further supporting its applicability to different high-order multistep ODE samplers.

Figure 6 provides additional qualitative comparisons. Consistent with the quantitative results, QuAKE effectively compensates for quantization-induced generation deviations across different model architectures and produces outputs that more closely resemble the corresponding full-precision generations than existing sampling-stage correction methods.