MEAN-SHIFTED INTERMEDIATE SAMPLING:TRAINING-FREE ACCELERATION FOR FEW-STEP DIFFUSION GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Few-step diffusion sampling suffers from severe discretization errors due to large step sizes. Some existing methods reduce the average step size by starting reverse sampling from an intermediate distribution, but they often require training additional models or rely on real samples during inference. In this paper, we propose Mean-Shifted Intermediate Sampling (MSIS), a simple training-free strategy that approximates the high-noise intermediate distribution with mean-shifted Gaussian noise. The mean shift is determined by the forward noising formula and the dataset itself, so it can be computed once offline and reused across samplers, timestep schedules, and models sharing the same input space, with negligible inference overhead. We further show that near the Gaussian endpoint, the mean shift is the dominant data-dependent deviation, while covariance and higher-order effects are weaker, justifying mean-shifted Gaussian noise as a first-order approximation. Extensive experiments demonstrate the effectiveness and generality of MSIS across datasets, samplers, schedules, and architectures. On conditional ImageNet 256×256, MSIS requires only 2.22 minutes of one-time preprocessing on a single RTX 3090 and reduces FID by 35.18%–55.33% across the evaluated DDIM step budgets. Our code will be made publicly available upon acceptance.

1 INTRODUCTION

Diffusion models (Ho et al., 2020; Song et al., 2020b; Liu et al., 2022) have achieved remarkable success in high-fidelity generation by transforming samples from a standard Gaussian prior into data samples through iterative denoising. In the few-step regime, large step sizes introduce significant discretization errors. To mitigate this issue, numerical solvers (Song et al., 2020a; Lu et al., 2022; Zhao et al., 2023b; Zheng et al., 2023) reduce the discretization error of each individual step, while timestep schedules (Karras et al., 2022; Sabour et al.; Park et al., 2024) reduce the overall error by choosing better discretization points along the reverse trajectory from the Gaussian prior to the data distribution.

Although these methods reduce discretization errors, they still operate along the full reverse sampling path, leaving the average step size large in the few-step regime. A complementary direction is to start sampling from an intermediate distribution, thereby shortening the reverse trajectory and reducing the average step size (Zheng et al., 2022). However, unlike the standard Gaussian prior, this intermediate distribution is not directly accessible for sampling. Existing approaches (Meng et al., 2021; Zheng et al., 2022; Lyu et al., 2022; Zhao et al., 2023a; Park et al., 2026) either train additional models to approximate it or rely on real samples to simulate intermediate states, introducing extra training cost or data dependency.

To overcome these limitations, we propose Mean-Shifted Intermediate Sampling (MSIS), a training-free strategy that approximates the high-noise intermediate distribution with mean-shifted Gaussian noise. MSIS estimates the data mean in the model input space and directly computes its mean shift at any target noise level using the forward noising formula. Since the estimate is tied to the input representation rather than to a particular sampling trajectory, it only needs to be computed once offline and can then be reused across samplers, timestep schedules, and models that operate on the same input space with negligible adaptation cost. Moreover, because MSIS estimates only a first-order

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.

statistic, the mean can be reliably computed from a small subset of the training data, further reducing the offline preprocessing cost, particularly for large-scale datasets. On ImageNet, using only 1% of the training data requires 2.22 minutes on a single RTX 3090 while preserving the effectiveness of MSIS. We further show that near the Gaussian endpoint, the mean shift is the main data-dependent difference in the intermediate distribution, while covariance and higher-order effects are weaker. This explains why a mean-shifted Gaussian noise can serve as a simple first-order approximation. Extensive experiments demonstrate the effectiveness and generality of our method.

Our contributions are summarized as follows:

1. We propose Mean-Shifted Intermediate Sampling (MSIS), a training-free method that initializes diffusion sampling using a mean-shifted Gaussian approximation to a high-noise intermediate distribution.
2. We theoretically show that near the Gaussian endpoint, the conditional mean shift is the leading data-dependent cumulant deviation, while covariance and higher-order non-Gaussian cumulants are of smaller order. This justifies a mean-only approximation as a first-order correction in the high-noise regime.
3. Extensive experiments across datasets, samplers, timestep selection methods, and model architectures demonstrate the effectiveness and generality of our method. On conditional ImageNet 256×256, MSIS requires only 2.22 minutes of one-time preprocessing on a single RTX 3090 and reduces FID by 35.18%–55.33% across the evaluated DDIM step budgets.

2 Related Work

Intermediate Sampling. Many works have explored starting generation from intermediate states rather than the standard Gaussian prior to shorten the reverse diffusion trajectory. SDEdit (Meng et al., 2021) adds noise to a user-provided guide image and runs the reverse SDE from the resulting noisy state for image synthesis and editing. TDPM/TLDM (Zheng et al., 2022) explicitly truncate the diffusion chain and learn an implicit generative distribution to approximate the noisy-data distribution at the truncation point. ES-DDPM (Lyu et al., 2022) similarly adopts an early-stopped diffusion process, using samples from a pretrained GAN or VAE and diffusing them to the target intermediate state. For conditional generation, PartDiff (Zhao et al., 2023a) observes that the intermediate latents of low- and high-resolution images gradually become similar, and uses the low-resolution latent as a proxy for the high-resolution intermediate state. Recently, AC-Sampler (Park et al., 2026) samples at intermediate timesteps with Metropolis–Hastings correction and a time-dependent discriminator to correct distributional bias. These methods show that intermediate sampling can effectively shorten the generation trajectory and improve few-step sampling efficiency.

Numerical Solvers and Time-step Selection. A major direction for accelerating diffusion sampling is to reduce discretization errors under a limited number of reverse steps. Numerical solvers focus on improving the accuracy of each individual step. DDIM (Song et al., 2020a) enables efficient deterministic sampling through a non-Markovian reverse process, while subsequent ODE/SDE-based samplers (Lu et al., 2022; Zhao et al., 2023b; Zheng et al., 2023) further improve few-step generation with high-order or predictor-corrector updates. Complementarily, time-step selection methods aim to reduce the overall accumulated error by choosing better discretization points along the reverse trajectory. EDM (Karras et al., 2022) proposes an effective noise-level discretization schedule within a unified design space, and recent works further optimize sampling schedules (Sabour et al.; Park et al., 2024) for different solvers, models, or data modalities. Together, these methods significantly improve few-step sampling efficiency by refining either the numerical update rule or the placement of sampling steps.

Noise Offset and Terminal-SNR Correction. Prior work has recognized that highly noisy states may retain low-frequency or mean information and that commonly used schedules can create a train–inference mismatch when the terminal signal-to-noise ratio is nonzero (Lin et al., 2024). Offset Noise adds a spatially shared noise component to improve low-frequency modeling and has also been adopted in large-scale latent diffusion systems such as SDXL (Guttenberg, 2023; Podell et al., 2023). One More Step (OMS) (Hu et al., 2024) instead appends a zero-SNR step and trains an auxiliary

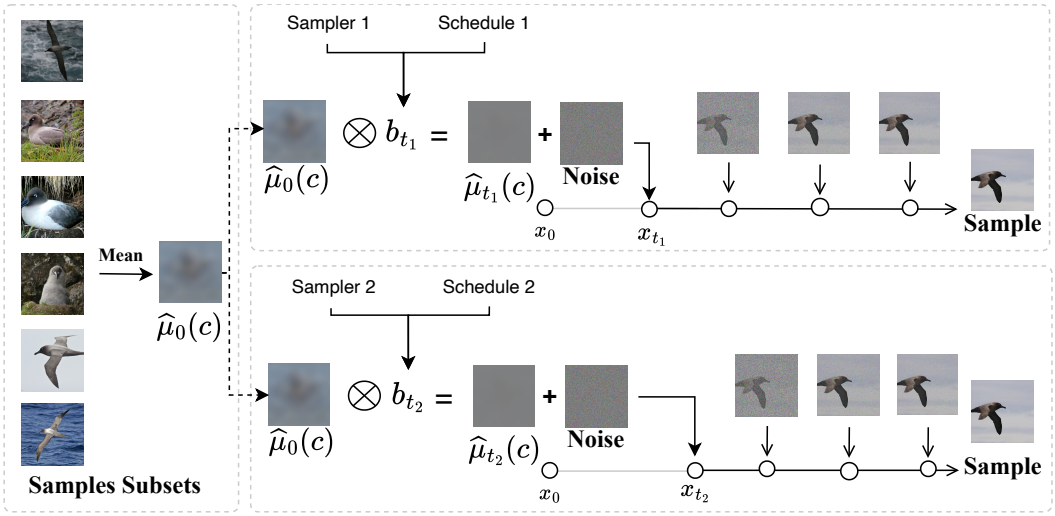

Figure 1: Overall framework of our method. We propose a mean-shifted Gaussian-noise approximation to the intermediate distribution. Left: the dataset mean is computed only once offline. Right: by scaling the mean with the coefficient b_t, the approximation can be instantiated at any timestep t near the Gaussian prior and reused with negligible adaptation cost.

conditional module to reconstruct the terminal state expected by a pretrained model. These approaches primarily focus on correcting the train–inference mismatch at or near the diffusion endpoint. In contrast, MSIS accelerates sampling by approximating a high-noise intermediate distribution in a training-free manner.

3 METHOD

Figure 1 illustrates our method. Instead of sampling from the standard Gaussian prior, we start from a mean-shifted intermediate state. For a condition c and starting timestep t_s, we precompute the class-conditional data mean $\widehat{\mu}_1(c)$ and scale it as $\widehat{\mu}_{t_s}(c) = b_{t_s}\widehat{\mu}_1(c)$. We then initialize the sampler with $x_{t_s}^{\text{sim}} = \widehat{\mu}_{t_s}(c) + z$, where $z \sim \mathcal{N}(0, I_d)$. This shortens the reverse trajectory and reduces the average step size in few-step sampling. Since the mean is computed offline, the inference overhead is negligible. Section 3.1 justifies this mean-shifted approximation, and Section 3.2 describes the sampling procedure.

3.1 THEORETICAL ANALYSIS

We first give a theoretical analysis to explain why a mean-shifted intermediate sampler is reasonable. Following the one-sided spatially-linear formulation (Albergo et al., 2023), the marginal paths of many diffusion models (Ho et al., 2020), score-based models (Song et al., 2020b), and flow models (Liu et al., 2022) can be written in the following unified form:

$$x_t^\star = a_t x_0 + b_t x_1, \tag{1}$$

where $x_0 \sim \mathcal{N}(0, I_d)$ is the Gaussian base variable, $x_1 \sim p_{\text{data}}(\cdot \mid c)$ is the data variable under condition c, and $x_0 \perp x_1 \mid c$. We also assume that the distribution of x_0 does not depend on c.

We focus on the high-noise region near the Gaussian endpoint $t = 0$. In this section, all asymptotic statements are for a fixed condition c, unless we state otherwise. In this region, we assume

$$\lim_{t\to 0} b_t = 0, \qquad 0 < a_{\min} \le a_t \le a_{\max} < \infty \quad \text{for } t \text{ sufficiently close to } 0, \tag{2}$$

where $a_{\min}$ and $a_{\max}$ are constants independent of t. This means that near the Gaussian endpoint, the Gaussian noise part stays well-scaled and dominant, while the data part only appears as a small

perturbation. The original VE diffusion path is not directly written in this normalized form, but it can be converted to Eq. equation 1 by a deterministic and invertible normalization; see Appendix B.1 for details.

Our goal is to build a simulated intermediate variable x_t^{sim} that approximates $x_t^\star$ without using real data samples during inference. Since x_t^{sim} lies in either image space or VAE latent space, we assume that the related distributions are multivariate moment-determinate. In other words, their mixed moments uniquely determine their distributions (Schmüdgen et al., 2017). This assumption holds automatically for bounded image variables with compact support. For latent variables, it can be treated as a standard regularity assumption, for example when their tails are sufficiently light.

We use a standard fact from the classical moment problem: for a multivariate moment-determinate distribution, its mixed raw moments uniquely determine the distribution (Schmüdgen et al., 2017). Mixed raw moments and cumulants can be converted to each other through the moment-cumulant formula. Therefore, if all moments exist, matching cumulants of all orders is the same as matching mixed raw moments of all orders. Under the moment-determinacy assumption, this is enough to determine the distribution. Based on this, we study the approximation error by looking at the orders of cumulant differences.

Theorem 1 (Mean shift is first-order near the Gaussian endpoint)**.** *Fix a condition c, and assume Eq. equation 2. Let $\mu_1(c) := \mathbb{E}[x_1 \mid c]$ and $\Sigma_1(c) := \mathrm{Cov}(x_1 \mid c)$. If $x_1 \mid c$ has finite moments up to order $R \geq 3$, then for the one-sided path $x_t^\star = a_t x_0 + b_t x_1$, we have*

$$\mathbb{E}[x_t^\star \mid c] = b_t \mu_1(c) = O(b_t), \tag{3}$$

and

$$\mathrm{Cov}(x_t^\star \mid c) - a_t^2 I_d = b_t^2 \Sigma_1(c) = O(b_t^2). \tag{4}$$

Moreover, for every fixed $3 \leq r \leq R$, the r-th order conditional cumulant satisfies

$$\kappa_r(x_t^\star \mid c) = b_t^r \kappa_r(x_1 \mid c) = O(b_t^r). \tag{5}$$

Therefore, near the Gaussian endpoint, the conditional mean shift is the only data-dependent term of order $O(b_t)$. The covariance change is only $O(b_t^2)$, and higher-order non-Gaussian effects are $O(b_t^3)$ or smaller.

The proof is given in Appendix B.2. Theorem 1 shows that, near the Gaussian endpoint, the mean shift is the main data-dependent difference. The covariance difference is smaller, of order $O(b_t^2)$, and higher-order non-Gaussian cumulants are even smaller, of order $O(b_t^3)$ or below. Therefore, in a suitable high-noise region, matching only the mean captures the first-order data-dependent difference of the intermediate distribution.

In our sampler, we use the same Gaussian noise scale as the standard prior of the pretrained model. Since both x_0 and $a_t x_0$ have zero mean, the conditional mean shift discussed above is not affected by this Gaussian scale choice. However, keeping the standard prior scale is important, because reducing the noise variance can hurt the initialization quality.

The cumulant hierarchy shows that, near the Gaussian endpoint, the dominant data-dependent difference between the true intermediate distribution and its Gaussian component is the conditional mean shift. Therefore, when $\mu_1(c)$ is nonzero, the leading nonzero data-dependent term in powers of b_t is the conditional mean shift. If $\mu_1(c)$ is zero, the covariance term becomes the leading remaining difference. Thus, our mean-shifted approximation is most directly justified in high-noise regions where the conditional mean carries useful information.

The hierarchy above suggests a simple mean-only approximation. Ideally, we can simulate the intermediate state by adding the estimated conditional mean shift to the Gaussian part of the theoretical path:

$$\widetilde{x}_t^{\mathrm{sim}} = \widehat{\mu}_t(c) + a_t z, \qquad z \sim \mathcal{N}(0, I_d), \qquad z \perp x_1 \mid c, \tag{6}$$

where $\widehat{\mu}_t(c)$ estimates the conditional intermediate mean $\mu_t(c) = b_t \mu_1(c)$. This simulator matches the Gaussian part $a_t z$ of the theoretical path and only adds the estimated mean shift. Its covariance is therefore

$$\mathrm{Cov}(\widetilde{x}_t^{\mathrm{sim}} \mid c) = a_t^2 I_d. \tag{7}$$

In practice, however, the direct simulator in Eq. equation 6 only keeps the Gaussian covariance $a_t^2 I_d$ and drops the data-dependent covariance around the intermediate mean. The true intermediate

distribution contains an additional covariance term from the data component, but this term is not included in a mean-only approximation. As a result, using $a_t z$ may reduce the total variation of the initialized state, especially when $a_t^2 < 1$, which may lead to over-smoothed samples or worse generation quality (Yuan et al., 2025). Therefore, we use a variance-preserving version:

$$x_t^{\text{sim}} = \widehat{\mu}_t(c) + z, \qquad z \sim \mathcal{N}(0, I_d), \qquad z \perp x_1 \mid c. \tag{8}$$

That is, we keep the standard Gaussian noise scale and only shift its mean. Compared with the sampler's original Gaussian initialization z, our simulator differs only in the mean:

$$\mathbb{E}[x_t^{\text{sim}} \mid c] = \widehat{\mu}_t(c), \qquad \text{Cov}(x_t^{\text{sim}} \mid c) = I_d. \tag{9}$$

This practical version keeps the standard Gaussian scale used by the sampler and adds the leading mean shift when $\mu_1(c) \neq 0$. It is not designed to match the full covariance of $x_t^\star$. The covariance difference between x_t^{sim} and $x_t^\star$ is

$$\text{Cov}(x_t^{\text{sim}} \mid c) - \text{Cov}(x_t^\star \mid c) = I_d - a_t^2 I_d - b_t^2 \Sigma_1(c). \tag{10}$$

If the path is chosen or normalized such that $a_t^2 = 1 + O(b_t^2)$ in the high-noise region, then this covariance difference is also $O(b_t^2)$. Otherwise, part of the difference comes from the Gaussian scale used by the sampler, which is independent of the data. In both cases, the leading data-dependent term that we explicitly model is still the conditional mean shift.

Thus, x_t^{sim} should be viewed as a variance-preserving implementation of the mean-shifted approximation: it keeps the standard Gaussian prior scale expected by the pretrained sampler and adds only the estimated conditional mean shift, without estimating class-dependent covariance or higher-order moments.

3.2 Mean Estimation and Sampling Procedure

The analysis in Section 3.1 suggests that, in the high-noise region, the dominant data-dependent correction is the conditional mean shift. From Eq. equation 3, the intermediate mean is

$$\mu_t(c) = \mathbb{E}[x_t^\star \mid c] = b_t \mu_1(c), \tag{11}$$

where b_t is determined by the forward noising path and $\mu_1(c)$ is the data mean under condition c. Thus, estimating the intermediate mean only requires the data mean in the model input space. It does not depend on the denoising network, the sampler, or the reverse trajectory. Once computed, the mean can be reused across samplers, timestep schedules, and models sharing the same input space; changing the starting timestep only rescales it by b_t.

Offline mean computation. We compute the data mean once offline. For class-conditional generation, the mean is computed separately for each class; for unconditional generation, the entire dataset is treated as one class. For pixel-space models, we use $\widehat{\mu}_1(c) = |\mathcal{D}_c|^{-1} \sum_{x_i \in \mathcal{D}_c} x_i$. For latent diffusion models, we first encode images with the VAE encoder and then average the resulting latents as $\widehat{\mu}_1(c) = |\mathcal{D}_c|^{-1} \sum_{x_i \in \mathcal{D}_c} E_{\text{VAE}}(x_i)$, where E_{VAE} denotes the VAE encoder. This computation requires only one streaming pass over the dataset and does not involve training, denoising, inversion, or storing all samples. The running mean can be updated as

$$\widehat{\mu}_1^{(n+1)}(c) = \widehat{\mu}_1^{(n)}(c) + \frac{y_{n+1} - \widehat{\mu}_1^{(n)}(c)}{n+1}, \tag{12}$$

where $y_{n+1} = x_{n+1}$ for pixel-space models and $y_{n+1} = E_{\text{VAE}}(x_{n+1})$ for latent-space models. The estimator can also be computed from a random subset $\mathcal{D}_c^{(\rho)} \subseteq \mathcal{D}_c$, where $\rho \in (0, 1]$ is the fraction of training samples retained.

Sampling with the precomputed mean. Given a starting timestep t_s in the high-noise region, we scale the precomputed data mean by the forward coefficient,

$$\widehat{\mu}_{t_s}(c) = b_{t_s} \widehat{\mu}_1(c), \tag{13}$$

and initialize sampling with

$$x_{t_s}^{\text{sim}} = \widehat{\mu}_{t_s}(c) + z, \qquad z \sim \mathcal{N}(0, I_d). \tag{14}$$

The sampler then runs from t_s to the data endpoint. Therefore, inference differs from standard sampling only by adding a precomputed mean shift to the initial Gaussian noise, introducing negligible overhead.

Open-ended and unconditional generation. For models without a finite set of conditions, including text-to-image and unconditional models, we treat the entire training set as a single class and compute one global mean in the model input space. For text-to-image generation, this global mean is shared across all text prompts; for unconditional generation, it is used for all generated samples. Therefore, MSIS does not require constructing prompt-specific subsets or estimating a separate mean for each prompt, and the same sampling procedure applies without modification.

Discretization gap. The mean in Eq. equation 13 comes from the ideal forward marginal, whereas practical few-step samplers follow a discretized reverse trajectory. One could estimate the actual trajectory mean by inverting real samples to t_s, but this is unreliable in the few-step regime because large step sizes can introduce substantial inversion errors. We therefore use the analytic forward mean, which is stable, sampler-independent, and does not require real samples during inference.

4 EXPERIMENTS

Experimental setting. Following prior work (Park et al., 2026), we conduct experiments on a wide range of pretrained diffusion models. These models are trained on CIFAR-10 (Krizhevsky et al., 2009), ImageNet 64×64 and 256×256 (Deng et al., 2009), and CelebA-HQ 256×256 (Karras et al., 2017), covering both U-Net-based (Ronneberger et al., 2015) and Transformer-based (Vaswani et al., 2017) architectures. We additionally evaluate zero-shot text-to-image generation with Stable Diffusion 2.1 on the MS-COCO 256×256 30K protocol (Kim et al., 2024). We use 30K prompts from the MS-COCO validation set, generate images at 512×512, and resize them to 256×256 for evaluation. The global latent mean is computed only from MS-COCO training images using the pretrained VAE encoder; captions and validation images are not used for mean estimation, and all Stable Diffusion parameters remain frozen. For sampling, we evaluate our method with various samplers, including Heun, SDE (Karras et al., 2022), KAR1(deterministic), KAR2(stochastic) (Kim & Ye, 2022) , DDIM (Song et al., 2020a), DPM++ (Lu et al., 2022), UniPC (Zhao et al., 2023b) and DPMv3 (Zheng et al., 2023). We also consider different timestep schedules, including linear, logSNR, and Karras (Karras et al., 2022) schedules. Across all datasets, we report FID (Heusel et al., 2017). On ImageNet 256×256, we additionally report sFID (Nash et al., 2021) and IS (Salimans et al., 2016); on MS-COCO, we additionally report IS and CLIP Score. For each setting, we keep the basic experimental configuration consistent with the corresponding baseline. Except for AC-Sampler, DiffRS(Na et al., 2024) and the CelebA-HQ baselines, all other results are re-test on our servers following the original settings to ensure a fair comparison. The starting point is selected once for each pretrained-model–timestep-schedule combination and then shared across the corresponding numerical samplers, so improvements are not obtained through sampler-specific tuning. For all main experiments, the dataset statistics are estimated from a randomly selected 1% ($\rho = 0.01$) subset of the training data, sampled within each class for conditional datasets. Detailed starting points for every dataset and step budget, together with the mean-estimation protocol, random seed, and hardware configuration, are provided in Appendix A.

4.1 MAIN RESULTS

Results on CIFAR-10 and ImageNet 64×64. The upper-left and upper-right blocks of Table 1 report the results on CIFAR-10 and conditional ImageNet 64×64, respectively. On CIFAR-10, our method consistently improves FID across different few-step samplers under the same NFE budgets, with the largest gains in the extremely low-step regime. At NFE $= 6$, it achieves FIDs of 8.06, 6.56, and 5.19 for DPM++, UniPC, and DPM-v3, respectively. For EDM-Heun, it achieves the best FID at larger budgets, reaching 2.04 and 1.96, while remaining competitive with AC at the smallest budget. On ImageNet 64×64, our method consistently improves EDM-SDE under all NFE budgets, achieving FIDs of 4.25, 2.56, and 2.15, respectively.

Results on CelebA-HQ 256×256. The middle-right block of Table 1 reports the results on unconditional CelebA-HQ 256×256. For a fair and consistent comparison across datasets, the AC-Sampler results are taken from their original paper under the same AC-Sampler setting used for the other datasets, rather than from the CelebA-HQ-specific variant tailored to unconditional generation. For ScoreSDE-KAR1, our method achieves substantial improvements, reducing FID from 121.27, 122.74, and 125.15 to 32.41, 28.49, and 28.71, respectively, and also clearly outperforms

Table 1: Main results on unconditional CIFAR-10, unconditional CelebA-HQ 256×256, conditional ImageNet 64×64, and conditional ImageNet 256×256. All +Ours results use statistics estimated with $\rho = 0.01$. The best results are highlighted in bold.

Unconditional CIFAR-10

Method	FID↓	NFE↓	FID↓	NFE↓	FID↓	NFE↓
EDM (Heun)	3.23	17	2.15	23	2.01	27
+DiffRS	7.79	19.87	3.26	23.13	2.17	28.15
+AC	**2.38**	15.81	2.10	22.78	1.97	26.19
+Ours	2.49	17	**2.04**	23	**1.96**	27
DPM++	12.07	6	4.56	8	3.00	10
+Ours	**8.06**	6	**4.01**	8	**2.91**	10
UniPC	13.02	6	4.37	8	3.07	10
+Ours	**6.56**	6	**3.69**	8	**2.79**	10
DPM-v3	8.66	6	3.58	8	2.59	10
+AC	7.12	5.61	3.09	7.53	2.54	9.88
+Ours	**5.19**	6	**2.92**	8	**2.45**	10

Conditional ImageNet 64×64

Method	FID↓	NFE↓	FID↓	NFE↓	FID↓	NFE↓
EDM (SDE)	5.33	21	2.77	41	2.30	61
+AC	–	–	–	–	2.25	58.75
+Ours	**4.25**	21	**2.56**	41	**2.15**	61

Unconditional CelebA-HQ 256×256

Method	FID↓	NFE↓	FID↓	NFE↓	FID↓	NFE↓
ScoreSDE (KAR1)	121.27	40	122.74	98	125.15	198
+AC(fixed t)	103.81	19.78	75.51	41.83	87.40	84.13
+Ours	**32.41**	20	**28.49**	40	**28.71**	80
ScoreSDE (KAR2)	83.21	40	57.28	98	29.74	198
+AC (fixed t)	45.74	19.55	**19.48**	42.97	**9.45**	94.34
+Ours	**28.35**	20	27.45	40	25.99	80

Conditional ImageNet 256×256

Method	Schedule	NFE=6			NFE=8			NFE=10		
		FID↓	sFID↓	IS↑	FID↓	sFID↓	IS↑	FID↓	sFID↓	IS↑
DDIM	Linear	20.37	23.52	127.77	10.02	14.37	179.82	6.51	11.35	209.61
+Ours		**9.10**	**6.76**	**180.33**	**5.56**	**5.68**	**211.15**	**4.22**	**5.57**	**227.75**
DPM++	logSNR	13.14	9.69	143.62	5.74	6.29	192.10	4.17	6.51	215.71
+Ours		**7.67**	**5.75**	**174.59**	**4.36**	**5.75**	**208.25**	**3.32**	**5.58**	**227.00**
UniPC	logSNR	10.70	7.41	152.83	5.21	6.63	196.73	4.01	7.09	219.91
+Ours		**6.80**	**6.05**	**179.70**	**4.11**	**6.41**	**212.28**	**3.09**	**5.76**	**230.21**

AC-Sampler under comparable NFE budgets. For ScoreSDE-KAR2, our method provides a strong improvement over the baseline in the lowest-budget setting, reducing FID from 83.21 to 28.35, but performs worse than AC-Sampler at larger budgets. A possible explanation is that the aligned face images in CelebA-HQ form a more concentrated and anisotropic distribution in image space. As a result, the covariance structure of the true intermediate distribution may be less well matched by the isotropic Gaussian carrier used in our mean-shifted approximation, making a mean-only correction less sufficient.

Results on ImageNet 256×256. The bottom block of Table 1 reports the results on conditional ImageNet 256×256, where all methods use classifier-free guidance scale $w = 1.5$. Our method consistently improves FID and sFID across DDIM, DPM++, and UniPC under the same NFE budgets. At NFE $= 6$, it achieves FIDs of 9.10, 7.67, and 6.80 for DDIM, DPM++, and UniPC, respectively. MSIS consistently improves FID, sFID, and IS for all three samplers and all evaluated NFE budgets, indicating better fidelity, spatial quality, and conditional sample quality in high-resolution generation.

Results on MS-COCO text-to-image generation. Table 2(a) reports zero-shot text-to-image results with Stable Diffusion 2.1. MSIS reduces FID from 19.36 to 17.23 at four steps, from 16.39 to 14.37 at six steps, and from 15.25 to 14.48 at eight steps. IS and CLIP Score remain comparable or improve, showing that the prompt-independent global mean improves distributional quality without weakening text–image alignment. These results demonstrate that MSIS extends naturally to open-ended conditioning spaces without estimating a separate mean for every prompt.

Table 2: Additional quantitative results and ablations. (a) Zero-shot text-to-image generation on the MS-COCO 256×256 30K protocol. (b) Component ablation on conditional ImageNet 256×256. (c) One-time ImageNet mean-estimation cost. (d) Sensitivity of sampling performance to the training-data fraction ρ used for mean estimation.

(a) MS-COCO zero-shot text-to-image

NFE	Method	FID ↓	IS ↑	CLIP ↑
4	DDIM	19.36	27.28	0.2948
	+Ours	**17.23**	**27.82**	**0.2989**
6	DDIM	16.39	**31.10**	0.3075
	+Ours	**14.37**	30.78	**0.3076**
8	DDIM	15.25	32.45	0.3105
	+Ours	**14.48**	**32.72**	**0.3108**

(b) Component ablation

Method	FID ↓	sFID ↓	IS ↑
DDIM	20.37	23.52	127.77
+ IG	15.89	8.06	134.75
+ MSIS	9.10	6.76	180.33

(c) Estimation cost

ρ	Min.
1.00	147.45
0.50	72.55
0.01	2.22

(d) Data-fraction sensitivity

NFE	ρ	ImageNet 256×256			CIFAR-10
		FID ↓	sFID ↓	IS ↑	FID ↓
6	1.00	9.36	6.93	179.18	6.580
	0.50	9.34	6.94	179.34	6.583
	0.01	9.10	6.76	180.33	6.561
8	1.00	5.68	5.74	208.20	3.695
	0.50	5.66	5.74	208.20	3.692
	0.01	5.56	5.68	211.15	3.692
10	1.00	4.28	5.54	225.93	2.798
	0.50	4.27	5.54	225.61	2.795
	0.01	4.22	5.57	227.75	2.794

4.2 Ablation Studies

Unless otherwise specified, all ablations on conditional ImageNet 256×256 use DDIM at NFE $= 6$, while those on unconditional CIFAR-10 use UniPC. All remaining settings follow the corresponding main experiments.

Component ablation. Table 2(b) compares the standard baseline, an intermediate Gaussian variant (IG), and the full MSIS method. IG moves the starting point to $T = 600$ while retaining standard Gaussian initialization, thereby isolating the benefit of shortening the reverse trajectory. MSIS uses the same starting point but additionally introduces the class-conditional mean shift. Trajectory truncation alone reduces FID from 20.37 to 15.89 and sFID from 23.52 to 8.06, confirming that large-step discretization error is a major bottleneck. Adding the mean shift further improves FID to 9.10, sFID to 6.76, and IS to 180.33, showing that it reduces the initialization bias left by the intermediate Gaussian approximation.

Variance and full-covariance ablation. We further compare mean-only matching with diagonal-variance and full-covariance matching across NFE $= 6, 8, 10$. Diagonal-variance matching performs similarly to, and sometimes slightly worse than, mean-only matching, suggesting that independent per-dimension variance adjustment does not capture useful second-order dependencies. Full-covariance matching improves FID and recall in some settings, but the gains are not consistent across all metrics and come with substantially higher storage cost: for 1,000 ImageNet classes, the required FP32 statistics increase from 15.63 MiB for the mean to 62.52 GiB for a dense full covariance. Therefore, mean-only matching provides the best overall effectiveness–efficiency trade-off. Complete results and the storage comparison are provided in Appendix C.

Sensitivity to the starting timestep. We vary t_s from 500 to 700 while keeping all other settings fixed. The results are shown in Figure 2. As t_s increases from 500 to 600, both FID and sFID consistently decrease, while IS increases, indicating that starting from a more suitable intermediate point can better reduce discretization error and improve sample quality. The best overall performance is achieved at $t_s = 600$. When t_s is further increased beyond 600, the performance gradually degrades, suggesting that an overly aggressive truncation may introduce a larger mismatch between the simulated initialization and the true intermediate distribution. These results show that MSIS is robust within a neighborhood of the selected starting point, while also confirming the importance of choosing an appropriate intermediate timestep.

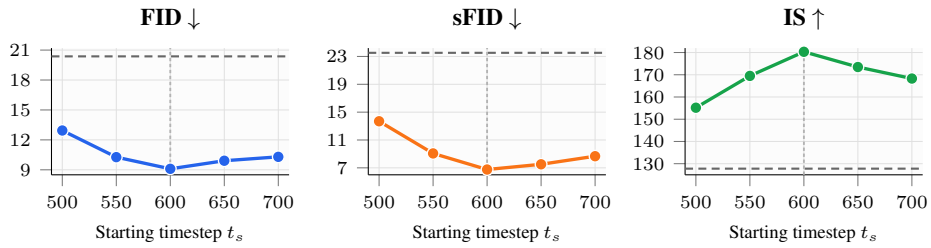

Figure 2: Starting-timestep sensitivity under the default ImageNet ablation setting. Dashed horizontal lines denote the NFE $= 6$ DDIM baselines in Table 2(b); dotted vertical lines mark $t_s = 600$.



Figure 3: Qualitative comparison under the default ImageNet ablation setting.

Sensitivity to the amount of data used for mean estimation. For this ablation, we evaluate all three NFE budgets and recompute the ImageNet class-conditional means and CIFAR-10 global mean using a fraction ρ of the training data. Table 2(d) compares results for $\rho \in \{1.00, 0.50, 0.01\}$. Across all three NFE budgets, performance is nearly unchanged as the estimation set is reduced. In terms of preprocessing cost, AC-Sampler reports less than two hours of training on CIFAR-10 using a single RTX 3090 (Park et al., 2026), whereas estimating the pixel-space global mean for MSIS takes only a few seconds. AC-Sampler does not report the corresponding cost on ImageNet; using 1% of the training data, MSIS requires only 2.22 minutes on a single RTX 3090 to estimate the ImageNet class-conditional means, as reported in Table 2(c).

Qualitative results. Figure 3 shows that the baseline often produces over-smoothed samples with ambiguous boundaries in the extremely low-step regime. MSIS produces clearer shapes, more detailed textures, and better preserved object layouts, consistent with the quantitative improvements in Table 2(b).

5 Conclusions

We propose MSIS, a training-free initialization method for few-step diffusion sampling. MSIS starts from a high-noise intermediate timestep and uses mean-shifted Gaussian noise to approximate the intermediate distribution. Our analysis shows that the mean shift is the leading data-dependent difference near the Gaussian endpoint, and experiments show consistent improvements across datasets, models, and samplers. On conditional ImageNet 256×256, MSIS requires only 2.22 minutes of one-time preprocessing on a single RTX 3090 Ti and reduces FID by 35.18%–55.33% across the evaluated DDIM step budgets.

A current limitation is that the starting point still needs to be manually selected. Although it can be reused within the same model–schedule setting, how to automatically choose a suitable starting point remains an open problem.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools solely to aid and polish writing. Specifically, they were used to improve grammar, clarity, and academic style, and to assist with LaTeX presentation and formatting, such as arranging tables. They were not used for literature retrieval or discovery, research ideation or execution, drafting substantive scientific content, generating synthetic datasets, or proving mathematical claims. All AI-assisted edits were reviewed and verified by the authors. We take full responsibility for the final content of this work, including its text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

(This section is **recommended** and does not count toward the page limit.)

We provide the complete formulation, theoretical motivation, and sampling procedure of MSIS in the Method section. The Experiments section specifies the datasets, pretrained models, samplers, timestep schedules, evaluation metrics, and evaluation protocols used in our study. Appendix A reports the starting points, data fractions, random seed, and hardware settings; Appendix B states the theoretical assumptions and provides detailed derivations and proofs; and Appendix C contains the complete variance and covariance ablations. All datasets and pretrained models used in our experiments are publicly available and cited in the paper. Upon acceptance, we will release the source code and experiment configurations needed to reproduce the mean estimation and sampling procedures.

REFERENCES

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions, 2023. *URL https://arxiv. org/abs/2303.08797*, 3:1, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Nicholas Guttenberg. Diffusion with offset noise. Cross Labs Blog, January 2023. URL `https://www.crosslabs.org/blog/diffusion-with-offset-noise`.

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. In *Advances in Neural Information Processing Systems*, 2017.

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

Minghui Hu, Jianbin Zheng, Chuanxia Zheng, Chaoyue Wang, Dacheng Tao, and Tat-Jen Cham. One more step: A versatile plug-and-play module for rectifying diffusion schedule flaws and enhancing low-frequency controls. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7331–7340, 2024.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

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.

Beomsu Kim and Jong Chul Ye. Denoising mcmc for accelerating diffusion-based generative models. *arXiv preprint arXiv:2209.14593*, 2022.

Bo-Kyeong Kim, Hyoung-Kyu Song, Thibault Castells, and Shinkook Choi. Bk-sdm: A lightweight, fast, and cheap version of stable diffusion. In *European Conference on Computer Vision*, pp. 381–399. Springer, 2024.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 5404–5411, 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.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022.

Zhaoyang Lyu, Xudong Xu, Ceyuan Yang, Dahua Lin, and Bo Dai. Accelerating diffusion models via early stop of the diffusion process. *arXiv preprint arXiv:2205.12524*, 2022.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

Byeonghu Na, Yeongmin Kim, Minsang Park, Donghyeok Shin, Wanmo Kang, and Il-Chul Moon. Diffusion rejection sampling. *arXiv preprint arXiv:2405.17880*, 2024.

Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W. Battaglia. Generating images with sparse representations. In *Proceedings of the International Conference on Machine Learning*, 2021.

Minsang Park, Gyuwon Sim, Hyungho Na, Jiseok Kwak, Sumin Lee, Richard Lee Kim, Donghyeok Shin, Byeonghu Na, Yeongmin Kim, and Il-chul Moon. Ac-sampler: Accelerate and correct diffusion sampling with metropolis-hastings algorithm. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kWl13kRJTQ.

Yong-Hyun Park, Chieh-Hsin Lai, Satoshi Hayakawa, Yuhta Takida, and Yuki Mitsufuji. *Jump Your Steps*: Optimizing sampling schedule of discrete diffusion models. *arXiv preprint arXiv:2410.07761*, 2024.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your steps: Optimizing sampling schedules in diffusion models, 2024. *URL https://arxiv. org/abs/2404.14507*.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, 2016.

Konrad Schmüdgen et al. *The moment problem*, volume 9. Springer, 2017.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

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*, 2020b.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Yunlong Yuan, Yuanfan Guo, Chunwei Wang, Wei Zhang, Hang Xu, and Li Zhang. Freqprior: Improving video diffusion models with frequency filtering gaussian noise. *arXiv preprint arXiv:2502.03496*, 2025.

Kai Zhao, Alex Ling Yu Hung, Kai-Lin Pang, Haoxin Zheng, and Kyunghyun Sung. Partdiff: Image super-resolution with partial diffusion models. *ArXiv*, abs/2307.11926, 2023a. URL https://api.semanticscholar.org/CorpusID:260126007.

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

Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Truncated diffusion probabilistic models and diffusion-based adversarial auto-encoders. *arXiv preprint arXiv:2202.09671*, 2022.

Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. Dpm-solver-v3: Improved diffusion ode solver with empirical model statistics. *Advances in Neural Information Processing Systems*, 36: 55502–55542, 2023.

A Implementation Details

The starting point mainly depends on the pretrained model and the timestep schedule, while being relatively insensitive to the choice of numerical sampler. For each model–schedule combination, we determine the starting point once by fitting a linear rule between the number of sampling steps and the corresponding starting time or noise level, and then reuse this configuration for all samplers under that combination. For CIFAR-10, this gives starting points $28, 37, 44$ (sigmas) for EDM-Heun at $9, 12, 14$ steps, respectively, and $18, 24, 30$ (sigmas) for DPM++, UniPC, and DPM-v3 at $6, 8, 10$ steps, respectively. For CelebA-HQ, we set the starting point to 40 for both KAR1 and KAR2 across all evaluated step budgets. For ImageNet 64×64, we use starting points $28, 36, 45$ at $11, 21, 31$ steps, respectively. For DiT on ImageNet 256×256, DDIM uses starting points $600, 650, 700$ at $6, 8, 10$ steps, respectively, while DPM++ and UniPC use starting point 800 for all three step budgets. For Stable Diffusion 2.1 on MS-COCO, DDIM uses starting timesteps $900, 900, 900$ at $4, 6, 8$ steps, respectively. For all main experiments, the statistics are estimated from a randomly selected 1% ($\rho = 0.01$) subset of the training data. For conditional datasets, the subset is sampled independently within each class; for unconditional and open-ended datasets, it is sampled globally. In the data-fraction ablation, we instead vary ρ as specified. We use 0 as the random seed for all experiments to ensure fairness. All experiments are conducted on four RTX 3090 GPUs, except for the statistics-estimation cost measurements, which are conducted on a single RTX 3090.

B Additional Theoretical Details

B.1 Deterministic Normalization of One-sided Paths

We explain how marginal paths that are spatially linear in a Gaussian base variable and a data variable can be converted into the normalized one-sided form used in Section 3.1. Consider a general linear marginal path

$$y_s = a(s)x_0 + b(s)x_1, \qquad x_0 \sim \mathcal{N}(0, I_d), \quad x_1 \sim p_{\mathrm{data}}(\cdot \mid c), \quad x_0 \perp x_1 \mid c. \tag{15}$$

For any deterministic scale $\lambda(s) > 0$, define $\bar{y}_s = \lambda(s)y_s$. Then

$$\bar{y}_s = \lambda(s)a(s)x_0 + \lambda(s)b(s)x_1. \tag{16}$$

Thus, by setting

$$\alpha(s) = \lambda(s)a(s), \qquad \beta(s) = \lambda(s)b(s),$$

the normalized variable can be written as

$$\bar{y}_s = \alpha(s)x_0 + \beta(s)x_1.$$

Since $\lambda(s) > 0$, this operation is an invertible deterministic change of coordinates at each fixed noise level s. Therefore, the normalization does not introduce any additional stochastic approximation; it

only changes the coordinate system in which the marginal path is analyzed. Moment and cumulant orders in the normalized coordinate system should be interpreted with respect to the normalized coefficients $\alpha(s)$ and $\beta(s)$.

For VE diffusion, the original marginal path is usually written as

$$y_\sigma^{\mathrm{VE}} = x_1 + \sigma x_0. \tag{17}$$

Although this form is spatially linear, its data coefficient is fixed to one and therefore does not directly match the normalized convention $b_t \to 0$ near the Gaussian endpoint. Choosing

$$\lambda(\sigma) = \frac{1}{\sqrt{1+\sigma^2}} \tag{18}$$

gives

$$\bar{y}_\sigma^{\mathrm{VE}} = \frac{y_\sigma^{\mathrm{VE}}}{\sqrt{1+\sigma^2}} = \frac{\sigma}{\sqrt{1+\sigma^2}} x_0 + \frac{1}{\sqrt{1+\sigma^2}} x_1. \tag{19}$$

Hence VE can be written in the normalized one-sided spatially-linear form

$$\bar{y}_\sigma^{\mathrm{VE}} = \alpha(\sigma) x_0 + \beta(\sigma) x_1,$$

where

$$\alpha(\sigma) = \frac{\sigma}{\sqrt{1+\sigma^2}}, \qquad \beta(\sigma) = \frac{1}{\sqrt{1+\sigma^2}}.$$

As $\sigma \to \infty$, $\alpha(\sigma) \to 1$ and $\beta(\sigma) \to 0$, so the Gaussian component is non-degenerate and the data component becomes a small perturbation. A monotone reparameterization of σ can then be used to match the time convention in Eq. equation 1.

B.2 Moment-determinacy, Cumulants, and Proof of Theorem 1

Moment-determinacy and cumulant representation. The moment-determinacy statement used in Section 3.1 follows from the classical moment problem (Schmüdgen et al., 2017): for a multivariate moment-determinate distribution, its mixed raw moments uniquely determine the corresponding probability law.

We also use the standard moment-cumulant relations. For every order r, mixed moments of order r can be expressed as polynomials of cumulants of orders at most r, and cumulants of order r can be expressed as polynomials of mixed moments of orders at most r. Hence the full collection of mixed raw moments and the full collection of cumulants contain the same information whenever all moments exist. Therefore, under the moment-determinacy assumption, matching cumulants of all orders is equivalent to matching mixed raw moments of all orders, and thus determines the distribution.

Proof of Theorem 1. Fix a condition c. Conditioned on c, the variables x_0 and x_1 are independent, with $x_0 \sim \mathcal{N}(0, I_d)$. We use standard properties of joint cumulants: cumulants are multilinear in their arguments, homogeneous under deterministic scaling, and additive over independent sums.

For the first cumulant, i.e., the mean, we have

$$\kappa_1(x_t^\star \mid c) = \mathbb{E}[x_t^\star \mid c] = a_t \mathbb{E}[x_0] + b_t \mathbb{E}[x_1 \mid c]. \tag{20}$$

Since $\mathbb{E}[x_0] = 0$, this gives

$$\kappa_1(x_t^\star \mid c) = b_t \mu_1(c), \qquad \mu_1(c) := \mathbb{E}[x_1 \mid c]. \tag{21}$$

Thus $\kappa_1(x_t^\star \mid c) = O(b_t)$.

For the second cumulant, i.e., the covariance, conditional independence gives

$$\kappa_2(x_t^\star \mid c) = \mathrm{Cov}(x_t^\star \mid c) = a_t^2 \mathrm{Cov}(x_0) + b_t^2 \mathrm{Cov}(x_1 \mid c). \tag{22}$$

Since $\mathrm{Cov}(x_0) = I_d$, we obtain

$$\kappa_2(x_t^\star \mid c) = a_t^2 I_d + b_t^2 \Sigma_1(c), \qquad \Sigma_1(c) := \mathrm{Cov}(x_1 \mid c). \tag{23}$$

Therefore,

$$\kappa_2(x_t^\star \mid c) - a_t^2 I_d = b_t^2 \Sigma_1(c) = O(b_t^2). \tag{24}$$

For any order $r \geq 3$, the additivity and homogeneity of cumulants yield

$$\kappa_r(x_t^\star \mid c) = a_t^r \kappa_r(x_0) + b_t^r \kappa_r(x_1 \mid c). \tag{25}$$

Because x_0 is Gaussian, all cumulants of x_0 of order $r \geq 3$ vanish:

$$\kappa_r(x_0) = 0, \qquad r \geq 3.$$

Hence,

$$\kappa_r(x_t^\star \mid c) = b_t^r \kappa_r(x_1 \mid c), \qquad r \geq 3. \tag{26}$$

If $x_1 \mid c$ has finite moments up to order R, then the above identity holds for every fixed $3 \leq r \leq R$. Since $\kappa_r(x_1 \mid c)$ is finite and independent of t, we have

$$\kappa_r(x_t^\star \mid c) = O(b_t^r), \qquad 3 \leq r \leq R.$$

Combining Eqs. equation 21, equation 24, and equation 26 proves the cumulant hierarchy near the Gaussian endpoint.

C Variance and Full-Covariance Ablation

We compare three fitted statistics on conditional ImageNet 256×256: the class-conditional mean (Mean), the mean together with element-wise diagonal variance (Mean + DV), and the mean together with a dense full covariance (Mean + FC). We randomly retain a fraction $\rho \in \{1.00, 0.50, 0.25, 0.10, 0.05, 0.01\}$ of the training images within each class. DDIM is evaluated at NFE $= 6, 8, 10$, using starting timesteps $600, 650, 700$, respectively. Tables 4– 6 report the complete results.

For ImageNet 256×256, the latent shape is $4 \times 32 \times 32$, giving a flattened dimension of $d = 4096$. Table 3 compares the theoretical storage cost of different FP32 class-conditional statistics. A dense full covariance requires more than four thousand times the storage of the mean when statistics are retained for all 1,000 classes.

Table 3: Theoretical FP32 storage cost of fitted statistics for ImageNet 256×256, whose latent dimension is $d = 4096$.

Fitted statistics	Scalars per class	Storage per class	1,000 classes
Mean	4,096	16 KiB	15.63 MiB
Mean + diagonal variance	8,192	32 KiB	31.25 MiB
Mean + full covariance	$4{,}096 + 4{,}096^2$	64.02 MiB	62.52 GiB

Table 4: Complete variance and full-covariance ablation on conditional ImageNet 256×256 with DDIM at NFE $= 6$.

Statistics	ρ	FID ↓	sFID ↓	IS ↑	Precision ↑	Recall ↑
Mean	1.00	9.36	6.93	179.18	0.715	0.506
	0.50	9.34	6.94	179.34	0.715	0.505
	0.25	9.35	6.93	178.42	0.715	0.513
	0.10	9.37	6.92	178.26	0.715	0.508
	0.05	9.33	6.83	178.91	0.717	0.506
	0.01	9.10	6.76	180.33	0.718	0.503
Mean + DV	1.00	9.44	6.97	177.96	0.714	0.498
	0.50	9.44	6.97	178.37	0.715	0.503
	0.25	9.45	6.96	178.38	0.715	0.497
	0.10	9.39	6.96	177.42	0.715	0.498
	0.05	9.39	6.97	177.79	0.716	0.499
	0.01	9.17	6.91	179.69	0.715	0.498
Mean + FC	1.00	8.12	7.82	178.52	0.713	0.541
	0.50	8.13	7.85	178.62	0.713	0.540
	0.25	8.23	7.79	177.58	0.712	0.536
	0.10	8.11	7.88	180.48	0.710	0.537
	0.05	8.14	7.87	179.05	0.717	0.536
	0.01	7.98	8.09	179.60	0.712	0.536

Table 5: Complete variance and full-covariance ablation on conditional ImageNet 256×256 with DDIM at NFE $= 8$.

Statistics	ρ	FID ↓	sFID ↓	IS ↑	Precision ↑	Recall ↑
Mean	1.00	5.68	5.74	208.20	0.769	0.517
	0.50	5.66	5.74	208.20	0.769	0.519
	0.25	5.67	5.74	207.69	0.768	0.519
	0.10	5.67	5.72	208.01	0.769	0.519
	0.05	5.67	5.68	210.69	0.772	0.524
	0.01	5.56	5.68	211.15	0.770	0.517
Mean + DV	1.00	5.72	5.77	209.02	0.772	0.518
	0.50	5.70	5.76	209.62	0.771	0.517
	0.25	5.73	5.77	208.63	0.769	0.520
	0.10	5.74	5.82	207.09	0.769	0.520
	0.05	5.72	5.81	207.79	0.770	0.515
	0.01	5.60	5.77	210.40	0.768	0.514
Mean + FC	1.00	5.11	6.62	207.81	0.762	0.551
	0.50	5.10	6.67	208.02	0.762	0.548
	0.25	5.14	6.57	208.20	0.759	0.545
	0.10	5.08	6.65	211.71	0.758	0.550
	0.05	5.06	6.61	208.58	0.763	0.542
	0.01	5.09	6.79	209.56	0.760	0.545

Table 6: Complete variance and full-covariance ablation on conditional ImageNet 256×256 with DDIM at NFE $= 10$.

Statistics	ρ	FID ↓	sFID ↓	IS ↑	Precision ↑	Recall ↑
Mean	1.00	4.28	5.54	225.93	0.794	0.522
	0.50	4.27	5.54	225.61	0.793	0.523
	0.25	4.27	5.54	225.54	0.793	0.529
	0.10	4.27	5.54	226.18	0.793	0.522
	0.05	4.26	5.49	227.42	0.794	0.522
	0.01	4.22	5.57	227.75	0.796	0.520
Mean + DV	1.00	4.31	5.55	226.41	0.795	0.525
	0.50	4.30	5.55	226.36	0.795	0.524
	0.25	4.30	5.56	226.56	0.795	0.525
	0.10	4.31	5.64	226.12	0.794	0.527
	0.05	4.29	5.63	226.43	0.796	0.521
	0.01	4.27	5.65	227.16	0.795	0.525
Mean + FC	1.00	4.08	6.63	224.68	0.785	0.545
	0.50	4.11	6.64	224.90	0.785	0.548
	0.25	4.10	6.55	224.76	0.785	0.545
	0.10	4.07	6.62	227.42	0.787	0.544
	0.05	4.07	6.62	226.14	0.788	0.542
	0.01	4.11	6.78	224.92	0.785	0.542