Q-SCHED: PUSHING THE BOUNDARIES OF FEW-STEP DIFFUSION MODELS WITH QUANTIZATION-AWARE SCHEDULING

Natalia Frumkin & Diana Marculescu
The Chandra Family Department of Electrical and Computer Engineering
The University of Texas at Austin
{nfrumkin,dianam}@utexas.edu

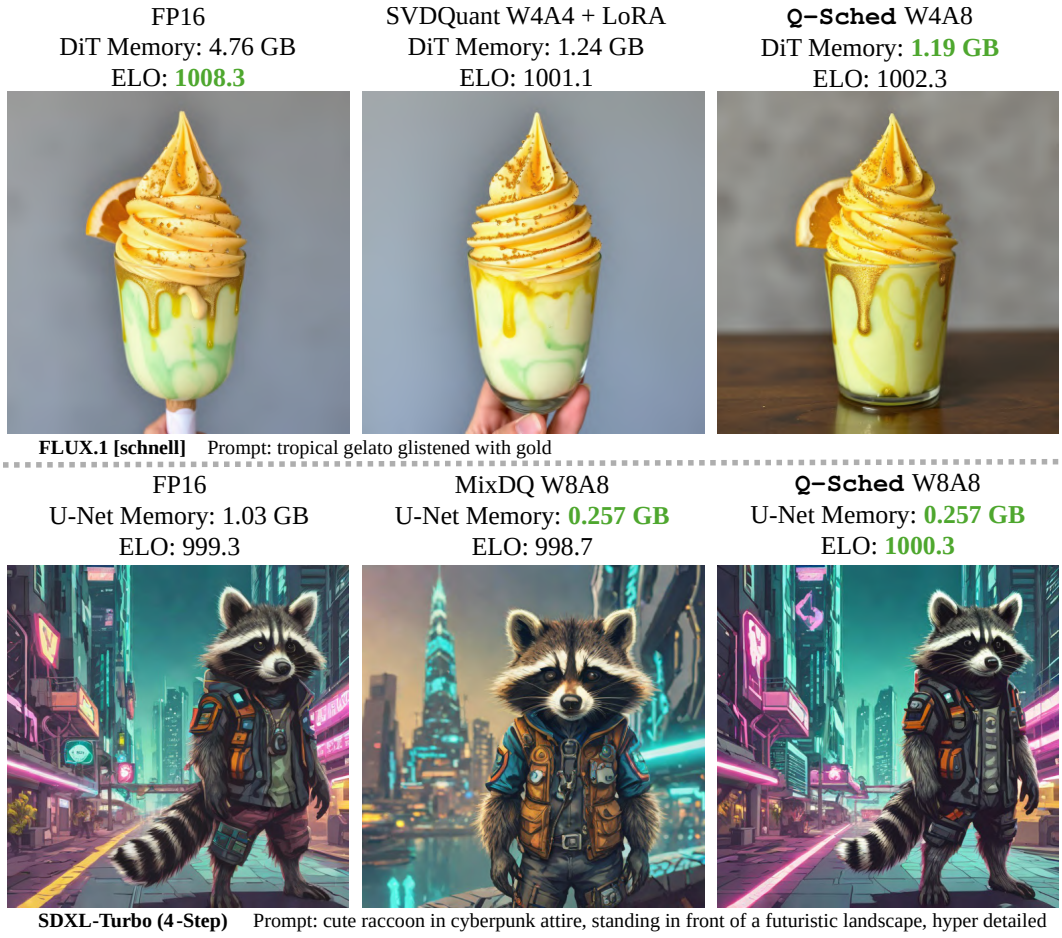


Figure 1: When large diffusion models are reduced to W8A8 or W4A8 for deployment, image fidelity drops. `Q-Sched` applies scheduler-level tuning, just two coefficients per step, to steer the sampler back to FP16-like quality, with no new checkpoints, no finetuning, and no extra FLOPs.

ABSTRACT

Text-to-image diffusion models remain computationally intensive: generating a single image typically requires dozens of passes through large transformer backbones (*e.g.* , SDXL uses ~ 50 evaluations of a 2.6B-parameter model). Few-step variants reduce the step count to 2–8 but still rely on large, full-precision backbones, making inference impractical on resource-constrained platforms, both on-device (latency/energy) and in data centers with multi-instance GPU (MIG) style GPU partitioning (limited memory/throughput per slice). Existing post-training quantization (PTQ) methods are further hampered by dependence on full-precision calibration.

We introduce `Q-Sched`, a scheduler-level PTQ approach that adapts the diffusion sampler while keeping quantized weights fixed. By adjusting the few-step sampling trajectory with quantization-aware preconditioning coefficients, `Q-Sched` matches or surpasses full-precision quality while delivering a $4\times$ reduction in model size and preserving a reusable checkpoint across bit-widths. To learn these coefficients, we propose a reference-free Joint Alignment–Quality (JAQ) loss, which combines text–image compatibility with an image-quality objective for fine-grained control; JAQ requires only a handful of calibration prompts and avoids full-precision inference during calibration.

Empirically, `Q-Sched` yields substantial gains: a **15.5%** FID improvement over the FP16 4-step Latent Consistency Model and a **16.6%** improvement over the FP16 8-step Phased Consistency Model, demonstrating that quantization and few-step distillation are complementary for high-fidelity generation. A large-scale user study with $\mathbf{80{,}000}+$ annotations further validates these results on both FLUX.1[schnell] and SDXL-Turbo. Our code is available at https://github.com/enyac-group/q-sched.

1 INTRODUCTION

Diffusion models have firmly established state-of-the-art performance across modalities, delivering unprecedented fidelity in image synthesis (Esser et al., 2024; Dai et al., 2023), video generation (Brooks et al., 2024; Polyak et al., 2024), and audio modeling (Liu et al., 2024). Beyond creative media, they have become indispensable in scientific discovery, driving breakthroughs in biomolecular structure prediction (Abramson et al., 2024; Watson et al., 2023). However, these capabilities come with significant computational overhead. Leading foundation models such as Stable Diffusion 3 (Esser et al., 2024) and Sora (Brooks et al., 2024) rely on massive Diffusion Transformer (DiT) backbones (Peebles & Xie, 2023), where inference necessitates iterative denoising through tens of billions of parameters, making real-time deployment prohibitively expensive.

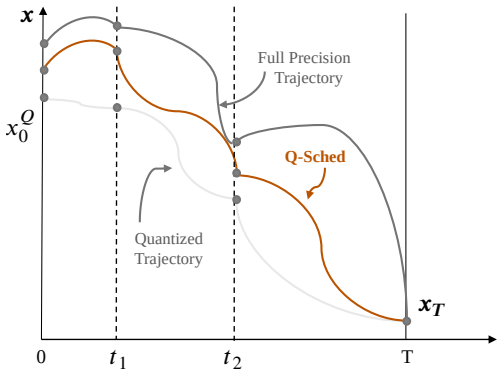
Practical deployment therefore hinges on two levers: (1) reducing the number of function evaluations (few-step sampling), and (2) lowering the cost per evaluation (compression via quantization (He et al., 2024; Guo et al., 2022), pruning (Fang et al., 2024), or distillation (Huang et al., 2024)). These levers are particularly important in two widely used settings. *On-device*, memory and compute budgets are tight, latency and energy constraints are strict, and privacy/offline use cases preclude server offloading (Zhao et al., 2024b). *In data centers with MIG partitioning*, a single GPU is sliced into multiple smaller instances to increase concurrency and predictability; each slice has limited memory/throughput, making model footprint and per-step cost decisive (Zhang et al., 2023; Li et al., 2022). In both cases, few-step sampling and quantization are natural, complementary choices.

However, few-step acceleration is sensitive to the accuracy of the underlying probability-flow ordinary differential equation (ODE) or variance-preserving stochastic differential equation (SDE) that links the noise-estimation network to the final sample (Song et al., 2021). Quantization perturbs that network, inducing a mismatch that alters the ODE/SDE trajectory and amplifies artifacts, an effect that becomes more pronounced as the number of steps shrinks. Simply reusing full-precision schedulers on quantized backbones will inevitably induce quality degradation.

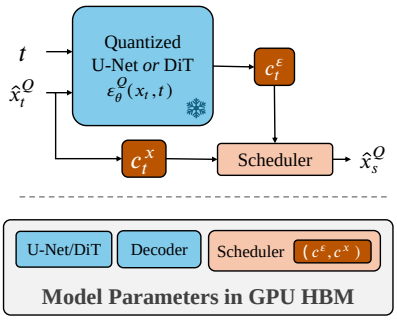
To bridge this gap, we introduce **`Q-Sched`**, a quantization-aware noise scheduler that adapts the few-step trajectory to the compressed model *while keeping quantized weights fixed*. Instead of correcting the model weights, `Q-Sched` inserts lightweight coefficients $(\mathbf{c}^{\mathbf{x}}, \mathbf{c}^{\epsilon})$ into the scheduler (Figures 2a and 2b), correcting quantization-induced drift while keeping a single U-Net/DiT checkpoint reusable across multiple precision settings. This design directly targets the constraints above: it preserves the latency benefits of few-step sampling, fits within on-device and MIG memory budgets, and avoids checkpoint sprawl in production.

Our contributions are summarized as follows:

1. In this work, we introduce **`Q-Sched`**, a quantization-aware scheduler that integrates seamlessly with few-step diffusion models. It achieves up to a **15.5% FID improvement** over a 4-step latent consistency model (LCM) (Luo et al., 2023) baseline and, as shown in Figure 1, can match or surpass full-precision arena scores *while simultaneously reducing model size*



(a) Quantization shifts the diffusion sampling trajectory, reducing image quality and departing from the full-precision model's sampling trajectory. `Q-Sched` corrects this drift by adapting the sampling scheduler.



(b) At timestep $t \to s$, `Q-Sched` applies lightweight scheduler coefficients (c_t^x, c_t^ϵ) to enable quantized inference from a single U-Net or DiT checkpoint, leaving model weights fixed and avoiding retraining or post-training modifications while supporting multiple precision settings.

Figure 2: **Overview of `Q-Sched`.** Quantization-aware scheduling corrects trajectory drift introduced by low-precision sampling while preserving a single pretrained model checkpoint.

on SDXL-Turbo (4-Step) (Sauer et al., 2024) and FLUX.1[schnell] (Black Forest Labs, 2024).

2. `Q-Sched`'s novel **preconditioning coefficients** enable quantized models to deliberately deviate from potentially overfit few-step baselines (Figure 2a), alleviating oversmoothing and texture artifacts from distillation and quantization while improving the balance between fidelity and artifact severity.
3. To optimize these coefficients, we propose the **Joint Alignment–Quality (JAQ) loss** which balances perceptual fidelity with text–image alignment. Being reference-free, JAQ also enables precise control over visual properties (*e.g.*, texture, detail, saturation) without requiring access to a full-precision model.
4. We establish a **theoretical existence guarantee** (Theorem 1), proving that there exists `Q-Sched` coefficients which reduce expected sampling error relative to the original quantized scheduler. This provides a principled explanation for `Q-Sched` 's systematic improvements.
5. Finally, a large-scale **human preference study** with over 80,000 annotations demonstrates that `Q-Sched` outperforms MixDQ (Zhao et al., 2024a) on SDXL-Turbo and SVDQuant (Li et al., 2025) on FLUX.1[schnell] in terms of perceived image quality.

As illustrated in Figure 1, `Q-Sched` attains the highest ELO rating in pairwise image-quality comparisons among evaluated methods. Furthermore, Figure 3 shows that `Q-Sched` is Pareto-optimal with respect to both ELO and model size, underscoring its ability to balance perceptual quality and efficiency more effectively than competing approaches.

2 Background and Related Work

Diffusion models generate samples by denoising corrupted data across a trajectory of timesteps $t \in [0, T]$, where T is typically large (≥ 25). Each step applies a denoising network $\mathcal{E}_\theta$, conditioned on both t and its noisy input x_t. While iterative denoising yields high-fidelity samples, invoking a large U-Net or DiT backbone across many timesteps makes inference prohibitively slow in deployment.Few-step models are an important distillation strategy to address slow diffusion model inference.

One such few-step model, **SDXL-Turbo**, leverages Adversarial Diffusion Distillation (ADD), combining score distillation with an adversarial loss, to reduce sampling to just 1–4 steps, enabling real-time generation on commodity GPUs (Sauer et al., 2024). More recently, **FLUX.1[schnell]** introduces a 12B-parameter rectified-flow transformer with open weights, optimized for 1-4 step inference, making it attractive for latency-constrained serving (Black Forest Labs, 2024). Most recently, **FLUX.1[kontext]** extends the family beyond text-to-image toward *in-context* generation and editing,

accepting text and images jointly and unifying both tasks in a flow-matching framework (Labs et al., 2025). These advances exemplify the field's shift toward *deployment-ready* diffusion systems that meet strict latency and memory budgets.

Few-step diffusion and distillation. Few-step methods compress the teacher's long trajectory into a handful of evaluations, preserving most of the fidelity at a fraction of the cost. Distillation is the primary approach: early demonstrations distilled long-run teachers into 1–8 step students, such as Instaflow (Liu et al., 2023), rectified-flow straightening (Liu et al., 2022), and adversarially guided ADD (Sauer et al., 2024). Consistency Models (CMs) (Song et al., 2023) frame generation as a self-consistency mapping from any noisy state to the clean sample, yielding efficient few-step samplers. Variants include Latent Consistency Models (LCMs) (Luo et al., 2023) with Stable Diffusion (Rombach et al., 2022) backbones, Trajectory Consistency Distillation (TCD) (Zheng et al., 2024) with trajectory-aware schedules, and Phased Consistency Models (PCMs) (Wang et al., 2024) with improved guidance and stability. Across these designs, the *scheduler* plays a critical role in determining quality in the few-step regime. The update rule for few-step diffusion models using quantized backbone $\mathcal{E}_\theta^Q$ is: $x_s = \Phi(t, x_t, \mathcal{E}_\theta^Q)$, where x_s denotes the intermediate sample at timestep $s \in [0, t]$ and $\Phi(\cdot)$ is a few-step scheduler. In Section 3, we illustrate our approach using the TCD scheduler (Zheng et al., 2024) as a running example. However, `Q-Sched` is fully general and can be applied on top of any few-step scheduler that fits the abstraction in Section 2.

Quantization for diffusion models. Post-training quantization (PTQ) for diffusion models has primarily targeted ϵ_t and its activations across timesteps. Timestep-aware calibration methods such as PTQ4DM (Shang et al., 2022), ADP-DM (Wang et al., 2023a), and Q-Diffusion (Li et al., 2023), along with dynamic schemes like TDQ (So et al., 2024) and error-compensation approaches such as Q-DM (Li et al., 2024c), modify weights or activations and require full-precision calibration. MixDQ (Zhao et al., 2024a) extends PTQ to few-step models via mixed-precision allocation guided by BOS-aware quantization and layer sensitivity, while SVDQuant (Li et al., 2025) targets 4-bit weights and activations by absorbing outliers into a high-precision low-rank branch via SVD.

In the few-step regime, quantization bias may additionally manifest as a *scheduler mismatch*, where a fixed full-precision schedule can over- or under-correct and amplify artifacts. PTQD (He et al., 2024) is a related weight-preserving approach that models quantization error as an affine perturbation of the denoiser, $\mathcal{E}_\theta^Q(x_t, t) = (1 + \gamma)\epsilon_t + \delta$, and compensates via variance scaling and a bias term in the sampler; γ is estimated by standard-deviation matching, while δ is assumed to be uncorrelated Gaussian noise. We adapt this bias-correction principle to TCD (see Section J) and extend it to other few-step samplers as a baseline.

In contrast, `Q-Sched` learns correction coefficients end-to-end from final image quality, avoiding Gaussian assumptions and intermediate denoising statistics. By adding a second coefficient on x_t, it decouples accumulated state error from current-step noise, enabling independent correction of both distortions in few-step distilled models and direct optimization of the final output distribution without full-precision activations.

3 Q-SCHED QUANTIZATION-AWARE SCHEDULING

`Q-Sched` reframes quantized few-step generation as a *scheduler adaptation* problem. It learns quantization-aware pre-conditioning coefficients that correct trajectory drift with negligible overhead, leaving the backbone frozen and requiring only lightweight calibration. Unlike prior PTQ approaches that modify weights or activations, `Q-Sched` adapts the scheduler itself, complementing existing PTQ and distillation techniques while retaining the latency advantages of few-step sampling.

To prepare the TCD scheduler for optimization with `Q-Sched`, consider sampling with a quantized network. TCD's Strategic Stochastic Sampling (SSS) (Zheng et al., 2024) using a quantized network $\mathcal{E}_\theta^Q(x_t, t)$ is given by:

$$\mathbf{x_s} = \frac{\alpha_s}{\alpha_{s'}} \left(\alpha_{s'} \frac{\mathbf{x_t} - \sigma_t \mathcal{E}_\theta^Q(x_t, t)}{\alpha_t} + \sigma_{s'} \mathcal{E}_\theta^Q(x_t, t) \right) + \eta \mathbf{z} \tag{1}$$

where the noise schedule is given by σ, α, and the sampler injects stochastic noise sampled from the distribution $\mathbf{z} \sim \mathcal{N}(0, I)$. The sampler relies on an intermediary timestep $s' \in [s, t]$ where noise is

Algorithm 1 Search for `Q-Sched` Coefficients

Input: search range $[c_{min}, c_{max}]$, search points n, diffusion steps ω, loss JAQ, calibration set $\mathcal{C}$, optimizer `opt`

$S^* \leftarrow \infty$
$(c^x_{start}, c^x_{end}, c^\epsilon_{start}, c^\epsilon_{end}) \leftarrow$ `opt.init`(c_{min}, c_{max}) ▷ uniform init in (c_{min}, c_{max})
for $i = 0$ to n **do**
  $\mathbf{c^x} \leftarrow$ `linspace`$(c^x_{start}, c^x_{end}, \omega)$, $\mathbf{c}^\epsilon \leftarrow$ `linspace`$(c^\epsilon_{start}, c^\epsilon_{end}, \omega)$
  $S \leftarrow \{\, \text{JAQ}(x; \mathbf{c^x}, \mathbf{c}^\epsilon) \mid x \in \mathcal{C} \,\}$
  if $\bar{S} < S^*$ **then** ▷ $\bar{S}$ is the mean of S
    $S^* \leftarrow \bar{S}$, $\mathbf{c^x_\star} \leftarrow \mathbf{c^x}$, $\mathbf{c}^\epsilon_\star \leftarrow \mathbf{c}^\epsilon$
  end if
  $(c^x_{start}, c^x_{end}, c^\epsilon_{start}, c^\epsilon_{end}) \leftarrow$ `opt.step`$(\bar{S})$
end for
return $\mathbf{c^x_\star}, \mathbf{c}^\epsilon_\star$

added. The degree of randomness is defined by the stochastic control parameter $\eta = \sqrt{1 - \alpha_s^2/\alpha_{s'}^2}$ which can be adjusted at sampling time to vary image randomness. The TCD sampler in Equation (1), used in Phased Consistency Models, is a state-of-the-art few-step diffusion method that depends on two inputs from the previous step, x_t and $\mathcal{E}^Q_\theta(x_t, t)$, which are central to applying `Q-Sched`.

A Learnable Schedule Pre-Conditioner To adapt the noise schedule of few-step diffusion models, `Q-Sched` applies two learnable scalar preconditioning coefficients, c^x_t and c^ϵ_t, applied respectively to x_t and $\mathcal{E}^Q_\theta(x_t, t)$ at time t. As illustrated in Figure 2b, `Q-Sched` operates independently of the model backbone (U-Net or transformer), making it broadly compatible with any few-step scheduler resembling TCD as shown in Equation (2). Under `Q-Sched`, the TCD sampling update becomes:

$$\mathbf{x}_s = \frac{\alpha_s}{\alpha_{s'}} \left(\alpha_{s'} \frac{c^x_t \mathbf{x}_t - \sigma_t c^\epsilon_t \mathcal{E}^Q_\theta(x_t, t)}{\alpha_t} + \sigma_{s'} c^\epsilon_t \mathcal{E}^Q_\theta(x_t, t) \right) + \sqrt{1 - \frac{\alpha_s^2}{\alpha_{s'}^2}}\, \mathbf{z}. \tag{2}$$

Because the update rule is affine in the preconditioning coefficients $(\mathbf{c^x}, \mathbf{c}^\epsilon) := (c^x_t, c^\epsilon_t)_{t=0}^T$, they can be fused into the existing TCD schedule without modifying the computational graph or incurring inference overhead. In TCD, the estimate at the proxy timestep s' is derived directly from the prediction at timestep t, so the corresponding noise term naturally shares the same coefficient c^ϵ_t.

We learn $(\mathbf{c^x}, \mathbf{c}^\epsilon)$ via a targeted hyperparameter search (Algorithm 1) using grid search and Nevergrad's `NGOpt` Rapin & Teytaud (2018). The search space is deliberately low dimensional, involving four coefficients per timestep over 2–8 steps, enabling efficient and robust optimization. In practice, grid search deterministically explores this space and yields strong, consistent performance across models and schedules; even small coefficient adjustments noticeably improve sharpness and reduce quantization artifacts, underscoring the importance of scheduler-level calibration in few-step diffusion. Additional details are provided in Section O.

A natural question is **why `Q-Sched` coefficients suffice to improve image quality**: the reconstruction error between full-precision and quantized outputs at $t = 0$, denoted Δx_0, can be strictly reduced through scheduler-level coefficient adjustment.

Theorem 1 (Strict Existence Guarantees)**.** *There exists `Q-Sched` coefficients* $(\mathbf{c^x}, \mathbf{c}^\epsilon) \neq 0$ *such that* $E[||\Delta \tilde{x_0}||] < E[||\Delta x_0||]$.

As shown in the proof in Appendix K, Δx_0 is a linear combination of per-step denoising errors $\Delta E_\theta(t)$ with coefficients k_t, m_t. Since the error is homogeneous in these terms, rescaling via $\tilde{k}_t = c^x_t k_t$ and $\tilde{m}_t = c^\epsilon_t m_t$ strictly reduces the expected error over naïve quantization. Thus, re-weighting the sampler, without modifying network weights, guarantees a reduction in error with respect to the full precision images. Next, we will discuss our reference-free metric, JAQ, and its advantages over existing PTQ loss objectives.

JAQ: A Joint Alignment Quality Loss Function Because full-precision intermediate states become unstable targets under quantization, due to structural and semantic drift, **optimizing for downstream image quality** provides a more reliable objective than attempting to match the full-precision trajectory.

Table 1: Comparison of scheduling strategies for Phased and Latent Consistency Models (PCMs/LCMs) on a Stable Diffusion v1.5 backbone, including original schedules (TCD (Zheng et al., 2024) and multi-step consistency sampling (Luo et al., 2023)), PTQD (He et al., 2024), and `Q-Sched`. FID and CLIPScore are evaluated on COCO-30k; NFEs denote the number of network passes $\mathcal{E}_\theta^Q(x_t, t)$, and latency (ms) is measured on an RTX A6000.

NFEs	Precision	Schedule	Latency (ms)	PCMs FID ↓	PCMs CLIPScore ↑	LCMs FID ↓	LCMs CLIPScore ↑
2	FP16	Original	148	24.17	<u>25.489</u>	38.74	<u>25.155</u>
	W4A8	Original	136	28.70	25.343	40.93	24.886
	W4A8	PTQD	137	<u>23.33</u>	25.265	<u>37.59</u>	24.919
	W4A8	`Q-Sched`	136	**22.24**	**25.543**	**32.50**	**25.152**
4	FP16	Original	193	23.29	25.482	<u>31.94</u>	**25.969**
	W4A8	Original	172	23.08	25.557	38.41	<u>25.456</u>
	W4A8	PTQD	172	<u>19.42</u>	<u>25.639</u>	39.72	24.678
	W4A8	`Q-Sched`	172	**17.39**	**25.715**	**26.98**	25.336
8	FP16	Original	286	20.15	**25.714**	<u>27.34</u>	**26.052**
	W4A8	Original	245	18.48	<u>25.664</u>	27.55	<u>25.397</u>
	W4A8	PTQD	246	**15.85**	25.770	28.06	25.241
	W4A8	`Q-Sched`	245	<u>16.83</u>	25.698	**25.82**	25.214

Reference-free metrics such as CLIPScore (Hessel et al., 2021) are therefore particularly well suited to this setting. Unlike comparative metrics such as FID (Heusel et al., 2017) or SSIM (Wang et al., 2004), reference-free metrics do not rely on a ground-truth reference image. Under quantization, the generated image $\hat{x}_0^Q$ follows an altered sampling trajectory (Figure A3), often yielding a different and sometimes cleaner image than its full-precision counterpart. As a result, reference-based metrics fail to capture fine-grained visual differences introduced by quantization.

To better balance prompt fidelity and visual quality, we introduce the **Joint Alignment Quality (JAQ) loss**, which combines a text-to-image compatibility metric $\texttt{TC}(x)$ with a pure image quality metric $\texttt{IQ}(x)$. Unlike optimizing CLIPScore or CLIP-IQA (Wang et al., 2023b) alone, JAQ better discriminates between highly similar images, a regime where standard metrics are less informative. We define

$$\texttt{JAQ}(x) = \texttt{TC}(x) + k \cdot \texttt{IQ}(x), \tag{3}$$

where k controls the tradeoff between semantic alignment and visual detail. Optimizing only for text-to-image compatibility tends to sacrifice fine detail and overlook quantization artifacts (Figures A2 and A4), while image-only objectives can introduce extra details. JAQ balances these effects.

Applying `Q-Sched` to full-precision models? While `Q-Sched` could in principle be applied to full-precision models, it is not designed for this setting, as full-precision models are better optimized through training-time objectives such as distillation or fine-tuning. Instead, `Q-Sched` specifically targets quantization-induced degradation by correcting drift in the sampling trajectory.

4 EXPERIMENTS

Setup We apply `Q-Sched` across few-step diffusion models, including U-Net (Ronneberger et al., 2015) and DiT (Peebles & Xie, 2023) backbones, and across various distillation strategies: consistency-based (LCM (Luo et al., 2023), PCM (Wang et al., 2024)) and flow-matching approaches (SDXL-Turbo (Sauer et al., 2024) and FLUX.1[schnell] (Black Forest Labs, 2024)). We quantize models in both 4-bit weights, 8-bit activations (W4A8) and 8-bit weights, 8-bit activations (W8A8). Only the U-Net or DiT backbone is quantized, as it dominates model size (see Table A5). Latency is measured on an Nvidia RTX A6000 GPU with Ampere compute architecture. Using BitsandBytes Dettmers et al. (2025), we quantize each model to 4-bit weights, 8-bit activations and average latency over 10 runs with a 3-run warmup.

LCM and PCM are tested at 2, 4, and 8 steps on COCO-30k (Lin et al., 2014), using FID (*vs.* real), `CLIPScore` (prompt alignment), and FID-SD (*vs.* Stable Diffusion). FLUX.1 and SDXL-Turbo

Table 2: Comparison across image quality metrics. "MixDQ" refers to the W8A8 MixDQ (Zhao et al., 2024a) variant and "SVDQ" refers to LoRA-based W4A4 SVDQuant Li et al. (2025).

	SDXL-Turbo (4-Step)				FLUX.1 [schnell]			
	FP16	W8A8	MixDQ	`Q-Sched`	FP16	W4A8	SVDQ	W4A8 `Q-Sched`
CLIP Score ↑	**25.62**	**25.62**	25.38	25.36	**25.61**	25.17	25.52	25.27
CLIP IQA ↑	0.725	0.727	0.727	**0.731**	**0.716**	0.712	0.714	0.707
HPV2 ↑	0.276	0.276	0.275	**0.278**	**0.275**	0.274	**0.275**	0.272
AQ-MAP ↑	0.693	0.694	0.693	**0.696**	**0.700**	**0.700**	0.697	**0.700**
Pick Score ↑	18.48	18.49	18.48	**18.51**	18.43	18.42	18.40	**18.46**
MANIQA ↑	0.508	0.513	0.502	**0.511**	**0.528**	0.500	0.514	0.506
JAQ (ours) ↑	1.663	1.665	1.659	**1.669**	**1.676**	1.675	1.669	1.673

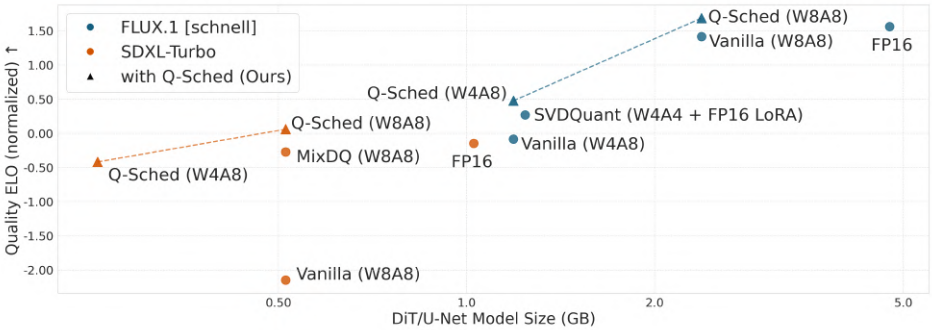


Figure 3: ELO Score *vs.* Model Size for various quantization methods on FLUX.1[schnell] (Black Forest Labs, 2024) and SDXL-Turbo (Sauer et al., 2024).

are evaluated on the SVDQuant (Li et al., 2025; 2024b) subset of MJHQ-30k (5,000 high-quality Midjourney prompts in 10 categories), using FID and human preference to capture perceptual quality.

We employ two variants of the Joint Alignment Quality (JAQ) loss: one derived from CLIP-based metrics and another from human preference scores. In the CLIP-based variant, we set `TC(x)` = `CLIPScore(x)` and `IQ(x)` = `CLIP-IQA(x)`. For SDXL-Turbo and FLUX.1, we instead adopt a preference-based variant, with `TC(x)` = `AQ-MAP(x)` and `IQ(x)` = `HPSV2(x)`. Here, AQ-MAP (Li et al., 2024a) provides a spatial alignment score, while HPSV2 (Wu et al., 2023) is fine-tuned on real human judgments. In both cases, we fix $k = 2$.

Latent and Phased Consistency Model Results In Table 1, we evaluate three schedulers across two consistency model families and show that `Q-Sched` learns an improved few-step trajectory that mitigates artifacts, achieves strong FID, and outperforms PTQD in 4/6 Stable Diffusion v1-5 variants while using only a fraction of the calibration data. Unlike PTQD (He et al., 2024), which relies on a 1,024-image full-precision calibration set, `Q-Sched` requires only 20 long-form sDCI prompts (Li et al., 2025), reused across evaluations and without full-precision references. By optimizing only a small number of coefficients on a highly descriptive calibration set, `Q-Sched` mitigates calibration overfitting, generalizes from complex scenes to simpler prompts, and exceeds a full-precision few-step model by **16.1%, 15.5%, and 5.6%** at 2, 4, and 8 steps, respectively, highlighting quantization and few-step distillation as complementary compression strategies. Moreover, both `Q-Sched` and PTQD introduce no additional inference latency, as their coefficients are fused into the sampling schedule. In Table A1, we show that on a 2-step PCM with a Stable Diffusion XL backbone, `Q-Sched` incurs only a 1.2% FID drop under W4A8 quantization, while PTQD degrades sharply due to the breakdown of its Gaussian noise assumption in the few-step setting, particularly for large models.

SDXL-Turbo and FLUX.1[schnell] Results We evaluate `Q-Sched` on FLUX.1[schnell] and SDXL-Turbo using human preference metrics and show that the proposed JAQ loss effectively captures image fidelity. In addition to our primary metrics, we compare against PickScore (Kirstain et al., 2023), which predicts human preferences from large-scale image–text comparisons, and MANIQA (Yang et al., 2022), a reference-free perceptual quality metric based on multi-dimensional attention. As shown in Table 2, JAQ aligns closely with established metrics while uniquely preserving fine-grained

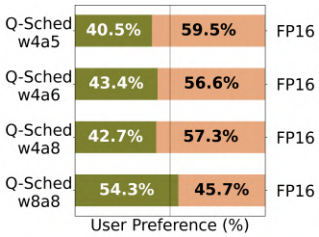


Figure 4: Comparing `Q-Sched` across various quantization bit-widths.

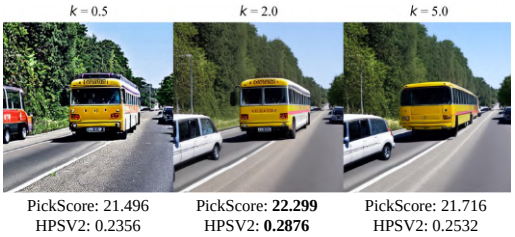


Figure 5: Choice of k for the JAQ loss, balancing $\texttt{TC}(x)$ versus $\texttt{IQ}(x)$ (prompt: "a car and a bus on a French highway").

Table 3: Ablation on pre-conditioning coefficients: jointly optimizing model and sample coefficients yields the best image quality (averaged over 1,024 images from a 4-step SDXL PCM).

Metric	c^ϵ	c^x	(c^ϵ, c^x)
PickScore ↑	21.83	22.25	**22.30**
HPSV2 ↑	**0.288**	0.262	**0.288**
JAQ (ours) ↑	3.367	3.383	**3.392**

Table 4: Comparison of optimization strategies for `Q-Sched` on SDXL-Turbo using Nevergrad optimizers; OPO denotes `OnePlusOne`.

Metric	Grid Search	OPO	`NGOpt`
Pick Score ↑	18.37	18.43	**18.46**
MANIQA ↑	0.434	0.506	0.495
JAQ (ours) ↑	**1.678**	1.673	1.674

details that are often degraded by quantization. In Figure 3, we further report user preference results for `Q-Sched` on SDXL-Turbo and FLUX.1[schnell], demonstrating that `Q-Sched` outperforms MixDQ (Zhao et al., 2024a) and SVDQuant (Li et al., 2025), respectively, at comparable model sizes (see Section D). We aggregate all pairwise 1v1 comparisons into an ELO rating, a relative quality ranking inspired by chess scoring, where higher scores indicate consistent user preference. Finally, in Figure 4, we analyze the effect of bit-width using a user study and find that while W4A4 is overly aggressive, W4A5 and W4A6 produce images comparable to full precision; all 1v1 comparisons against full-precision FLUX.1 follow the protocol in Appendix D.

Ablation on Pre-Conditioning Coefficients We ablate the choice of pre-conditioning coefficients in the Phased Consistency Model by comparing performance when optimizing only the model-side coefficient $\mathbf{c}^\epsilon$, the sample-side coefficient $\mathbf{c}^\mathbf{x}$, or both jointly. As shown in Table 3, jointly optimizing both $\mathbf{c}^\epsilon$ and $\mathbf{c}^\mathbf{x}$ consistently yields the best results across all three metrics: PickScore, HPSv2, and JAQ Loss. These findings highlight the importance of treating both denoising and reconstruction terms as tunable components rather than fixing one a priori. All metrics are averaged over 1024 images generated with the SDXL backbone.

How Do We Choose *k* For The JAQ Loss? We optimize the `Q-Sched` preconditioners using the JAQ loss, which balances image quality and text-image consistency via a tradeoff hyperparameter, k. As shown in Figure 5, small k values can lead to color distortion, while larger values (*e.g.*, $k = 5$) cause outputs to drift from the true data distribution. In such cases, the JAQ loss behaves similarly to CLIP-IQA-Q, which lacks sensitivity to concept alignment. We find that a hand-tuned value of k is sufficient for producing a high-quality noise schedule, and the final results are not highly sensitive to its exact choice. Throughout our experiments, we use $k = 2$.

Choosing an Optimizer for `Q-Sched` We evaluate `Q-Sched` using grid search and Nevergrad Rapin & Teytaud (2018) optimizers (`OnePlusOne`, `NGOpt`), with representative results in Table 4 and full results in Table A3 (Appendix). As reflected by the generic update in line 16 of Algorithm 1, the formulation is optimizer-agnostic. Across methods, performance differences are small and largely within evaluation noise, with no optimizer consistently dominating; while evolutionary and meta-optimization strategies yield modest gains on some perceptual metrics, overall results suggest that `Q-Sched` is not sensitive to the specific optimizer choice.

5 Conclusion

Few-step diffusion models reduce inference cost by distilling large pretrained diffusion models (e.g., Stable Diffusion XL) to 2–8 denoising steps, yielding 5–25× speedups, but they do not reduce model size. `Q-Sched` extends this regime by enabling effective quantization via noise-aware preconditioning coefficients with minimal quality loss. We observe **8.0% and 16.1% FID**

improvements over full-precision baselines for PCMs and LCMs, respectively, and a user study shows that **`Q-Sched` outperforms prior quantization methods on FLUX.1[schnell] and SDXL-Turbo in perceived image quality**. Together, these results indicate that quantization and few-step distillation are complementary, enabling substantial efficiency gains without sacrificing generation quality.

5.0.1 ACKNOWLEDGEMENTS

This work was supported in part by NSF CCF Grant No. 2107085, iMAGiNE - the Intelligent Machine Engineering Consortium at UT Austin, and UT Cockrell School of Engineering Doctoral Fellowships. Human evaluation studies were conducted using Rapidata Rapidata (2025).

6 ETHICS STATEMENT

Model compression broadens the accessibility of AI by enabling large foundation models to run on resource-constrained GPUs. The potential societal consequences of our work are similar to those of prior approaches, as both quantization and few-step diffusion serve as compression methods for text-to-image generative models. Such models can produce synthetic images that may mislead, misrepresent, or cause social harm. We conduct a user preference study on a crowdsourcing platform in which participants worldwide are shown generated content, which, like all synthetic media, carries inherent potential for misuse and harm.

7 LLM USAGE

We made use of large language models (LLMs) to assist in the preparation of this manuscript. LLMs were employed for language polishing, formatting support (e.g., LaTeX macros, algorithm pseudocode, figure/table captions), and iterative feedback on clarity and conciseness of explanations.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

Black Forest Labs. Flux.1-schnell. `https://huggingface.co/black-forest-labs/FLUX.1-schnell`, 2024. Accessed: 2025-05-14.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Technical Report*, 2024. URL `https://openai.com/research/video-generation-models-as-world-simulators`.

Junsong Chen, Shuchen Xue, Yuyang Zhao, Jincheng Yu, Sayak Paul, Junyu Chen, Han Cai, Song Han, and Enze Xie. Sana-sprint: One-step diffusion with continuous-time consistency distillation. *arXiv preprint arXiv:2503.09641*, 2025.

Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Dai, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, et al. Emu: Enhancing image generation models using photogenic needles in a haystack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20359–20369, 2023.

Tim Dettmers et al. bitsandbytes: Accessible large-language models via k-bit quantization for pytorch. `https://github.com/bitsandbytes-foundation/bitsandbytes`, 2025. GitHub repository, accessed 2025-11-30.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Axelrod, Zeitlin Millett, et al. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.03206*, 2024.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Structural pruning for diffusion models. *Advances in neural information processing systems*, 36, 2024.

Cong Guo, Yuxian Qiu, Jingwen Leng, Xiaotian Gao, Chen Zhang, Yunxin Liu, Fan Yang, Yuhao Zhu, and Minyi Guo. Squant: On-the-fly data-free quantization via diagonal hessian approximation. *arXiv preprint arXiv:2202.07471*, 2022.

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, 2024.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 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.

Tao Huang, Yuan Zhang, Mingkai Zheng, Shan You, Fei Wang, Chen Qian, and Chang Xu. Knowledge diffusion for distillation. *Advances in Neural Information Processing Systems*, 36, 2024.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL `https://arxiv.org/abs/2506.15742`.

Baolin Li, Tirthak Patel, Siddharth Samsi, Vijay Gadepally, and Devesh Tiwari. Miso: exploiting multi-instance gpu capability on multi-tenant gpu clusters. In *Proceedings of the 13th Symposium on Cloud Computing*, pp. 173–189, 2022.

Chunyi Li, Haoning Wu, Zicheng Zhang, Hongkun Hao, Kaiwei Zhang, Lei Bai, Xiaohong Liu, Xiongkuo Min, Weisi Lin, and Guangtao Zhai. Q-refine: A perceptual quality refiner for ai-generated image. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2024a.

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, 2024b.

Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Junxian Guo, Xiuyu Li, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. Svdquant: Absorbing outliers by low-rank component for 4-bit diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Xiuyu Li, Long Lian, Yijiang Liu, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. *arXiv preprint arXiv:2302.04304*, 2023.

Yanjing Li, Sheng Xu, Xianbin Cao, Xiao Sun, and Baochang Zhang. Q-dm: An efficient low-bit quantized diffusion model. *Advances in Neural Information Processing Systems*, 36, 2024c.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Haohe Liu, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Qiao Tian, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D Plumbley. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2023.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on image processing*, 21(12):4695–4708, 2012.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

Rapidata. Rapidata: An api that provides fast access to large-scale human evaluations, 2025. URL `https://www.rapidata.ai/`. Accessed: 2025-05-16.

J. Rapin and O. Teytaud. Nevergrad - A gradient-free optimization platform. `https://GitHub.com/FacebookResearch/Nevergrad`, 2018.

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.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pp. 87–103. Springer, 2024.

Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. *arXiv preprint arXiv:2211.15736*, 2022.

Junhyuk So, Jungwon Lee, Daehyun Ahn, Hyungjun Kim, and Eunhyeok Park. Temporal dynamic quantization for diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Changyuan Wang, Ziwei Wang, Xiuwei Xu, Yansong Tang, Jie Zhou, and Jiwen Lu. Towards accurate data-free quantization for diffusion models. *arXiv preprint arXiv:2305.18723*, 2(5), 2023a.

Fu-Yun Wang, Zhaoyang Huang, Alexander William Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, et al. Phased consistency model. *arXiv preprint arXiv:2405.18407*, 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, 2023b.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *CoRR*, 2023.

Sidi Yang, Tianhe Wu, Shuwei Shi, Shanshan Lao, Yuan Gong, Mingdeng Cao, Jiahao Wang, and Yujiu Yang. Maniqa: Multi-dimension attention network for no-reference image quality assessment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1191–1200, 2022.

Huaizheng Zhang, Yuanming Li, Wencong Xiao, Yizheng Huang, Xing Di, Jianxiong Yin, Simon See, Yong Luo, Chiew Tong Lau, and Yang You. Migperf: A comprehensive benchmark for deep learning training and inference workloads on multi-instance gpus. *arXiv preprint arXiv:2301.00407*, 2023.

Tianchen Zhao, Xuefei Ning, Tongcheng Fang, Enshu Liu, Guyue Huang, Zinan Lin, Shengen Yan, Guohao Dai, and Yu Wang. Mixdq: Memory-efficient few-step text-to-image diffusion models with metric-decoupled mixed precision quantization. In *European Conference on Computer Vision*, pp. 285–302. Springer, 2024a.

Yang Zhao, Yanwu Xu, Zhisheng Xiao, Haolin Jia, and Tingbo Hou. Mobilediffusion: Instant text-to-image generation on mobile devices. In *European Conference on Computer Vision*, pp. 225–242. Springer, 2024b.

Jianbin Zheng, Minghui Hu, Zhongyi Fan, Chaoyue Wang, Changxing Ding, Dacheng Tao, and Tat-Jen Cham. Trajectory consistency distillation. *arXiv preprint arXiv:2402.19159*, 2024.

A 2-STEP PHASED CONSISTENCY MODEL ON AN SDXL BACKBONE

Table A1 reports results on a 2-step Phased Consistency Model with a Stable Diffusion XL backbone under aggressive W4A8 quantization. While the original TCD schedule experiences a noticeable quality drop, `Q-Sched` largely preserves full-precision performance, incurring only a 1.2% increase in FID relative to FP16. In contrast, PTQD degrades severely in this setting, with large increases in both FID and FID-SD.

This behavior is consistent with the assumptions underlying each method. PTQD models quantization error as an affine perturbation with Gaussian noise, an approximation that becomes unreliable in few-step diffusion and for large models such as SDXL. By contrast, `Q-Sched` directly learns scheduler-level correction coefficients using final image quality, allowing it to compensate for quantization-induced trajectory drift without relying on intermediate denoising statistics or full-precision references. As a result, `Q-Sched` remains stable under aggressive compression while preserving both perceptual quality and text-image alignment.

Table A1: Comparison on a 2-step Phased Consistency Model using the Stable Diffusion XL backbone. FID-SD (↓) is computed relative to images generated by SDXL using COCO-30k prompts.

Scheduler	Precision	FID ↓	FID-SD ↓	CLIPScore ↑
TCD	FP16	18.65	10.45	26.531
TCD	W4A8	22.70	12.51	26.241
PTQD	W4A8	161.96	176.29	25.910
`Q-Sched`	W4A8	**18.89**	**12.17**	**26.513**

B Q-SCHED IMAGES

In Figure A1, we provide images comparing SVDQuant with `Q-Sched` on a W4A4 quantized FLUX.1[schnell].

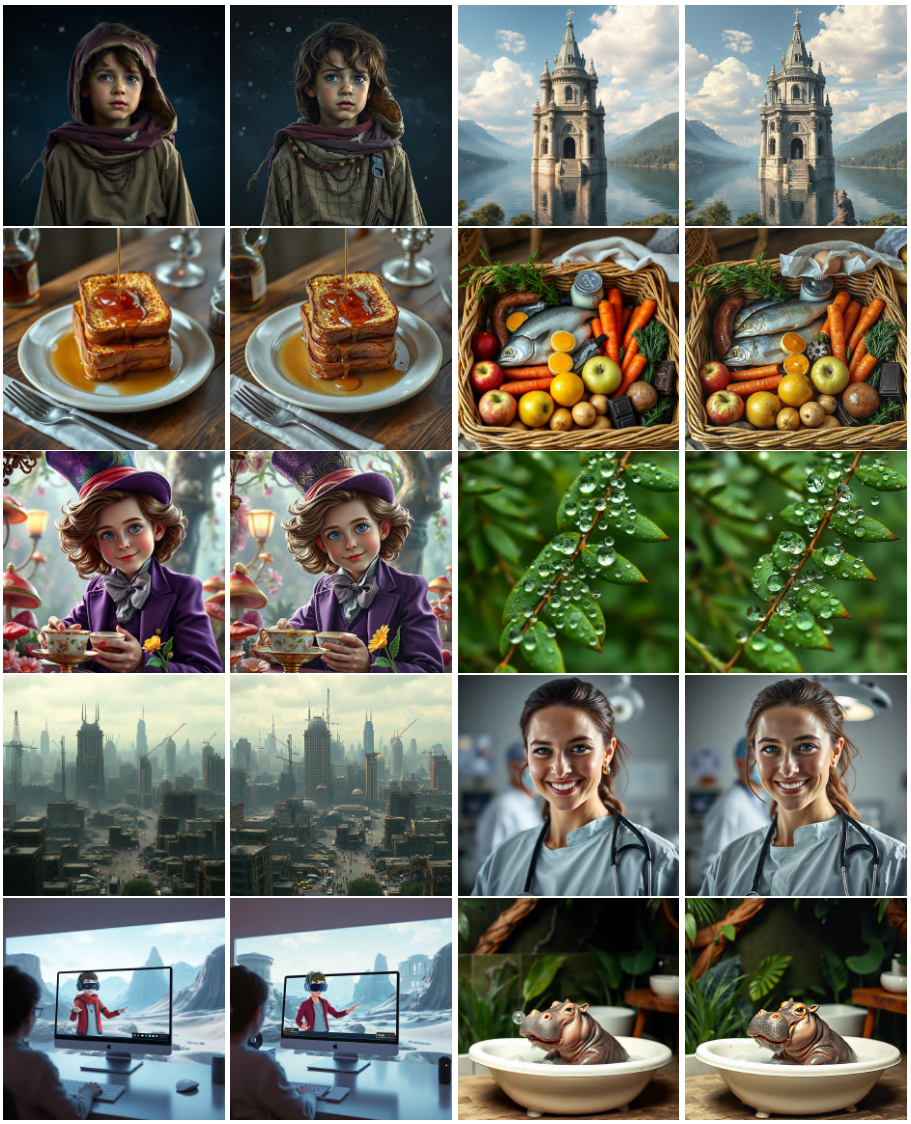


Figure A1: Side by side visual comparison of SVDQuant results and Q Sched corrected outputs for 10 MJHQ prompts. Each pair shows the SVDQuant output on the left and the corresponding Q Sched output on the right.

C Additional Experiments on SDXL-Turbo

Table A2 reports FID on MJHQ for SDXL-Turbo under different quantization precisions and calibration strategies. Under W4A8 quantization, naive and MixDQ baselines provide limited gains, while `Q-Sched` substantially reduces FID from 25.75 to 21.41, outperforming the FP16 reference. This indicates that scheduler-level calibration can steer the sampler toward a different and more favorable generation trajectory, rather than merely recovering the FP16 solution. For W8A8, all methods perform similarly, suggesting that the benefits of `Q-Sched` are most pronounced in the low-precision regime.

More broadly, we observe that FID trends do not always align with perceptual quality or user preference metrics for modern, highly optimized diffusion models such as SDXL-Turbo, where small trajectory changes can induce distinct image distributions. In contrast, for older architectures such as consistency models, FID remains a more informative signal and we therefore report it as a primary

metric in that setting. Accordingly, we present FID here as a complementary diagnostic alongside perceptual and preference-based evaluations, rather than as a sole indicator of image quality.

Table A2: FID (↓) on MJHQ for SDXL-Turbo under different quantization precisions. The baseline FID is 25.48 for the FP16 model.

Precision	Naive	MixDQ	Q-Sched
W4A8	25.75	25.36	**21.41**
W8A8	25.49	**25.16**	26.34

Table A3: Comparison of optimization strategies for Q-Sched on SDXL-Turbo. We use Nevergrad's OnePlusOne and NGOpt optimizers as representative of evolutionary search and meta-optimizers.

Method	CLIPScore ↑	CLIP-IQA ↑	HPV2 ↑	AQ-MAP ↑	Pick Score ↑	MANIQA ↑	JAQ (ours) ↑
Grid Search	**25.65**	0.709	**0.277**	**0.701**	18.37	0.434	**1.678**
OnePlusOne	25.17	**0.713**	0.274	0.700	18.43	**0.506**	1.673
NGOpt	25.27	0.706	0.272	0.700	**18.46**	0.495	1.674

D Details on User Preference Assessment

We design our evaluation setup following the user preference study methodology from SDXL-Turbo (Sauer et al., 2024), with several improvements. For each model pair in this study, we perform 1-vs-1 comparisons based on shared prompts. Human responses, collected via Rapidata (Rapidata, 2025), come from evaluators who are presented with two images, each generated by a different model for the same prompt, and are asked: "Which image is of higher quality and more aesthetically pleasing?"

Evaluators are globally sourced and must pass a set of validation questions designed to assess annotation quality. Only those who successfully complete this qualification step are allowed to rate the models.

ELO scores are computed using the same approach as SDXL-Turbo (Sauer et al., 2024), with K = 32 K=32. We find that this value of K enables more noticeable ranking adjustments, especially when models have similar performance levels.

All models in our study are evaluated using 1,000 prompts sampled from the MJHQ-30k dataset. We release this subset, which we call the Q-Sched split, to enable consistent benchmarking of future quantization methods. Each prompt is evaluated by four unique annotators. Therefore, each 1-vs-1 comparison results in 4,000 total human annotations.

E Compute Resources & Statistical Significance

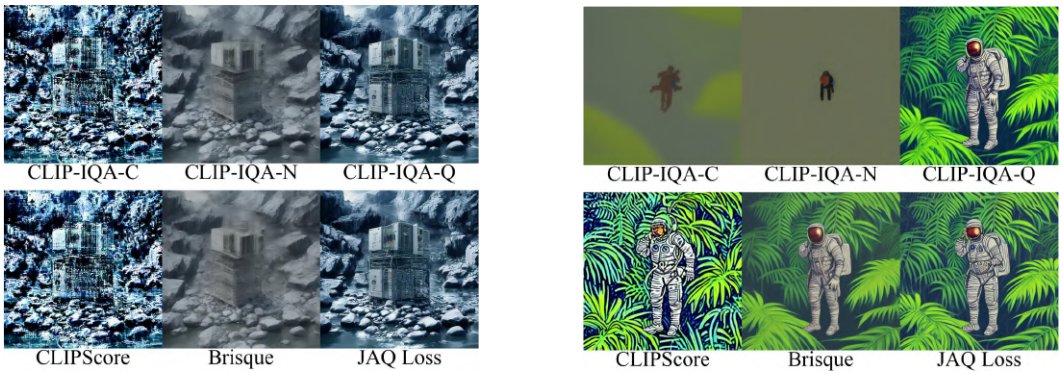
We conduct all our experiments on a high-end AI server with eight Nvidia A6000s. Each model can be run independently on one A6000 and Q-Sched takes approximately twenty minutes to run the full grid search.

Our main experiments are averaged over two-three runs but we do not report error bars at this time.

F Ablation Studies

F.1 Comparing Loss Functions for Q-Sched

To evaluate the overall image quality for text-image generative modeling, CLIPScore (Hessel et al., 2021) is specifically designed to capture text-image compatibility and does not consider overall image quality. In Figure A2, we illustrate that Q-Sched optimized with CLIPScore produces an updated noise schedule that is over saturated and lacks image depth. In contrast, Brisque (Mittal et al., 2012)



(a) Prompt: "An Eniac computer balanced on top of a stack of rocks over a river"

(b) Prompt: "Astronaut in a jungle, cold color palette, muted colors, detailed, 8k"

Figure A2: Optimizing `Q-Sched` with various reference-less image quality metrics. Our loss function, JAQ, is a linear combination of CLIPScore and CLIP-IQA-Q. We compare against three CLIP-IQA prompts: Complexity, Noisiness, and Quality denoted as -C, -N, -Q respectively.

Table A4: Adding stochasticity and its effect on W4A8 quantization for PCM using a Stable Diffusion v1-5 backbone. We report FID on COCO-30k. The stochasticity term, η, controls the amount of added Gaussian noise. $\eta = 0$ is deterministic sampling.

Method	$\eta =$					
	0	0.1	0.3	0.5	0.7	0.9
TCD	28.70	24.06	**23.44**	22.97	26.74	22.40
PTQD	23.33	25.59	24.95	25.69	24.53	26.71
`Q-Sched`	**22.24**	**19.29**	**23.44**	**19.67**	**19.46**	**17.87**

is often used as a standard reference-free image quality metric, but when used in `Q-Sched` it creates images with smoother and less detailed features. We consider three variants of CLIP-IQA (Wang et al., 2023b) and find that CLIP-IQA using the predefined quality prompt (we denote this version by CLIP-IQA-Q) achieves a noise schedule with high-fidelity images. However, CLIP-IQA-Q has a significant weakness: it cannot properly score images with hallucinations because it does not have an understanding of the underlying image prompt or concept. Therefore, we combine the benefits of CLIPScore and CLIP-IQA-Q into the JAQ loss and find that the resulting schedule fares extremely well with respect to raw image quality as well as to concept adherence.

F.2 Adding Stochasticity

Phased Consistency Model's implementation of the original sampler, TCD, is deterministic, meaning that there is no additive noise during sampling. The controllable noise parameter, η, allows a practitioner to adjust the additive noise during the sampling process and is defined in Equation (1). In order to compare PTQD's correction to our method, we ablate across different levels of stochasticity and report performance for six stochasticity levels in Table A4. $\eta = 0$ refers to deterministic sampling and PTQD's uncorrelated noise correction is not used since it adds stochastic noise by construction. Please see the appendix for more details on PTQD's implementation in both deterministic and stochastic sampling regimes.

We find that `Q-Sched` outperforms PTQD for all stochasticity regimes on the 2-step phased consistency model. With learnable coefficients and no Gaussian priors, `Q-Sched` can outperform PTQD and the original TCD scheduler in different sampling regimes.

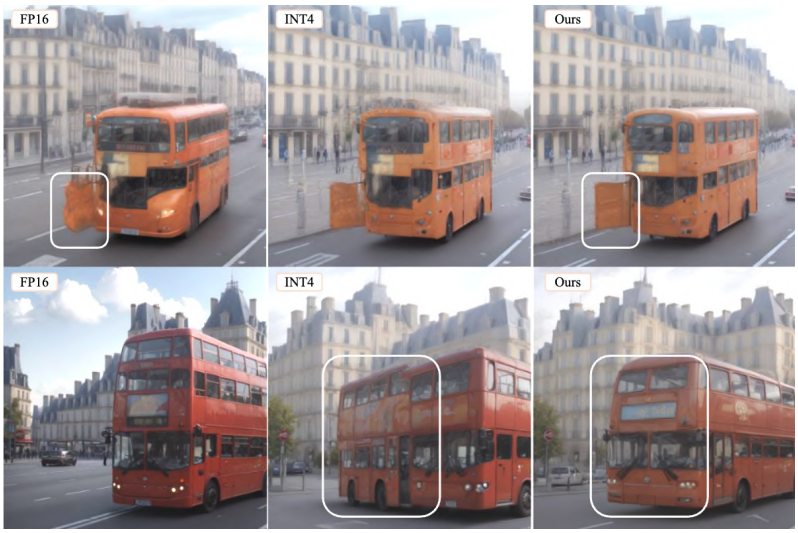


Figure A3: 4-Step (top row) and 8-Step (bottom row) LCMs. Prompt: "a car and a bus on a french highway". `Q-Sched` is capable of avoiding artifacts present in the FP16 or INT4 generative images. `Q-Sched` is close to the original schedule since it generates similar images yet our optimized schedule allows for `Q-Sched` to avoid some artifacts generated from the original schedule.

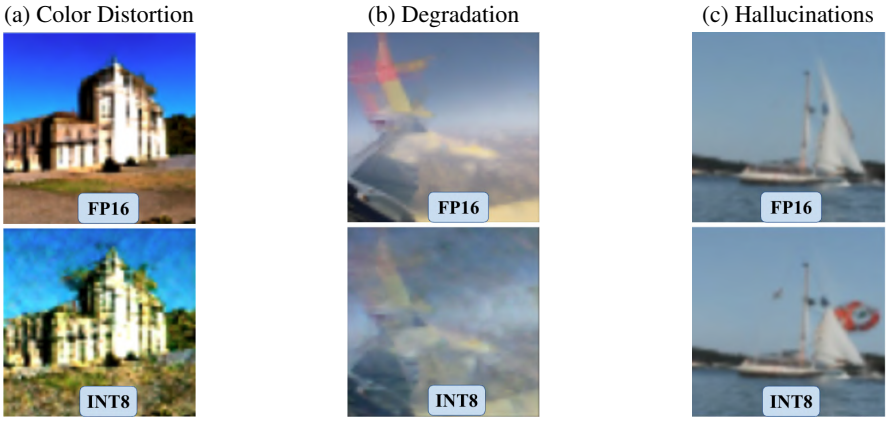


Figure A4: Three types of image artifacts that occur when quantizing image generation models. Images are unconditionally generated from a Two-Step Consistency Model (Song et al., 2023).

G Quantization-Induced Artifacts

As shown in Figure A3, `Q-Sched` is able to generate images that differ sufficiently from the full precision model. We ground our quantized diffusion model with image quality metrics, rather than the error with respect to full precision.

In our preliminary analysis using a two-step Consistency Model, we observed several characteristic ways in which quantization degrades image quality. As shown in Figure A4, quantized models tend to exhibit three prominent types of artifacts: color distortion, image degradation, and hallucinated structures. These issues are especially pronounced in low-bit settings and appear consistently across a variety of models and prompts.

H LATENCY AND MODEL SIZE ANALYSIS

In Table A5, we show the full model size breakdown for the diffusion model backbone, text encoders, and the VAE decoder. During inference, either one or both text encoders are used, and we do not need the VAE encoder, since this is for training exclusively.

Table A5: FP16 Diffusion Model Size Breakdown (in GB)

	LCM	PCM	SDXL-Turbo	FLUX.1[schnell]
UNet/DiT	1.72	4.84	1.03	4.76
Text Encoder(s)	0.25	0.29	0.33	1.95
VAE Decoder	0.07	0.13	0.02	0.02
Total	2.04 GB	5.26 GB	1.37 GB	6.73 GB

For our ELO *vs.* Model Size Pareto front in Figure 3, we consider the DiT memory and compute model size by taking the parameter count and multiplying it by the number of bytes required per parameter. For W4A4 + LoRA 64, the setup used for SVDQuant (Li et al., 2025), we compute the number of LoRA parameters using the back-of-the-envelope calculation provided in SVDQuant and add it to this calculation. We provide raw data for clarity in Table A6.

Table A6: DiT Memory (in GB) for various bitwidths.

Precision	SDXL-Turbo	FLUX.1[schnell]
FP16	1.03	4.76
W8A8	0.51	2.38
W4A4 + LoRA 64	0.28	1.24
W4A8	0.26	1.19

In Table A7, we provide latency on an RTX A6000 on all models following the same benchmarking procedure as in Table 1.

Table A7: Latency on Nvidia RTX A6000 in milliseconds. All models are evaluated in their 4-step setting.

	FLUX.1	SDXL	SDv1.5	SDXL-Turbo
bfloat16	3.881	0.640	0.193	0.252
w4a8	3.372	0.625	0.172	0.190
w4a8 `Q-Sched`	3.256	0.621	0.172	0.191

Prior work has already benchmarked latency for SVDQuant on FLUX.1[schnell] and MixDQ on SDXL-Turbo. Here, we directly summarize those experiments. In Tables A8 and A9, INT4 and INT8 are the models that we apply `Q-Sched` with no additional overhead.

Table A8: Reported in SVDQuant (Li et al., 2025) and summarized here for easy reference. All measurements are for FLUX.1[schnell] on an RTX 4090.

Method	**Latency (ms)**
BF16	657
INT8	282
INT4	212
SVDQuant	250
SVDQuant + Nunchaku	218

Table A9: Latency for SDXL-Turbo on RTX 4080. These results are reported in MixDQ (Zhao et al., 2024a) and summarized here for easy reference.

Method	**Latency (ms)**
BF16	24
INT8	16

I ADDITIONAL ANALYSIS ON COCO-30K

This result reinforces the core finding of our paper: quantization, when paired with a scheduler designed to account for noise sensitivity (as in `Q-Sched`), can be synergistic with few-step diffusion rather than detrimental. Notably, our quantized model achieves a lower FID than the original full-precision model, suggesting that `Q-Sched` helps overcome limitations introduced by both step reduction and bit-level compression.

These findings complement the results on SDXL-Turbo and FLUX.1[schnell] discussed in the main paper, and further establish `Q-Sched` as a general-purpose solution for high-fidelity, compressed diffusion generation.

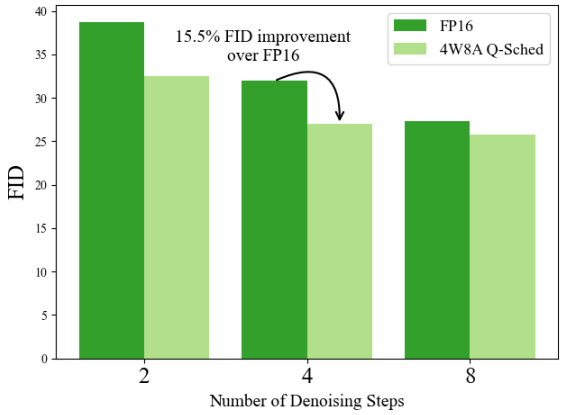


Figure A5: FID on COCO-30k. A W4A8 compressed model with our `Q-Sched` scheduler outperforms its FP16 counterpart with a $4\times$ reduction in model size.

J APPLYING PTQD TO THE TCD SCHEDULER

Using PTQD's linear parameterization for the quantization error, we substitute $\mathcal{E}_\theta^Q(x_t, t) = (1+\gamma) \cdot \mathcal{E}_\theta + \delta$ into Equation (1):

$$\mathbf{x_s} = \frac{\alpha_s}{\alpha_{s'}}\Big(\alpha_{s'}\frac{\mathbf{x_t} - \sigma_t \mathcal{E}_\theta(x_t, t)}{\alpha_t} + \sigma_{s'}\mathcal{E}_\theta(x_t, t)\Big) + \frac{\alpha_s}{\alpha_{s'}(1+\gamma)}(\sigma_{s'} - \frac{\alpha_{s'}\sigma_t}{\alpha_t})\delta + \sqrt{1 - \frac{\alpha_s^2}{\alpha_{s'}^2}}\mathbf{z}. \quad (4)$$

PTQD assumes the uncorrelated noise is sampled from a normal distribution $\delta \sim N(\mu_\delta, \sigma_\delta)$. This method applies bias correction to handle the mean deviation, μ_δ, and analytically compute standard deviation, σ_δ. We adapt PTQD's approach to the TCD schedule and use the new standard deviation, σ_δ for sampling δ:

$$\sigma_\delta^2 = 1 - \frac{\alpha_s^2}{\alpha_{s'}^2}(1 - (\frac{\delta(\sigma_{s'} - \frac{\alpha_{s'}\sigma_t}{\alpha_t})}{(1+\gamma)})^2). \quad (5)$$

For the edge case, where $\sigma_\delta < 0$, the deviation is set to zero ($\sigma_\delta = 0$). The proof for extending PTQD to the TCD scheduler is in the appendix.

PTQD attempts to model the distribution shift from a full precision to quantized model using two assumptions:

1. The quantized model's distribution shift can be modeled through a linear correction term.
2. The uncorrelated quantization noise is normally distributed.

While these assumptions are similar to prior work on diffusion models, they are likely to break down on the few-step diffusions where the denoising process is distilled from many steps and is not expected to be linear nor follow a Gaussian distribution.

Quantization Noise Correction using PTQD

Based on the PTQD quantization noise assumption, the quantization error is linearly parametrized as $\Delta\mathcal{E}_\theta = \gamma \cdot \mathcal{E}_\theta + \delta$ where γ, δ are learnable parameters corresponding to the correlated noise w.r.t. full precision and the uncorrelated noise respectively. PTQD models the uncorrelated noise as Gaussian (*i.e.*, $\delta \sim \mathcal{N}(\mu_q, \sigma_q)$).

Variance Schedule Calibration for Trajectory Consistency Distillation (TCD)

TCD's Strategic Stochastic Sampling (SSS) using a quantized network $\mathcal{E}_\theta^Q(x_t, t)$ is given by:

$$\mathbf{x_s} = \frac{\alpha_s}{\alpha_{s'}}\Big(\alpha_{s'}\frac{\mathbf{x_t} - \sigma_t\mathcal{E}_\theta^Q(x_t,t)}{\alpha_t} + \sigma_{s'}\mathcal{E}_\theta^Q(x_t,t)\Big) + \sqrt{1 - \frac{\alpha_s^2}{\alpha_{s'}^2}}\mathbf{z} \tag{6}$$

Using PTQD's linear parametrization for the quantization error, we substitute $\mathcal{E}_\theta^Q(x_t, t) = (1+\gamma) \cdot \mathcal{E}_\theta + \delta$:

$$\mathbf{x_s} = \frac{\alpha_s}{\alpha_{s'}}\Big(\alpha_{s'}\frac{\mathbf{x_t} - \sigma_t((1+\gamma)\cdot\mathcal{E}_\theta(x_t,t) + \delta)}{\alpha_t} + \sigma_{s'}((1+\gamma)\cdot\mathcal{E}_\theta(x_t,t) + \delta)\Big) + \sqrt{1 - \frac{\alpha_s^2}{\alpha_{s'}^2}}\mathbf{z} \tag{7}$$

$$= \frac{\alpha_s}{\alpha_{s'}}\Big(\alpha_{s'}\frac{\mathbf{x_t} - \sigma_t(1+\gamma)\mathcal{E}_\theta(x_t,t)}{\alpha_t} + \sigma_{s'}(1+\gamma)\mathcal{E}_\theta(x_t,t) - \frac{\alpha_{s'}\sigma_t\delta}{\alpha_t} + \sigma_{s'}\delta\Big) + \sqrt{1 - \frac{\alpha_s^2}{\alpha_{s'}^2}}\mathbf{z} \tag{8}$$

$$= \frac{\alpha_s}{\alpha_{s'}}\Big(\alpha_{s'}\frac{\mathbf{x_t} - \sigma_t(1+\gamma)\mathcal{E}_\theta(x_t,t)}{\alpha_t} + \sigma_{s'}(1+\gamma)\mathcal{E}_\theta(x_t,t)\Big) + \frac{\alpha_s}{\alpha_{s'}}(\sigma_{s'} - \frac{\alpha_{s'}\sigma_t}{\alpha_t})\delta + \sqrt{1 - \frac{\alpha_s^2}{\alpha_{s'}^2}}\mathbf{z} \tag{9}$$

The correlated noise can be corrected by applying:

$$\frac{\mathcal{E}_\theta^Q(x_t,t)}{1+\gamma} = \frac{(1+\gamma)\mathcal{E}_\theta(x_t,t) + \delta}{1+\gamma} \tag{10}$$

$$= \mathcal{E}_\theta(x_t,t) + \frac{\delta}{1+\gamma} \tag{11}$$

The resultant SSS sampling step becomes:

$$\mathbf{x_s} = \frac{\alpha_s}{\alpha_{s'}}\Big(\alpha_{s'}\frac{\mathbf{x_t} - \sigma_t\mathcal{E}_\theta(x_t,t)}{\alpha_t} + \sigma_{s'}\mathcal{E}_\theta(x_t,t)\Big) + \frac{\alpha_s}{\alpha_{s'}(1+\gamma)}(\sigma_{s'} - \frac{\alpha_{s'}\sigma_t}{\alpha_t})\delta + \sqrt{1 - \frac{\alpha_s^2}{\alpha_{s'}^2}}\mathbf{z} \tag{12}$$

The variance schedule becomes:

$$\sigma_\delta^2 = 1 - \frac{\alpha_s^2}{\alpha_{s'}^2} - (\frac{\alpha_s}{\alpha_{s'}(1+\gamma)}(\sigma_{s'} - \frac{\alpha_{s'}\sigma_t}{\alpha_t}))^2\delta^2 \tag{13}$$

$$= 1 - \frac{\alpha_s^2}{\alpha_{s'}^2}(1 - \frac{(\sigma_{s'} - \frac{\alpha_{s'}\sigma_t}{\alpha_t})^2}{(1+\gamma)^2}\delta^2) \tag{14}$$

$$= 1 - \frac{\alpha_s^2}{\alpha_{s'}^2}(1 - (\frac{\delta(\sigma_{s'} - \frac{\alpha_{s'}\sigma_t}{\alpha_t})}{(1+\gamma)})^2) \tag{15}$$

Since $\mathbf{z} \sim N(\mu_\delta, \sigma_\delta)$, we must handle the edge case when $\sigma_\delta < 0$. If the variance is negative, we simply set $\sigma_\delta = 0$.

Upon comparing `Q-Sched` to PTQD you may ask "*Why is `Q-Sched` able to learn a better noise schedule when it is also a linear correction?*" `Q-Sched` learns scalar coefficients on x_t and $\mathcal{E}_\theta$ that are optimized with respect to the reference-free JAQ loss. This allows us to learn a new schedule with linear corrections to improve our overall noise schedule, rather than matching the existing full precision schedule. This is an important distinction from PTQD, which tries to learn a linear correction with respect to full precision, which may not be possible since quantization produces a nonlinear distortion on the diffusion model. In short, PTQD attempts to match the full precision sampling trajectory, whereas `Q-Sched` aims to learn a new sampling trajectory given a compressed $\mathcal{E}_\theta$.

K Proof of Theorem 1: Strict Existence Guarantees for Quantization-Aware Scheduling

Theorem 1 (Strict Existence Guarantees)**.** *There exist `Q-Sched` coefficients* $(\mathbf{c}^\epsilon, \mathbf{c}^\mathbf{x}) \neq 0$ *such that* $E[||\Delta\tilde{x_0}||] < E[||\Delta x_0||]$.

Proof. Let us consider the few-step sampling trajectories for the pre-trained and quantized models, parametrized by $\mathcal{E}_\theta(t)$ and $\mathcal{E}_\theta^Q(t)$ respectively. These two few-step diffusion models sample at the same time-steps, $0 = t_0 < t_1, t_2 \cdots t_N = T$, where N represents the number of steps in the few-step model. For ease of notation, we will use the time-step 0 to refer to t_0 and 1 to refer to t_1, *etc.* A denoising step going from time $t+1 \to t$, produces a partially denoised image, x_t, and its quantized counterpart, x_t^Q. Following directly from Equation 9, the denoising error, $\Delta x_t = x_t - x_t^Q$, can be explicitly computed as:

$$\Delta x_t = \frac{\alpha_t}{\alpha_{t'}}\Big(\alpha_{t'}\frac{\Delta x_{t+1} - \sigma_{t+1}(\mathcal{E}_\theta(t+1) - \mathcal{E}_\theta^Q(t))}{\alpha_{t+1}} + \sigma_{t'}(\mathcal{E}_\theta(t+1) - \mathcal{E}_\theta^Q(t+1))\Big) \tag{16}$$

$$= \frac{\alpha_t}{\alpha_{t+1}}\Delta_{t+1} + \frac{\alpha_t}{\alpha_{t'}}(\sigma_{t'} - \frac{\sigma_{t+1}}{\alpha_{t+1}})(\mathcal{E}_\theta(t+1) - \mathcal{E}_\theta^Q(t+1)) \tag{17}$$

$$= k_t\Delta x_{t+1} + m_t\Delta\mathcal{E}_\theta(t+1)) \tag{18}$$

where we define the sampler coefficients as $k_t = \frac{\alpha_t}{\alpha_{t+1}}$, $m_t = \frac{\alpha_t}{\alpha_{t'}}(\sigma_{t'} - \frac{\sigma_{t+1}}{\alpha_{t+1}})$ and denote the change in the network as $\Delta\mathcal{E}_\theta(t) = \mathcal{E}_\theta(t) - \mathcal{E}_\theta^Q(t)$. Assuming the initial denoised image is the same ($x_N = x_N^Q$), the error in the final denoised image, Δx_0, is given by:

$$\Delta x_0 = k_0\Delta x_1 + m_0\Delta\mathcal{E}_\theta(1) \tag{19}$$

$$= k_0k_1k_2...(k_N\Delta x_N + m_{N-1}\Delta\mathcal{E}_\theta(N)) + \cdots + k_0k_1m_2\Delta\mathcal{E}_\theta(3) + k_0m_1\Delta\mathcal{E}_\theta(2) + m_0\Delta\mathcal{E}_\theta(1) \tag{20}$$

$$= \sum_{s=1}^{S}\Big(\Pi_{v=0}^{s-2}k_v\Big)m_{s-1}\Delta\mathcal{E}_\theta(s) \tag{21}$$

The average expected error over all images in a given dataset, $x_0 \in \mathcal{D}$, is given by:

$$E[||\Delta x_0||] = \sum_{s=1}^{S} \left(\Pi_{v=0}^{S-2} k_v\right) m_{s-1} E[||\Delta \mathcal{E}_\theta(s)||] \tag{22}$$

since $E[||\Delta x_0||]$ is a homogeneous function.

In `Q-Sched`, we apply our quantization-aware pre-conditioning on every noise coefficient: $\tilde{m}_t = c_t^\epsilon \cdot m_t$ and $\tilde{k}_t = c_t^x \cdot k_t$. Let us denote the expected error induced by `Q-Sched` with respect to the pre-trained model's x_0 as $E[||\Delta \tilde{x_0}||]$.

We empirically show in Tables 1 and 2 that $E[||\Delta x_0||] \neq 0$ since the images produced by the naive quantization method produce a different FID from the original pre-trained model's image distribution. Since Equation 22 is a linear function of $k_t, m_t, \forall t \in 1 \ldots N$, and there is a global minimum at $E[||x_0 - x_0||] = 0$, it must be that $\exists \tilde{m}_t^*, \tilde{k}_t^* \forall t$ such that $E[||\Delta \tilde{x_0}||] < E[||\Delta x_0||]$. In short, we guarantee that there exists quantization-aware coefficients that strictly improve our expected quantization error over naive quantization. □

K.1 Aside: Positive Sampler Coefficients

The TCD Scheduler has $\beta_0 = 0.0085, \beta_N = 0.012$, $\alpha_t = 1 - \beta_t, \sigma_t = \Pi_{i=0}^{t} \alpha_i$ with a scaled linear schedule:

$$\beta_t = \left(\sqrt{\beta_0} + t \cdot (\sqrt{\beta_N} - \sqrt{\beta_0})\right)^2 \tag{23}$$

Therefore: $1 > \alpha_0 > \alpha_1 > \cdots > \alpha_N > 0$ and $1 > \sigma_0 > \sigma_1 > \ldots \sigma_N > 0$. We note the $t' = (1-\gamma)t$ where $\gamma \in [0, 1]$, so $t' \leq t$. This implies that $\sigma_{t'} > \sigma_{t+1}$ so:

$$k_t > 0 \qquad , \qquad m_t = \frac{\alpha_t}{\alpha_{t'}} (\sigma_{t'} - \frac{\sigma_{t+1}}{\alpha_{t+1}}) > 0 \tag{24}$$

This illustrates that $k_t, m_t \in \mathbb{R}^+$.

L Scheduler coefficients affect final image quality

Two coefficients are expressive enough for modest compression scenarios where quantization causes a mild drift in the sampling trajectory. Adjusting the relative contribution of the predicted noise versus the intermediate state is typically enough to realign this trajectory. We observe catastrophic error when quantizing further to 3-bit integer weights and in contrast, observe only modest gains in the 8-bit scenario across both SDXL-Turbo and FLUX.1 [schnell]. We observe that `Q-Sched` works best in the 4-bit setting, where compression yields noticeable degradation but is not as effective in the 3-bit setting, where the generated images are very poor quality.

In Figure A6, we illustrate how three different coefficient combinations can yield images with unique artifacts and details. We take five images generated from FLUX.1[schnell] with different sampler coefficients parameterized by (c_{min}, c_{max}).

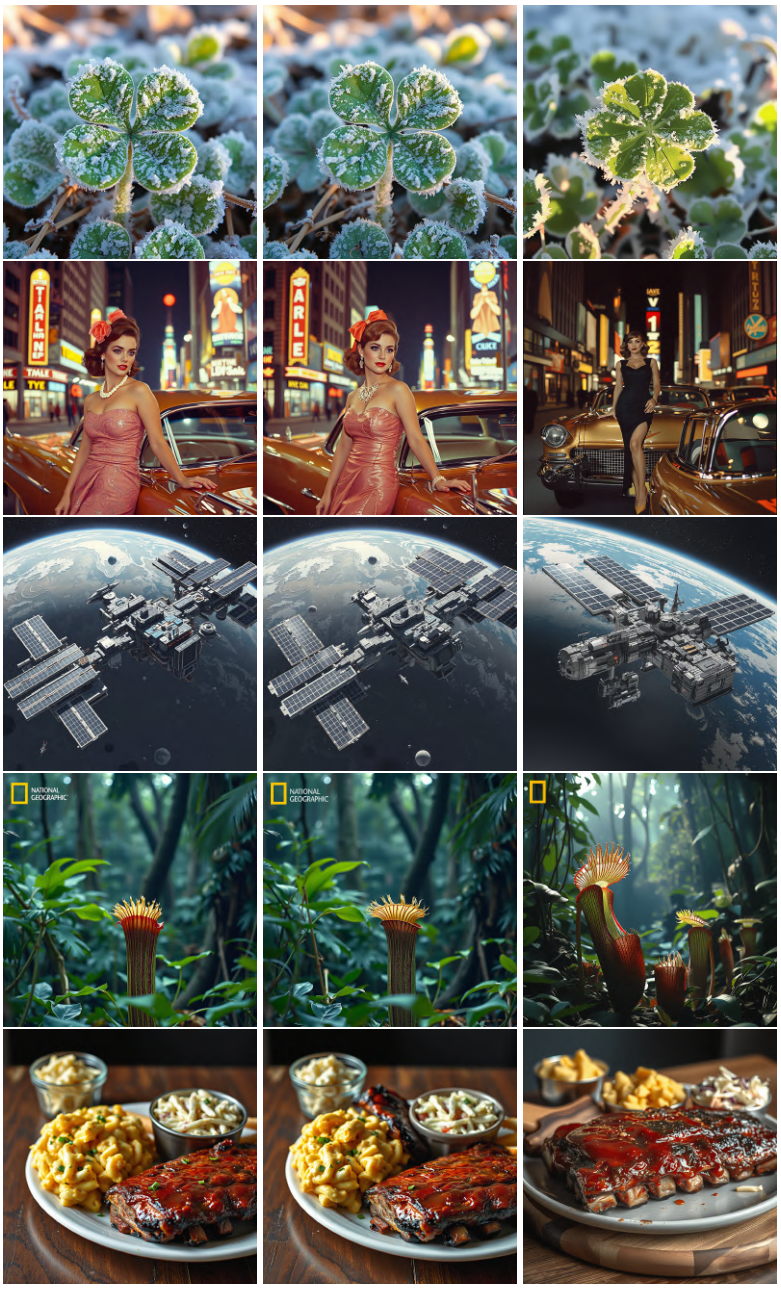


Figure A6: Qualitative comparison of generated samples across three coefficient settings. Each row corresponds to a consistent prompt, random seed and model. From left to right, coefficients (c_{min}, c_{max}) are $(0.8, 0.933)$, $(0.843, 0.909)$, and $(1.10, 0.82)$.

M Ablation on `TC()` *vs.* `IQ()`

In Table A10, we show that optimizing `Q-Sched` with respect to `TC()` or `IQ()` alone improves some metrics but not others. For example, optimizing purely for text compatibility with AQ-MAP yields strong CLIP Score and Pick Score, but noticeably lower CLIP-IQA and HPV2. In contrast, optimizing with JAQ produces consistent gains across both image quality and text compatibility. We see improvements in CLIP Score and CLIP-IQA while still maintaining strong performance on the text compatibility metrics.

Table A10: Comparison of scheduler objectives across metrics.

Metric	AQ-MAP	HPV2	JAQ
CLIP Score ↑	25.266	25.043	**25.269**
CLIP-IQA↑	0.698	0.705	**0.707**
HPV2 ↑	0.271	**0.278**	0.272
AQ-MAP ↑	**0.701**	0.696	0.700
JAQ (ours) ↑	1.673	1.669	**1.673**
Pick Score ↑	**18.50**	18.38	18.46

N Scalability to Other Types of Diffusions

In Table 1, we display scalability across 2,4, and 8-step consistency models and have every indication that our method will scale to longer trajectories. Yet, we emphasize that the scope of our work is to consider few-step diffusion models, which are typically less than 8 steps.

Following the same setting as in Table 1, we also provide a single run on PCM with 16 steps, illustrating exciting performance and the potential for Q-Sched to improve on longer step regimes:

Table A11: `Q-Sched` on 16-Step Phased Consistency Model, illustrating scalability to a larger number of steps.

16-step	**FID**
FP16	19.829
INT4	19.567
INT4 `Q-Sched`	15.969

Why not other types of few-step models? One popular few-step diffusion model, SANA-sprint (Chen et al., 2025), is a much smaller model (0.6B parameters) than SDXL-Turbo or FLUX.1, so quantization offers limited benefit and tends to cause more degradation at this scale. `Q-Sched` is most impactful for distilled, few-step models where size meaningfully affects inference speed, making SANA-sprint a less natural target. While `Q-Sched` could be applied, we leave this direction for future work.

O Optimization Details

Optimizing scheduler coefficients independently at each timestep is a natural baseline, but it induces a prohibitively large search space that grows exponentially with the number of denoising steps. Instead, `Q-Sched` parameterizes the scheduler using only two coefficients per variable, (c_{start}, c_{end}) for both $\mathbf{c}^{\mathbf{x}}$ and $\mathbf{c}^{\epsilon}$, and applies linear spacing across timesteps. This parameterization avoids an exponentially large search space while yielding strong empirical performance.

Under a naive formulation, the search complexity scales as $O(n^{\omega} \cdot |\mathcal{C}|)$, where:

- n is the number of candidate values per coefficient,
- ω is the number of diffusion steps, and
- $|\mathcal{C}|$ is the calibration dataset size.

As the number of steps increases, grid search over per-timestep coefficients quickly becomes intractable. As shown in Algorithm 1, `Q-Sched` removes the exponential dependence on ω by learning only the start and end coefficients and deterministically interpolating intermediate values. This reduces the effective search space to a constant dimensionality, benefiting both grid search and evolutionary optimization methods. As a result, `Q-Sched` scales linearly with the calibration set size and remains independent of the number of denoising steps.