LIGHTWEIGHT SPECTRAL STATISTICAL CORRECTION FOR FAST SAMPLING IN FLOW MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion and flow matching models achieve remarkable quality in various generation tasks. However, solving the flow ODE to a good precision typically requires many network evaluations, leading to high latencies. Multiple lightweight methods have been proposed for fast sampling with pretrained models. However, the vast majority of them focus on minimizing the truncation error of the underlying ODE solver, and do not attempt to directly control the distributional shift caused by taking large discretization steps. In this paper, we demonstrate that adding a simple fast analytical statistical correction after each solver step can significantly improve the quality of the generated images. Specifically, we present Spectral Statistical Correction (SSC) – a linear transformation applied independently to each DCT coefficient of the latent state so as to optimally correct its distribution. SSC can be learned from a small number of images without requiring backpropagation through the model. Furthermore, it can be applied on top of other fast sampling methods. As we show, SSC consistently leads to substantial improvements even over strong samplers, allowing to obtain high quality results with as little as 5 NFEs with modern flow models.

1 INTRODUCTION

Diffusion (Sohl-Dickstein et al., 2015; Ho et al., 2020) and flow (Lipman et al., 2023; Liu et al., 2022b) based generative models have achieved remarkable quality in image generation (Labs, 2024; Esser et al., 2024), image editing (Labs et al., 2025; Wu et al., 2025), audio generation (Liu et al., 2023), video generation (Wan et al., 2025) and various other generative tasks (Hoogeboom et al., 2022). However, these models often require many network evaluations for reaching high quality generation. Distillation based approaches (Salimans & Ho, 2022; Song et al., 2023; Yin et al., 2024) can produce models that retain high sampling quality with fewer network evaluations. However, these techniques require substantial resources, especially when considering the size of modern models. As a result, much attention has been given to lightweight approaches, which do not require model training or finetuning (Zhou et al., 2023; Tong et al., 2025; Chen et al., 2024). These methods may still use some type of optimization, but which does not involve updating model weights and thus are computationally negligible compared to model distillation.

Most existing approaches for lightweight fast sampling focus on reducing the truncation error induced by solving the diffusion or flow ODE with a coarse temporal discretization. This is commonly achieved by improving the timestep grid (Chen et al., 2024; Tong et al., 2025), the solver (Zhou et al., 2023), both (Frankel et al., 2025; Shaul et al., 2024b), or the noise scheduler (Shaul et al., 2024a). Superficially, matching the trajectory of a coarsely sampled flow process with that of a finely sampled one may seem similar to the objective targeted by distillation approaches. For example, Consistency Distillation (Song et al., 2023) focuses on endpoint matching. However, many distillation approaches combine such teacher regression losses with distributional losses. For example, ADD (Sauer et al., 2024b) and LADD (Sauer et al., 2024a) use GANs, while DMD (Yin et al., 2024) uses a surrogate score model. In contrast, lightweight fast sampling approaches ignore the distributional shift caused by the trajectory discretization, despite its impact on perceptual quality.

In this paper, we adopt the distributional motivation behind many distillation methods, while avoiding heavy optimization. This is achieved by adding a simple and fast statistical correction after each solver step. The key ingredient is a spectral statistical correction (SSC) – a univariate affine

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.



Figure 1: Comparison between images sampled from FLUX.1 [dev] with and without SSC. Our correction consistently improves the quality of the base solver (demonstrated here for Euler, DPM++ (2M) and iPNDM (2M), TORS, BézierFlow and DyWeight), leading to clearer fine details and better contrast at both 5 and 7 NFEs. Prompt is from MS-COCO (Lin et al., 2014).

transform corresponding to the Gaussian optimal transport (OT), applied to each coefficient of the discrete cosine transform (DCT) independently. This transform is supplemented by a correction in a learned basis, which further improves the results. SSC is a general framework that can be used on top of a wide variety of solvers and sampling methods. As we show, it achieves strong results by itself (*i.e.*, by correcting the Euler solver) but also consistently improves other solvers, including handcrafted (DPM++ (Lu et al., 2025), iPNDM (Zhou et al., 2026; Zhang & Chen, 2022)) and learned ones (DyWeight (Zhao et al., 2026)), as well as other fast sampling methods that modify the noise schedule or the timestep discretization (Fig. 1, for additional examples see Fig. 4 and Fig. 5).

2 Background

Flow Models. Flow models directly predict a time dependent velocity field $v_\theta(x, t)$ such that solving the ODE

$$\frac{dx}{dt} = v_\theta(x, t), \quad x(1) \sim p_1, \tag{1}$$

from $t = 1$ to $t = 0$ results in samples from the target distribution p_{data} (Lipman et al., 2023; Liu et al., 2022b). The distribution p_1 is commonly chosen to be the standard Gaussian distribution, $\mathcal{N}(0, I)$. A common schedule choice is the rectified flow schedule, $X(t) = tX(1)+(1-t)X(0)$ with $X(0) \sim p_{\text{data}}, X(1) \sim p_1$. Under the rectified flow schedule $v_\theta(x, t)$ approximates $\mathbb{E}[X(1)-X(0) \mid X(t) = x]$. At each time t along the trajectory, the clean image $X(0)$ can be estimated from the

current state $X(t)$ as

$$\hat{x}(0 \mid t) := \mathbb{E}[X(0) \mid X(t)] = X(t) - tv_\theta(X(t), t). \tag{2}$$

Optimal Transport. Given two distributions, p and q on $\mathcal{X}$, the Monge problem concerns with finding a mapping T that transforms the distribution p into q while attaining a minimal cost,

$$\inf_T \mathbb{E}\left[c(x, T(x))\right] \quad \text{such that } T_{\#}p = q, \tag{3}$$

for some cost function c (Villani et al., 2009). Under mild regularity assumptions such T exists. For $\mathcal{X} = \mathbb{R}^d$ and $c(x, x') = \|x - x'\|^2$ the minimal cost is the square of the Wasserstein-2 distance between p and q. The solution for two Gaussian distributions $p(x) = \mathcal{N}(x; \mu_1, \Sigma_1)$, $q(x) = \mathcal{N}(x; \mu_2, \Sigma_2)$ is given by (Villani et al., 2009)

$$T(x) = \mu_2 + \Sigma_1^{-1/2}\left(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2}\right)^{1/2}\Sigma_1^{-1/2}(x - \mu_1). \tag{4}$$

3 Related Work

Lightweight Sampling Acceleration. Most lightweight sampling acceleration methods optimize the solution process of the flow ODE by modifying either the time discretization grid, the ODE solver, or both. Early works such as Karras et al. (2022), use handcrafted timestep discretization methods and classical ODE solvers. In the context of diffusion models, DPM-Solver (Lu et al., 2022), DEIS (Zhang & Chen, 2022) and DPM-Solver++ (Lu et al., 2025) design analytic solvers that utilize the structure of the probability flow ODE (PF-ODE) (Song et al., 2020). In contrast, iPNDM (Liu et al., 2022a; Zhang & Chen, 2022) heuristically uses the AB-2 coefficients (Butcher, 2016), even when the time discretization is nonuniform. A different line of work adds model specific optimization. AMED-Solver and AMED-Plugin (Zhou et al., 2023) dynamically select midpoint evaluation points using a tiny neural network that is trained to match the full student and teacher trajectories. DMN (Xue et al., 2024) optimizes the discretization grid using a theoretical bound on the global truncation error, while GITS (Chen et al., 2024) uses dynamic programming and estimated local truncation errors. TORS (Zhou et al., 2026) analyzes the curvature of the sampling trajectory in order to select the timesteps. LD3 (Tong et al., 2025) optimizes the timestep grid using a modified version of the global truncation error. Bespoke Solvers (Shaul et al., 2024a) and BézierFlow (Min et al., 2026) optimize both the discrete timetsteps and the noise scheduler using local and global truncation error, respectively. BSN (Shaul et al., 2024b), S4S-Alt (Frankel et al., 2025) and DyWeight (Zhao et al., 2026) optimize both the timestep grid and a multistep solver using the global truncation error.

Corrections During Sampling. Exposure Bias (Ning et al., 2023) refers to the difference between training and inference distributions of diffusion models. It has been suggested (Ning et al., 2024) to add corrections during the sampling process to correct the denoiser's input or output. Related to our work, Yu & Zhan (2025) add an extremely simple discrete wavelet transform (DWT) correction, but do not provide an algorithm for its estimation. Kobayashi et al. (2026) perform a discrete Fourier transform (DFT) based correction, but it is different from ours as it is non linear, and is not designed for low NFEs.

Concurrent Work. Our algorithm is conceptually similar to the algorithm of Liang et al. (2026), with both algorithms using sequential estimation. However, their approach is applied specifically on feature caching, and their calibration method is vastly different from ours, as their correction is a high dimensional rigid transformation applied on cached tokens. SpanLift (Kim et al., 2026), similar to our method, operates on top of an existing solver. However, similar to previous works, it targets the truncation error rather than the distributional shift.

4 Method

4.1 Problem Definition

Let $D = CHW$ denote the dimension of a $C \times H \times W$ flattened image. We denote an ODE solver by $(\Phi_i)_{i=1}^N$, which is a collection of N functions $\Phi_i : \mathbb{R}^D \to \mathbb{R}^D$, each performing a single solver

Algorithm 1 Statistical Correction Estimation

Require: Student solver $(\Phi_i)_{i=1}^{N}$ that uses timetseps $1 = t_0, t_1, \ldots, t_N = 0$.
Require: Teacher solver $\left(\Phi_i^T\right)_{i=1}^{N_T}$ that uses timetseps $1 = t_0^T, t_1^T, \ldots, t_{N_T}^T = 0$
Sample M samples $x_0^{(1)}, \ldots x_0^{(M)} \sim \mathcal{N}(0, I)$
Simulate trajectories of $x_0^{(1)}, \ldots x_0^{(M)}$ using the teacher solver
for $i = 1, \ldots, N$ **do**
    for $j = 1, \ldots,$ M **do**
        $\bar{x}_i^{(j)} = \Phi_i(x_{i-1}^{(j)})$
    end for
    Estimate correction C_i based on statistics of $\bar{x}_i^{(1)}, \ldots \bar{x}_i^{(M)}$ and teacher statistics at t_i.
    for $j = 1, \ldots,$ M **do**
        $x_i^{(j)} = C_i(\bar{x}_i^{(j)})$
    end for
end for

step. In our notation Φ encapsulates the timesteps and the velocity. For example the Euler solver is given by $\Phi_i(x) = x + \Delta t_i v(x, t_i)$. Let $\Psi(x, t_i) = \Phi_i \circ \cdots \circ \Phi_1(x)$ denote the state at timestep t_i obtained by initializing with x and taking i solver steps. Let $(\Phi_i^T)_{i=1}^{N_T}$ be a teacher solver with a large number of NFEs, and let $(\Phi_i^S)_{i=1}^{N_S}$ be a student solver with a small number of NFEs ($N_T > N_S$). As discussed in Sec. 3, most lightweight fast sampling methods aim at making the student solver mimic the teacher solver at the final time $t = 0$, so that the student and teacher trajectories satisfy

$$\Psi_S(x, 0) \approx \Psi_T(x, 0), \tag{5}$$

on average for $x \sim p_1$.

We claim that this approach is too strict, as for most practical use cases it is enough for the student solver to merely generate samples from the same distribution as the teacher solver. Formally, it is enough to match the push forward distributions induced by the solvers at $t = 0$,

$$\left(\Psi_S(\cdot, 0)\right)_{\#} p_1 \approx \left(\Psi_T(\cdot, 0)\right)_{\#} p_1. \tag{6}$$

To address the distributional mismatch, we propose to add a correction $C_i : \mathbb{R}^D \to \mathbb{R}^D$ after each student solver step, resulting in a new student solver $\tilde{\Phi}_i^S$ defined by

$$x_i = \tilde{\Phi}_i^S(x_{i-1}) = C_i\left(\Phi_i^S(x_{i-1})\right), \tag{7}$$

and new trajectory $\tilde{\Psi}_S$ such that the distribution of the states induced by the corrected solver $(\tilde{\Phi}_i^S)_{i=1}^{N_S}$ approximately matches that of the teacher,

$$\left(\tilde{\Psi}_S(\cdot, t_i)\right)_{\#} p_1 \approx \left(\Psi_T(\cdot, t_i)\right)_{\#} p_1, \tag{8}$$

at each student timestep t_i.

In order to avoid overhead during inference and heavy computations for finding the corrections, we limit ourselves to simple corrections. Indeed, complex corrections require substantial compute and complicated estimation, as *e.g.,* in Transition Matching (Shaul et al., 2026). As each of our corrections is weak, and since the distributional drift increases with time (Chen et al., 2022), it is natural to employ a correction after each timestep. Such corrections have the added benefit of bringing the model's input closer to the model's training distribution. For a discussion about conditional sampling, see App. A.2.

As earlier corrections affect later ones, we estimate them sequentially, as shown in Alg. 1. Specifically, we maintain a set of states, advance them together at each solver step, use them to estimate a correction, and then apply the correction to obtain the states at the next timestep. This ensures that each correction correctly depends on the previous ones, and ensures that Eq. (8) approximately holds after each solver step. Since we use a relatively small number of samples, they can be stored in memory. Sampling is shown in Alg. 2.

Algorithm 2 Statistical Correction Sampling

Require: Student solver $(\Phi_i)_{i=1}^N$, corrections $(C_i)_{i=1}^N$
Require: Initial sample $x_0 \sim \mathcal{N}(0, I)$
1: **for** $i = 1, \ldots, N$ **do**
2: $\quad x_i = C_i(\Phi_i(x_{i-1}))$
3: **end for**
4: **return** x_N



Figure 2: Comparison of FLUX.1 [dev] generated images with DPM++ (2M) (Lu et al., 2025), using FLUX.1's default timestep discretization. Top row: 35 NFEs, Bottom row: 5 NFEs. Low contrast and corrupted high frequencies are visible in the 5 NFE images. Prompts are from MS-COCO 2014 (Lin et al., 2014).

4.2 Analyzing the Statistical Difference

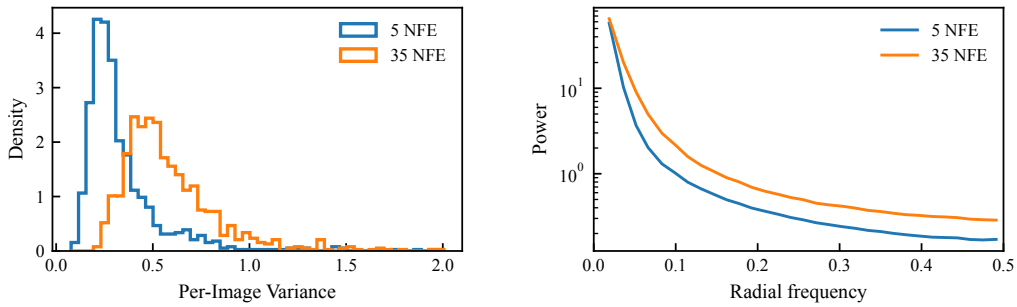
To construct effective corrections, we begin by analyzing the statistics of images sampled using FLUX.1 [dev] with a small number of NFEs (5), and a large number of NFEs (35). Figure 2 depicts several visual examples. As can be seen, images sampled with a small number of timesteps suffer from low contrast and corrupted high frequencies, especially in regions of constant texture (see the road in the bike image). Similar distributional shifts are easily identified in the latent space (prior to VAE decoding), as seen in Fig. 3. Specifically, the distribution of per-image contrast, measured as the variance of values within a single image, is distinctively different between the 5 and 35 NFE settings. Furthermore, differences in spatial spectral contents, which we quantify using radial mean power spectrum, are quite profound. Images sampled with 5 NFEs have lower per image variance and lower spectral power at all frequencies, consistent with the visual artifacts in pixel space. As we show in App. A.3, these statistical differences persist across the entire sampling trajectory.

4.3 Designing the Correction

To make the correction amenable to estimation from a small number of training samples and add negligible overhead during sampling, we limit ourselves to affine corrections of the form

$$C = C_{\text{learned}} \circ C_{\text{fixed}}. \tag{9}$$

Here, C_{fixed} transforms each channel of x into its representation in a fixed orthogonal basis for $\mathbb{R}^{HW}$ (not estimated from the training samples), applies a per-coefficient affine correction in that basis, and transforms x back, while C_{learned} works in a similar manner only in a subspace spanned by a small set of $k \ll HW$ orthogonal directions estimated from the training samples. This choice has a small number of learnable parameters (significantly smaller than the D^2 parameters required for a general linear transformation in latent space). As such, it can be estimated efficiently from a small set of samples. However, for it to be effective, the transformation C_{fixed} must capture significant statistical deviations along its basis vectors. In addition, both C_{fixed} and its inverse must be fast to apply. To satisfy those two requirements, we use our observations from Sec. 4.2.



(a) Distribution of per-image variance.

(b) Radial mean power spectrum.

Figure 3: Comparison of channel 0 (out of 16) statistics of 1000 FLUX.1 [dev] generated images in latent space (prior to VAE decoding). (a) Distribution of image variance (variance of latent channel values inside a single image). (b) Estimated radial mean power spectrum of generated images.

4.4 DCT-Based Correction

Based on the conclusions about spectral mismatch, as seen in Fig. 3, we focus on correcting spectral statistics. We choose to use DCT for C_{fixed}. Unlike the discrete Fourier transform (DFT), DCT yields real-valued coefficients and does not suffer from boundary issues associated with periodicity assumptions. After applying the DCT transform, we correct each frequency band and channel independently. Specifically, denoting $x = C_{\text{fixed}}(\bar{x})$, the correction is obtained as

$$\begin{aligned} \bar{z} &= \text{DCT}(\bar{x}), \\ z_{c,i,j} &= a_{c,i,j}\bar{z}_{c,i,j} + b_{c,i,j}, \\ x &= \text{IDCT}(z). \end{aligned} \tag{10}$$

The coefficients $a_{c,i,j}$ and $b_{c,i,j}$ are chosen to implement the Gaussian optimal transport between the distributions of the student and the teacher (the 1D case of Eq. (4)). This choice is supported by the fact that most coefficients are approximately normally distributed (see App. A.1). Specifically, denoting the mean and standard deviation of the (c, i, j) coefficient of the student by $\mu^S_{c,i,j}, \sigma^S_{c,i,j}$ and those of the teacher by $\mu^T_{c,i,j}, \sigma^T_{c,i,j}$, we set

$$a_{c,i,j} = \frac{\sigma^T_{c,i,j}}{\sigma^S_{c,i,j}}, \quad b_{c,i,j} = \mu^T_{c,i,j} - a_{c,i,j}\mu^S_{c,i,j}. \tag{11}$$

4.5 Learned Basis Correction

While the DCT correction has a strong effect by itself, some statistical deviations are not captured by its individual basis vectors. The transform C_{learned} is intended to capture such deviations. Specifically, C_{learned} uses a learned set of k orthogonal vectors $u_1, \ldots, u_k \in \mathbb{R}^{HW}$. Each of these vectors is used to transform each of the channels, $x \to (u_1^\top x[1], \ldots, u_k^\top x[1], \ldots, u_1^\top x[C], \ldots, u_k^\top x[C])^\top$.

We want the vectors $u_1, \ldots, u_k$ to capture the directions of maximal variance of the updates at each solver step. We identify that the quantity

$$\tilde{x} := \hat{x}(0 \,|\, t_i) - \hat{x}\,(0 \,|\, t_{i-1}) \tag{12}$$

contains the innovation added to the image at step i (we omit the index i for conciseness). Therefore, to find directions for timestep t_i, we aggregate all the channels of the innovations corresponding to the M training samples at that timestep into one large dataset, $\{\tilde{x}^{(m)}[1]\}_{m=1}^M \cup \cdots \cup \{\tilde{x}^{(m)}[C]\}_{m=1}^M$, and select $u_1, \ldots, u_k$ as the top k principal components of this set.

As our DCT correction does not compensate for differences in the cross-covariances between the channels, we compensate for such differences in C_{learned}. Specifically, we jointly correct each one of the k vectors $(u_i^\top x[1],\ u_i^\top x[2], \ldots, u_i^\top x[C])^\top \in \mathbb{R}^C$ using Eq. (4). The orthogonality of $u_1, \ldots, u_k$ allows to efficiently project the corrected coefficients back into the latent image space.

We call the combined DCT and learned correction the Spectral Statistical Correction (SSC).

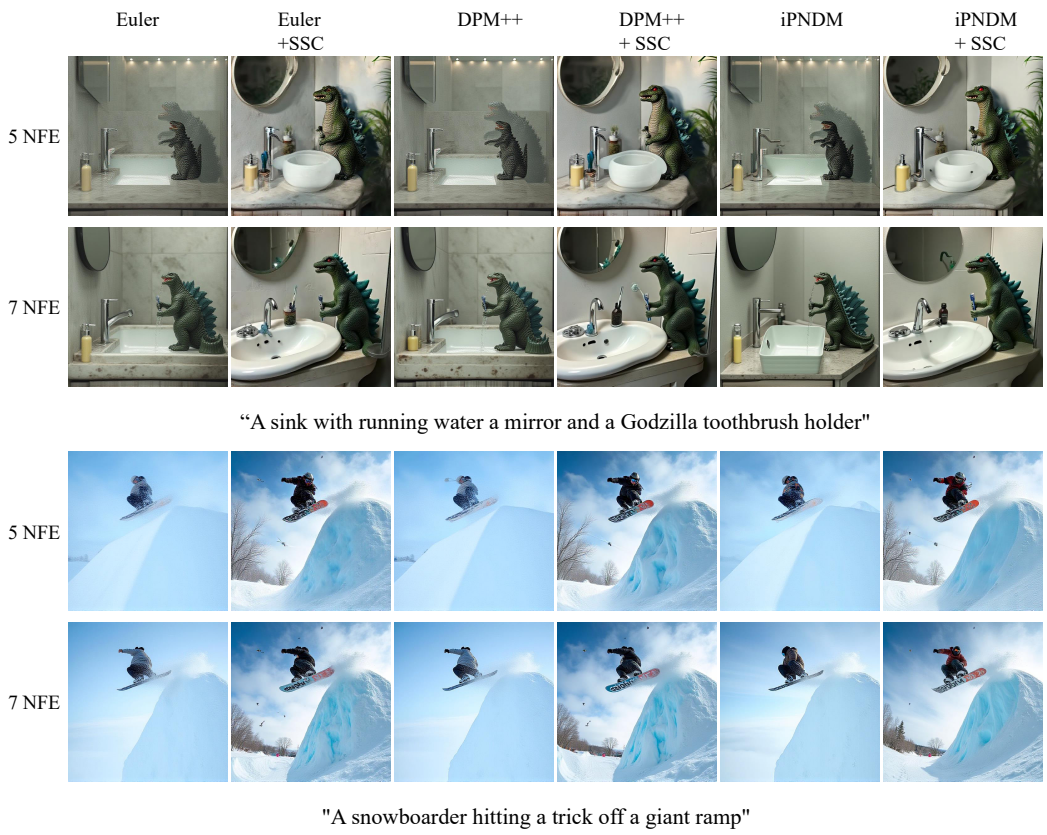


Figure 4: Comparison between images sampled from FLUX.1 [dev] with and without SSC, for handcrafted solvers (Euler, DPM++ (2M) and iPNDM (2M)). Clearer fine details and better contrast at both 5 and 7 NFEs are visible. Prompts are from MS-COCO (Lin et al., 2014).

5 Experiments

5.1 Experimental Setup

Our main evaluation is performed on FLUX.1 [dev]. We additionally provide Stable Diffusion 1.5 (Rombach et al., 2022) results, in order to show that SSC is not limited to flow based models. We perform the evaluation on a subset of 30k prompts from the evaluation set of MS-COCO 2014 (Lin et al., 2014), and fix a noise seed per image. All generated images have a resolution of 512×512. For all FLUX.1 [dev] experiments the guidance scale is set to 3.5, while for SD 1.5 the classifier free guidance (CFG) parameter is set to 7.5. FLUX.1 [dev] is evaluated for 5 and 7 NFEs, while SD 1.5 is evaluated for 4, 6 and 8 steps, corresponding to 8, 12 and 16 NFEs, due to the CFG.

For FLUX.1 we select $k = 40$ directions, following an ablation (see App. A.4). For SD 1.5, $k > 0$ leads to only a small improvement, so we select $k = 0$, resulting in a purely DCT-based correction.

For both FLUX.1 and SD 1.5, the SSC correction estimation is done using 1000 trajectories generated by a teacher solver. For FLUX.1 the teacher solver is a 35 steps DPM++ (2M) solver, and for SD 1.5 it is a 48 step DPM++ (2M) solver. For all experiments but SD 1.5 DyWeight, the teacher solver's timesteps are a finer version of the student solver's timesteps. For SD 1.5 DyWeight, we use interpolation of teacher solver statistics from the closest timesteps. BézierFlow and DyWeights use shifted timesteps for model input. As these clash with our correction, we do not shift the timesteps when combining BézierFlow and DyWeights with SSC.

Excluding text encoding and VAE decoding, FLUX.1 image generation with a 5 step Euler solver with and without SSC takes 1.47 seconds, proving that SSC has negligible inference overhead.



Figure 5: Comparison between images sampled from FLUX.1 [dev] with and without SSC, for several fast sampling methods (TORS, BézierFlow and DyWeight). Clearer fine details and better contrast at both 5 and 7 NFEs are visible. Prompts are from MS-COCO (Lin et al., 2014).

5.2 Baselines

We evaluate SSC on several families of fast sampling methods. For FLUX.1 [dev], we use the Euler, DPM++ (2M) (Lu et al., 2025), and iPNDM (2M) (Liu et al., 2022a; Zhang & Chen, 2022) solvers. We additionally evaluate TORS (Zhou et al., 2026) and BézierFlow (Min et al., 2026) as representatives for timestep and noise scheduler optimization methods, and DyWeight (Zhao et al., 2026) as a learned solver. For SD 1.5, we evaluate DPM++ (2M), AMED-Plugin (Zhou et al., 2023), and DyWeight.

To disentangle the effect of the solver from that of the timestep discretization, for the non-learned solvers we use the FLUX.1 [dev] default shifted timestep scheduler, and the uniform t scheduler for SD 1.5. For TORS and BézierFlow we use the Euler solver.

For all experiments we use official code bases, and use the official published checkpoints for DyWeight and AMED-Plugin. As the official BézierFlow code lacks a FLUX.1 implementation, we implemented it ourselves on top of the existing official code. Because both SD 3.5 and FLUX.1 are rectified flow models, we mostly used the SD 3.5 setting presented in the paper. For stabler training of BézierFlow, we increased the training and validation sets (a $100, 50$ split instead of $25, 25$), trained for 100 epochs, and initialized from the default FLUX.1 shifted timesteps.

5.3 Evaluation Metrics

Even though several previous works measure FID against MS-COCO images (Tong et al., 2025; Zhao et al., 2026), we observe that lower values do not correlate with image quality. For example, our SD 1.5 teacher solver achieves an FID of 15.29, while the newer FLUX.1 [dev] teacher

Table 1: FLUX.1 [dev] results at 5 and 7 NFE.

		5 NFE			7 NFE		
Method Family	**Method**	**T. FID ↓**	**IR ↑**	**HPSv2.1 ↑**	**T. FID ↓**	**IR ↑**	**HPSv2.1 ↑**
Fixed Solvers	Euler	17.64	0.66	0.261	7.53	0.85	0.281
	Euler **+ SSC**	**9.48**	**0.87**	**0.286**	**4.45**	**0.93**	**0.296**
	DPM++ (2M)	13.10	0.70	0.268	5.53	0.87	0.285
	DPM++ (2M) **+ SSC**	**7.51**	**0.86**	**0.288**	**3.66**	**0.92**	**0.298**
	iPNDM (2M)	15.36	0.71	0.272	4.51	0.87	0.287
	iPNDM (2M) **+ SSC**	**6.99**	**0.87**	**0.289**	**3.29**	**0.93**	**0.297**
Timestep Opt.	BézierFlow	13.99	**0.83**	0.280	4.63	**0.96**	0.295
	BézierFlow **+ SSC**	**11.52**	0.79	0.280	**4.12**	0.92	0.295
	TORS	13.08	0.82	0.272	6.33	0.92	0.289
	TORS **+ SSC**	**7.02**	**0.90**	**0.290**	**3.65**	**0.95**	**0.296**
Learned Solvers	DyWeight	16.48	0.82	0.271	8.52	0.89	0.282
	DyWeight **+ SSC**	**14.62**	**0.90**	**0.282**	**7.41**	**0.96**	**0.293**

Table 2: SD 1.5 Teacher FID

Method	**4 Steps**	**6 Steps**	**8 Steps**
DPM++ (2M)	12.87	4.73	2.82
DPM++ (2M) **+ SSC**	**7.83**	**3.38**	**2.31**
AMED-Plugin	12.29	4.18	2.33
AMED-Plugin **+ SSC**	**9.67**	**3.76**	**2.29**
DyWeight	10.65	3.87	3.00
DyWeight **+ SSC**	**8.50**	**2.97**	**2.48**

solver achieves a significantly worse FID of 26.35. Instead, we evaluate FID against generated teacher images, as previously used in prior works, for example by Cai et al. (2026). In addition, for FLUX.1 [dev] we measure image quality with ImageReward (Xu et al., 2023) and HPSv2.1 (Wu et al., 2023). MS-COCO FID results can be found in App. A.5.

5.4 Results

Figure 1, Fig. 4 and Fig. 5 present qualitative examples of SSC results, showing that for the tested methods SSC improves both global image statistics, such as contrast, and local statistics, *i.e.* fine high frequency details.

Results of our FLUX.1 experiments are shown in Table 1, and SD 1.5 results are shown in Table 2. SSC improves almost all methods, in both FID, ImageReward and HPSv2.1. BézierFlow is the only exception, but SSC still manages to improve its FID. Surprisingly, SSC by itself, *i.e.* Euler+SSC, is a very strong method by itself, surpassing many of the considered methods.

Appendix A.4 reports an ablation quantifying the improvement provided by the learned correction, as a function of the number of learned directions, over using the DCT correction alone. For FLUX.1, the learned correction provides a non-negligible improvement, which peaks at around $k = 40$ learned directions. For SD 1.5, the learned correction leads to only a marginal improvement.

6 Conclusion

We introduced a framework for statistical correction of flow and diffusion solvers. Our Spectral Statistical Correction (SSC) introduces a negligible overhead, and can be efficiently estimated. On FLUX.1 experiments, SSC achieves substantial improvement by itself. Furthermore, it can be combined with other fast sampling methods, improving their results even further.

7 AI USE STATEMENT

AI was used in implementing parts of the code in the paper (codex), and for writing assistance, mainly for improving phrasings (ChatGPT). All AI generated code was checked for correctness.

REFERENCES

John Charles Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.

Xu Cai, Yang Wu, Qianli Chen, Haoran Wu, Lichuan Xiang, and Hongkai Wen. Shortcutting pre-trained flow matching diffusion models is almost free lunch. *Advances in Neural Information Processing Systems*, 38:153066–153094, 2026.

Defang Chen, Zhenyu Zhou, Can Wang, Chunhua Shen, and Siwei Lyu. On the trajectory regularity of ode-based diffusion sampling. *arXiv preprint arXiv:2405.11326*, 2024.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. *arXiv preprint arXiv:2209.11215*, 2022.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Eric Frankel, Sitan Chen, Jerry Li, Pang Wei Koh, Lillian J. Ratliff, and Sewoong Oh. S4s: Solving for a fast diffusion model solver. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=9OaZCNbV2w.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pp. 8867–8887. PMLR, 2022.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Sihyeon Kim, Seunghun Lee, Vikas Singh, and Hyunwoo J Kim. Learning to solve generative odes beyond the linear span. *arXiv preprint arXiv:2606.08672*, 2026.

Yuya Kobayashi, Masato Ishii, Yuhta Takida, Takashi Shibuya, and Yuki Mitsufuji. Spectral prior for reducing exposure bias in diffusion models. *arXiv preprint arXiv:2607.22091*, 2026.

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

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.

Mingyu Liang, Dingkun Xu, and Jingwei Xu. Trajectory-consistent calibration for cache-accelerated diffusion models. *arXiv preprint arXiv:2605.24870*, 2026.

Hubert W Lilliefors. On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American statistical Association*, 62(318):399–402, 1967.

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 *European conference on computer vision*, pp. 740–755. Springer, 2014.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

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*, 2022a.

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*, 2022b.

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.

Yunhong Min, Juil Koo, Seungwoo Yoo, and Minhyuk Sung. Bézierflow: Learning bézier stochastic interpolant schedulers for few-step generation. In *International Conference on Learning Representations*, volume 2026, pp. 6961–6984, 2026.

Mang Ning, Enver Sangineto, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Input perturbation reduces exposure bias in diffusion models. *arXiv preprint arXiv:2301.11706*, 2023.

Mang Ning, Mingxiao Li, Jianlin Su, Albert Ali Salah, and Itir Onal Ertugrul. Elucidating the exposure bias in diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 15167–15189, 2024.

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 (CVPR)*, pp. 10684–10695, June 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In *SIGGRAPH Asia 2024 Conference Papers*, SA '24, New York, NY, USA, 2024a. Association for Computing Machinery. ISBN 9798400711312. doi: 10.1145/3680528.3687625. URL https://doi.org/10.1145/3680528.3687625.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXXVI*, pp. 87–103, Berlin, Heidelberg, 2024b. Springer-Verlag. ISBN 978-3-031-73015-3. doi: 10.1007/978-3-031-73016-0_6. URL https://doi.org/10.1007/978-3-031-73016-0_6.

Neta Shaul, Juan Perez, Ricky T. Q. Chen, Ali Thabet, Albert Pumarola, and Yaron Lipman. Bespoke solvers for generative flow models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 12510–12546, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/35c1d69d23bb5dd6b9abcd68be005d5c-Paper-Conference.pdf.

Neta Shaul, Uriel Singer, Ricky T. Q. Chen, Matthew Le, Ali Thabet, Albert Pumarola, and Yaron Lipman. Bespoke non-stationary solvers for fast sampling of diffusion and flow models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024b.

Neta Shaul, Uriel Singer, Itai Gat, and Yaron Lipman. Transition matching: Scalable and flexible generative modeling. *Advances in Neural Information Processing Systems*, 38:124347–124388, 2026.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning - Volume 37*, ICML'15, pp. 2256–2265. JMLR.org, 2015.

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.

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

Vinh Tong, Trung-Dung Hoang, Anji Liu, Guy Van den Broeck, and Mathias Niepert. Learning to discretize denoising diffusion odes. In *Proceedings of the 13th International Conference on Learning Representations*, 2025.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. URL https://arxiv.org/abs/2508.02324.

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. *arXiv preprint arXiv:2306.09341*, 2023.

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. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 15903–15935, 2023.

Shuchen Xue, Zhaoqiang Liu, Fei Chen, Shifeng Zhang, Tianyang Hu, Enze Xie, and Zhenguo Li. Accelerating diffusion sampling with optimized time steps. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8292–8301. IEEE, 2024.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623. IEEE, 2024.

Meng Yu and Kun Zhan. Frequency regulation for exposure bias mitigation in diffusion models. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10370–10378, 2025.

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

Tong Zhao, Mingkun Lei, Liangyu Yuan, Yanming Yang, Chenxi Song, Yang Wang, Beier Zhu, and Chi Zhang. Dyweight: Dynamic gradient weighting for few-step diffusion sampling. *arXiv preprint arXiv:2603.11607*, 2026.

Zhenyu Zhou, Defang Chen, Can Wang, and Chun Chen. Fast ode-based sampling for diffusion models in around 5 steps. *arXiv preprint arXiv:2312.00094*, 2023.

Zhenyu Zhou, Defang Chen, Siwei Lyu, Chun Chen, and Can Wang. Analyzing and improving fast sampling of text-to-image diffusion models. *arXiv preprint arXiv:2603.00763*, 2026.

A Appendix

A.1 Gaussianity of DCT Coefficients

We assess the Gaussianity assumption of DCT coefficients of FLUX.1 generated images in both qualitative and quantitative ways. Figure 6 depicts distributions of several DCT coefficients of a wide range of frequencies, showing a relatively good fit to Gaussian PDF. We test the normality of each DCT coefficients using the Lilliefors test (Lilliefors, 1967), which is based on the Kolmogorov–Smirnov test. We select this particular test because its use of of the empirical CDF. With a significant level of 0.05, the Gaussianity of 84% of coefficients can not be rejected, validating our assumption.

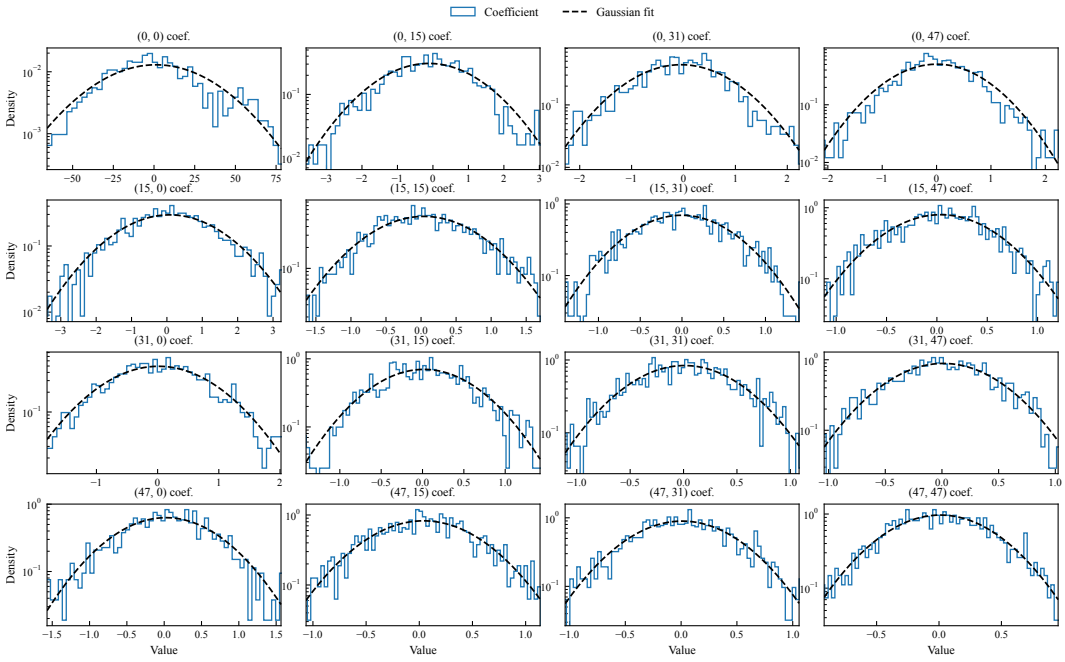


Figure 6: Histograms of channel 0 (out of 16) DCT coefficients of latent space images generated by FLUX.1, with fitted Gaussian PDFs. Central 98% of values are shown.

A.2 Conditional Sampling

Algorithm 1 and Alg. 2 are justified in the case of unconditional sampling. For conditional sampling, each condition c (a prompt in the case of T2I models), induces a different distribution, with condition-dependent mean μ_c and covariance Σ_c. Ideally, the correction should depend on μ_c and Σ_c. Even though model prediction at time $t = 1$ (pure noise) can be used to approximate μ_c, approximating Σ_c requires additional NFEs. Therefore, we estimate a single correction that does not depend on the prompt. In order to estimate a robust correction that generalizes to unseen prompts, we select a different prompt for each latent in Alg. 1.

A.3 Statistical Difference Results

Figure 7 extends Fig. 3 to the entire generation process, showing that the statistical difference is present even after one coarse solver step. The spectra show early low frequency divergence, motivating our use of DCT, as it can naturally modify low frequencies.

Figure 8 depicts that the spectral difference between latent images generated with a 5 and 35 NFE solvers, showing that the channel 0 behavior shown in Fig. 3 extends to all latent channels.

Table 3: FLUX.1 Ablation of Number of Learned Directions.

Correction k	T. FID ↓
$k = 0$ (DCT only)	10.23
$k = 20$	9.60
$k = 40$	**9.51**
$k = 60$	9.61
$k = 80$	9.75
$k = 100$	9.80

Table 4: SD 1.5 Ablation of Number of Learned Directions.

Correction k	T. FID ↓
$k = 0$ (DCT only)	7.67
$k = 20$	7.58
$k = 40$	7.53
$k = 60$	**7.49**
$k = 80$	7.52
$k = 100$	**7.49**

A.4 SSC Ablation

We perform a short ablation for selecting the number k of directions for SSC. We perform a single ablation for FLUX.1 and for SD 1.5. For FLUX.1 the ablation is done on a 5 NFE Euler solver, see Table 3, and for SD 1.5 on a DPM++ (2M) 4 step solver, see Table 3. According to the results of Table 3, we select $k = 40$ for FLUX.1. For SD 1.5, $k = 60$ achieves the lowest teacher FID, but since the difference from a pure DCT correction is quite small, we select $k = 0$.

The effect of the additional learned correction is visible in Fig. 9. While the major correction is done by the DCT correction, the learned correction adds additional quality improvement, visible by more natural colors and better fine details.

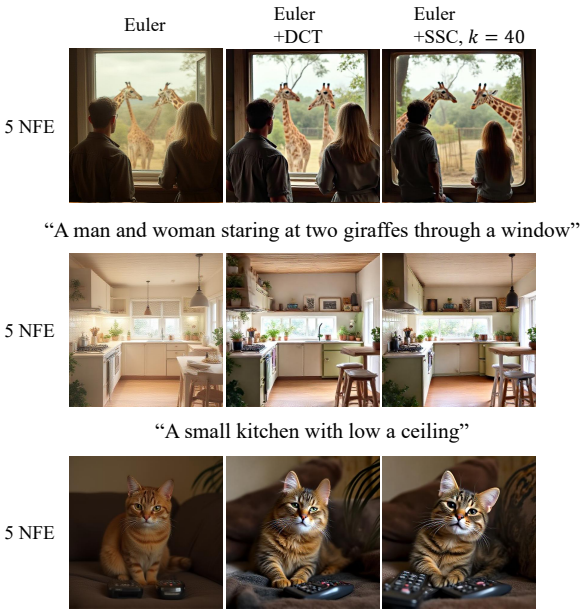


Figure 9: Comparison between images sampled from FLUX.1 [dev] without correction, with a DCT only correction and with full SSC correction. The full SSC correction further improves the DCT correction with better colors and fine details. Prompts are from MS-COCO (Lin et al., 2014).

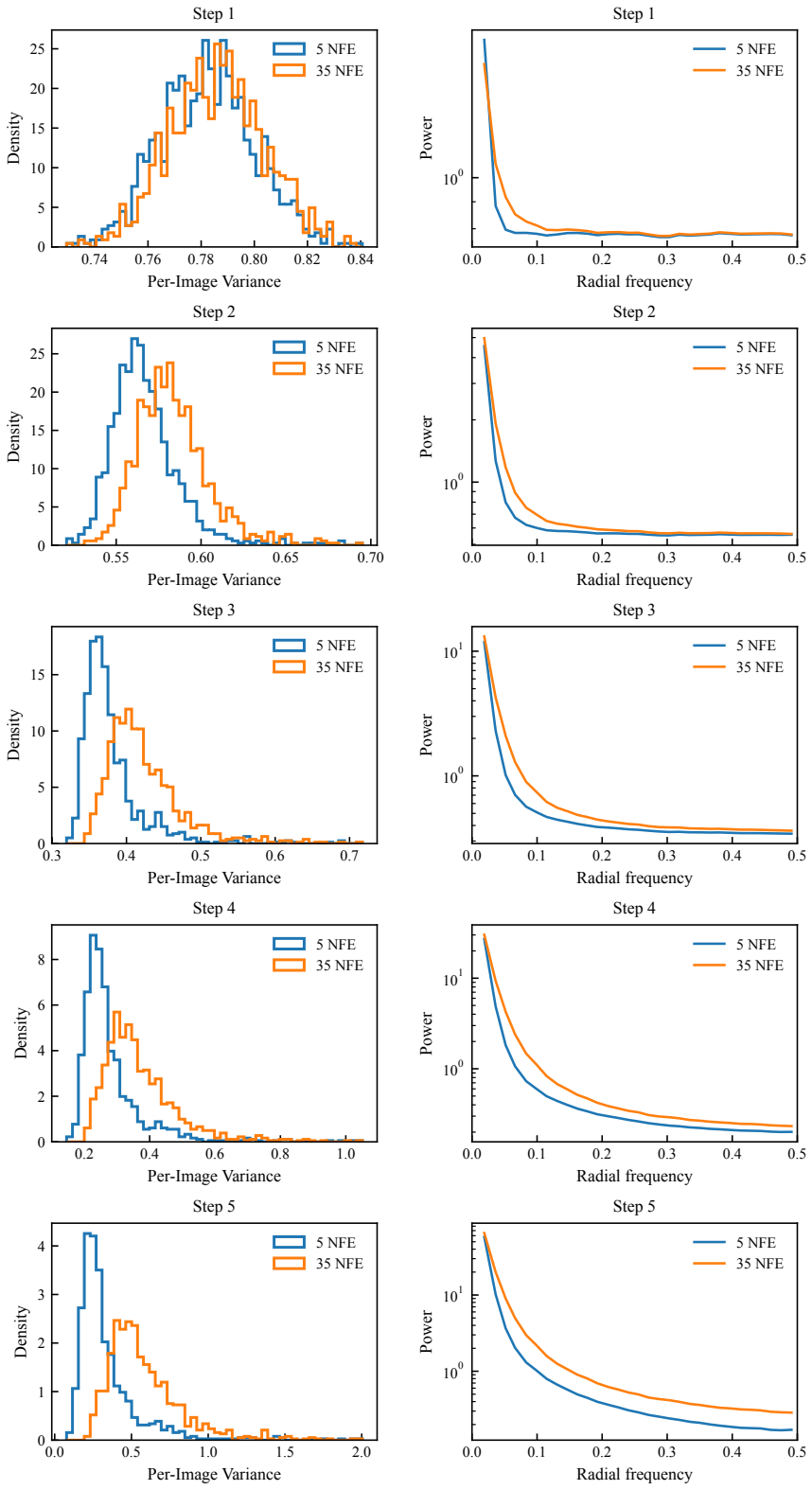


Figure 7: Channel 0 per image variance distribution and radial mean power spectrum of FLUX.1 latent space images generated with 5 and 35 NFE solvers, during the generation process. The i-th row correspond to i steps of the 5 NFE solver, and to $7 \cdot i$ steps of the 35 NFE solver.

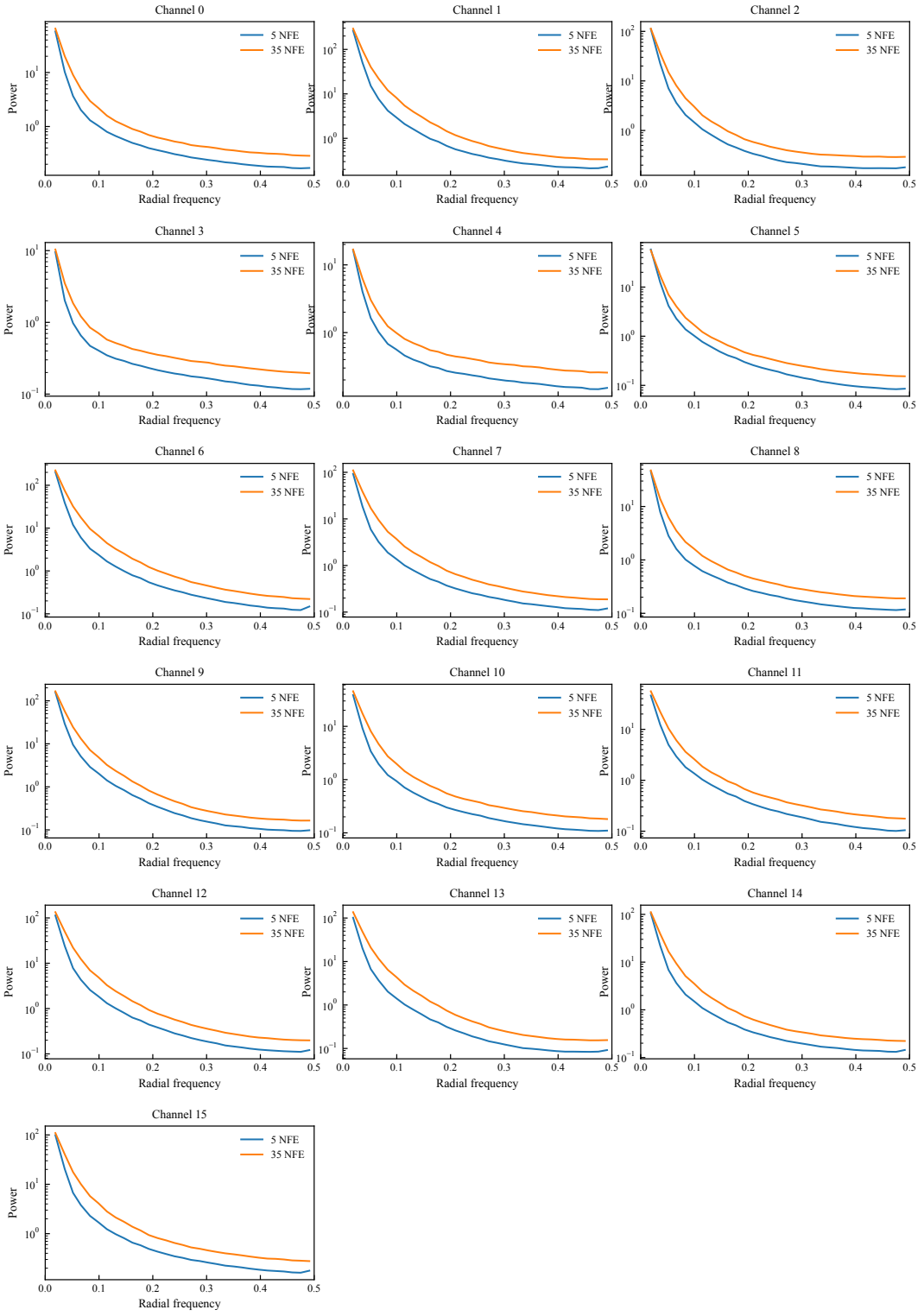


Figure 8: Power Spectra of FLUX.1 generated latent space images with 5 and 35 NFEs. All 16 channels are shown.

A.5 MS-COCO FID RESULTS

We provide FID results for our FLUX.1 and SD 1.5 experiments in Table 5 and Table 6 respectively.

Table 5: FLUX.1 [dev] FID results.

Method Family	Method	5 NFE	7 NFE
Fixed Solvers	Euler	22.04	20.58
	Euler + **SSC**	19.16	21.08
	DPM++ (2M)	20.48	20.75
	DPM++ (2M) + **SSC**	19.46	21.56
	iPNDM (2M)	23.93	20.74
	iPNDM (2M) + **SSC**	19.49	21.73
Timestep Opt.	BézierFlow	20.96	22.30
	BézierFlow + **SSC**	19.61	22.19
	TORS	24.24	23.76
	TORS + **SSC**	20.36	23.50
Learned Solvers	DyWeight	20.43	19.56
	DyWeight + **SSC**	20.21	20.54

Table 6: SD 1.5 FID

Method	4 Steps	6 Steps	8 Steps
DPM++ (2M)	20.57	16.27	15.59
DPM++ (2M) + **SSC**	15.34	14.74	14.89
AMED-Plugin	17.39	13.95	13.92
AMED-Plugin + **SSC**	14.26	13.13	13.76
DyWeight	14.22	12.47	12.67
DyWeight + **SSC**	14.66	14.00	13.05