One More Step: A Versatile Plug-and-Play Module for Rectifying Diffusion Schedule Flaws and Enhancing Low-Frequency Controls

Minghui Hu[†] Jianbin Zheng[⋆] Chuanxia Zheng[‡] Chaoyue Wang[§] Dacheng Tao[†] Tat-Jen Cham[†]
[†]Nanyang Technological University, [‡]University of Oxford
[⋆]South China University of Technology, [§]The University of Sydney
{e200008, astjcham}@ntu.edu.sg, jabir.zheng@outlook.com, cxzheng@robots.ox.ac.uk

Figure 1. **Example results of our One More Step method on various sceceries.** Traditional sampling methods (Top row) not only lead to (a) generated images converging towards the mean value, but also cause (b) the structure of generated objects to be chaotic, or (c) the theme to not follow prompts. Our proposed *One More Step* addresses these problems effectively *without modifying any parameters* in the pre-trained models. **Avg.** denotes the average pixel value of the images, which are normalized to fall within the range of [0, 1].

Abstract

It is well known that many open-released foundational diffusion models have difficulty in generating images that substantially depart from average brightness, despite such images being present in the training data. This is due to an inconsistency: while denoising starts from pure Gaussian noise during inference, the training noise schedule retains residual data even in the final timestep distribution, due to difficulties in numerical conditioning in mainstream formulation, leading to unintended bias during inference. To mitigate this issue, certain ϵ-prediction models are combined with an ad-hoc offset-noise methodology. In parallel, some contemporary models have adopted zero-terminal SNR noise schedules together with $\mathbf{v}$-prediction, which necessitate major alterations to pre-trained models. However, such changes risk destabilizing a large multitude of community-driven applications anchored on these pre-trained models. In light of this, our investigation revisits the fundamental causes, leading to our proposal of an innovative and principled remedy, called One More Step (OMS). By integrating a compact network and incorporating an additional simple yet effective step during inference, OMS elevates image fidelity and harmonizes the dichotomy between training and inference, while preserving original model parameters. Once trained, various pre-trained diffusion models with the same latent domain can share the same OMS module. Codes and models are released at here.

1. Introduction

Diffusion models have emerged as a foundational method for improving quality, diversity, and resolution of generated images [6, 25], due to the robust generalizability and straightforward training process. At present, a series of open-source diffusion models, exemplified by Stable Diffusion [20], hold significant sway and are frequently cited within the community. Leveraging these open-source models, numerous researchers and artists have either directly adapted [9, 29] or employed other techniques [8] to fine-tune and craft an array of personalized models.

However, recent findings by Karras et al. [10], Lin et al. [13] identified deficiencies in existing noise schedules, leading to generated images primarily characterized by medium brightness levels. Even when prompts include explicit color orientations, the generated images tend to gravitate towards a mean brightness. Even when prompts specify "a solid black image" or "a pure white background", the models will still produce images that are obviously incongruous with the provided descriptions (see examples in Fig. 1). We deduced that such inconsistencies are caused by a divergence between inference and training stages, due to inadequacies inherent in the dominant noise schedules. In detail, during the inference procedure, the initial noise is drawn from a *pure Gaussian distribution*. In contrast, during the training phase, previous approaches such as linear [6] and cosine [17] schedules manifest a non-zero SNR at the concluding timestep. This results in low-frequency components, especially the mean value, of the training dataset remaining residually present in the final latents during training, to which the model learns to adapt. However, when presented with pure Gaussian noise during inference, the model behaves as if these residual components are still present, resulting in the synthesis of suboptimal imagery [3, 7].

In addressing the aforementioned issue, Guttenberg and CrossLabs [4] first proposed a straightforward solution: introducing a specific offset to the noise derived from sampling, thereby altering its mean value. This technique has been designated as *offset noise*. While this methodology has been employed in some of the more advanced models [18], it is *not* devoid of inherent challenges. Specifically, the incorporation of this offset disrupts the iid distribution characteristics of the noise across individual units. Consequently, although this modification enables the model to produce images with high luminance or profound darkness, it might inadvertently generate signals incongruent with the distribution of the training dataset. A more detailed study [13] suggests a *zero terminal SNR* method that rescaling the model's schedule to ensure the SNR is zero at the terminal timestep can address this issue. Nonetheless, this strategy necessitates the integration of $\mathbf{v}$-prediction models [23] and mandates subsequent fine-tuning across the entire network, regardless of whether the network is based on $\mathbf{v}$-prediction or ϵ-prediction [6]. Besides, fine-tuning these widely-used pre-trained models would render many community models based on earlier releases incompatible, diminishing the overall cost-to-benefit ratio.

To better address this challenge, we revisited the reasons for its emergence: *flaws in the schedule result in a mismatch between the marginal distributions of terminal noise during the training and inference stages*. Concurrently, we found the distinct nature of this terminal timestep: the latents predicted by the model at the terminal timestep continue to be associated with the data distribution.

Based on the above findings, we propose a plug-and-play method, named **O**ne **M**ore **S**tep, that solves this problem without necessitating alterations to the pre-existing trained models, as shown in Fig. 1. This is achieved by training an auxiliary text-conditional network tailored to map pure Gaussian noise to the data-adulterated noise assumed by the pre-trained model, optionally under the guidance of an additional prompt, and is introduced prior to the inception of the iterative sampling process.

OMS can rectify the disparities in marginal distributions encountered during the training and inference phases. Additionally, it can also be leveraged to adjust the generated images through an additional prompt, due to its unique property and position in the sampling sequence. It is worth noting that our method exhibits versatility, being amenable to any variance-preserving [27] diffusion framework, irrespective of the network prediction type, whether ϵ-prediction or $\mathbf{v}$-prediction, and independent of the SDE or ODE solver employed. Experiments demonstrate that SD1.5, SD2.1, LCM [15] and other popular community models can *share the same* OMS module for improved image generation.

2. Preliminaries

2.1. Diffusion Model and its Prediction Types

We consider diffusion models [6, 25] specified in discrete time space and variance-preserving (VP) [27] formulation. Given the training data $\mathbf{x} \in p(\mathbf{x})$, a diffusion model performs the forward process to destroy the data $\mathbf{x}_0$ into noise $\mathbf{x}_T$ according to the pre-defined variance schedule $\{\beta_t\}_{t=1}^T$ according to a perturbation kernel, defined as:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \tag{1}$$

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}\right). \tag{2}$$

The forward process also has a closed-form equation, which allows directly sampling x_t at any timestep t from x_0:

$$q(\mathbf{x}_t|\mathbf{x}_0) := \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{3}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ and $\alpha_t = 1 - \beta_t$. Furthermore, the *signal-to-noise ratio* (SNR) of the latent variable can be defined as:

$$\text{SNR}(t) = \bar{\alpha}_t / (1 - \bar{\alpha}_t). \tag{4}$$

The reverse process denoises a sample $\mathbf{x}_T$ from a standard Gaussian distribution to a data sample $\mathbf{x}_0$ following:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}_t, \tilde{\sigma}_t^2 \mathbf{I}). \tag{5}$$

$$\tilde{\mu}_t := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}\mathbf{x}_t \tag{6}$$

Instead of directly predicting $\tilde{\mu}_t$ using a network θ, predicting the reparameterised ϵ for $\mathbf{x}_0$ leads to a more stable result [6]:

$$\tilde{\mathbf{x}}_0 := (\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t}\epsilon_\theta(\mathbf{x}_t, t))/\sqrt{\bar{\alpha}_t} \tag{7}$$

and the variance of the reverse process $\tilde{\sigma}_t^2$ is set to be $\sigma_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$ while $\mathbf{x}_t \sim \mathcal{N}(0, 1)$. Additionally, predicting velocity [23] is another parameterisation choice for the network to predict:

$$\mathbf{v}_t := \sqrt{\bar{\alpha}_t}\epsilon - \sqrt{1 - \bar{\alpha}_t}\mathbf{x}_0; \tag{8}$$

which can reparameterise $\tilde{\mathbf{x}}_0$ as:

$$\tilde{\mathbf{x}}_0 := \sqrt{\bar{\alpha}_t}\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t}\mathbf{v}_\theta(\mathbf{x}_t, t) \tag{9}$$

2.2. Offset Noise and Zero Terminal SNR

Offset noise [4] is a straightforward method to generate dark or light images more effectively by fine-tuning the model with modified noise. Instead of directly sampling a noise from standard Gaussian Distribution $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, one can sample the initial noise from

$$\epsilon \sim \mathcal{N}(0, \mathbf{I} + 0.1\mathbf{\Sigma}), \tag{10}$$

where $\mathbf{\Sigma}$ is a covariance matrix of all ones, representing fully correlated dimensions. This implies that the noise bias introduced to pixel values across various channels remains consistent. In the initial configuration, the noise attributed to each pixel is independent, devoid of coherence. By adding a common noise across the entire image (or along channels), changes can be coordinated throughout the image, facilitating enhanced regulation of low-frequency elements. However, this is an unprincipled ad hoc adjustment that inadvertently leads to the noise mean of inputs deviating from representing the mean of the actual image.

A different research endeavor proposes a more fundamental approach to mitigate this challenge [13]: rescaling the beta schedule ensures that the low-frequency information within the sampled latent space during training is thoroughly destroyed. To elaborate, current beta schedules are crafted with an intent to minimize the SNR at $\mathbf{x}_T$. However, constraints related to model intricacies and numerical stability preclude this value from reaching zero. Given a beta schedule used in LDM [20]:

$$\beta_t = \left(\sqrt{0.00085}\frac{T - t}{T - 1} + \sqrt{0.012}\frac{t - 1}{T - 1}\right)^2, \tag{11}$$

the terminal SNR at timestep $T = 1000$ is 0.004682 and $\sqrt{\bar{\alpha}_T}$ is 0.068265. To force terminal SNR=0, rescaling can be done to make $\bar{\alpha}_T = 0$ while keeping $\bar{\alpha}_0$ fixed. Subsequently, this rescaled beta schedule can be used to fine-tune the model to avoid the information leakage. Concurrently, to circumvent the numerical instability induced by the prevalent ϵ-prediction at zero terminal SNR, this work mandates the substitution of prediction types across all timesteps with $\mathbf{v}$-prediction. However, such approaches cannot be correctly applied for sampling from pre-trained models that are based on Eq. 11.

3. Methods

3.1. Discrepancy between Training and Sampling

From the beta schedule in Eq. 11, we find the SNR *cannot* reach zero at terminal timestep as $\bar{\alpha}_T$ is not zero. Substituting the value of $\bar{\alpha}_T$ in Eq. 3, we can observe more intuitively that during the training process, the latents sampled by the model at T deviate significantly from expected values:

$$\mathbf{x}_T^{\mathcal{T}} = \sqrt{\bar{\alpha}_T^{\mathcal{T}}}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_T^{\mathcal{T}}}\mathbf{z}, \tag{12}$$

where $\sqrt{\bar{\alpha}_T^{\mathcal{T}}} = 0.068265$ and $\sqrt{1 - \bar{\alpha}_T^{\mathcal{T}}} = 0.997667$.

During the training phase, the data fed into the model is not entirely pure noise at timestep T. It contains minimal yet data-relevant signals. These inadvertently introduced signals contain low-frequency details, such as the overall mean of each channel. The model is subsequently trained to denoise by respecting the mean in the leaked signals. However, in the inference phase, *sampling is executed using standard Gaussian distribution*. Due to such an inconsistency in the distribution between training and inference, when given the zero mean of Gaussian noise, the model unsurprisingly produces samples with the mean value presented at T, resulting in the manifestation of images with median values. Mathematically, the directly sampled variable $\mathbf{x}_T^{\mathcal{S}}$ in the inference stage adheres to the standard Gaussian distribution $\mathcal{N}(0, \mathbf{I})$. However, the marginal distribution of the forward process from image space $\mathcal{X}$ to the latent space $\mathbf{x}_T^{\mathcal{T}}$ during training introduces deviations of the low-frequency information, which is non-standard Gaussian distribution.

This discrepancy is more intuitive in the visualization of high-dimensional Gaussian space by estimating the radius

r [30], which is closely related to the expected distance of a random point from the origin of this space. Theoretically, given a point $\mathbf{x} = (x_1, x_2, \ldots, x_d)$ sampled within the Gaussian domain spanning a d-dimensional space, the squared length or the norm of $\mathbf{x}$ inherently denotes the squared distance from this point to the origin according to:

$$E(x_1^2 + x_2^2 + \cdots + x_d^2) = dE(x_1^2) = d\sigma^2, \tag{13}$$

and the square root of the norm is Gaussian radius r. When this distribution is anchored at the origin with its variance represented by σ, its radius in Gaussian space is determined by:

$$r = \sigma\sqrt{d}, \tag{14}$$

the average squared distance of any point randomly selected from the Gaussian distribution to the origin. Subsequently, we evaluated the radius within the high-dimensional space for both the variables present during the training phase $r^{\mathcal{T}}$ and those during the inference phase $r^{\mathcal{S}}$, considering various beta schedules, the results are demonstrated in Tab. 1. Additionally, drawing from [2, 30], we can observe that the concentration mass of the Gaussian sphere resides above the equator having a radius magnitude of $\mathcal{O}\left(\frac{r}{\sqrt{d}}\right)$, also within an annulus of constant width and radius n. Therefore, we can roughly visualize the distribution of terminal variables during both the training and inference processes in Fig. 2. It can be observed that a discernible offset emerges between the terminal distribution $\mathbf{x}_T^{\mathcal{T}}$ and $\mathbf{x}_T^{\mathcal{S}}$ and $r^{\mathcal{S}} > r^{\mathcal{T}}$. This intuitively displays the discrepancy between training and inference, which is our primary objective to mitigate. Additional theoretical validations are relegated to the Appendix for reference.

Schedule	SNR(T)	$r^{\mathcal{T}}$	$r^{\mathcal{S}}$	Δr
cosine	2.428e-09	443.404205	443.404235	3.0518e-05
linear	4.036e-05	443.393676	443.399688	6.0119e-03
LDM Pixels	4.682e-03	442.713593	443.402527	6.8893e-01
LDM Latents†	4.682e-03	127.962364	127.996811	3.4447e-02

† LDMs were conducted both in the *unit variance* latent space (4*64*64) and pixel space (3*256*256) while others are conducted in pixel space.

Table 1. Estimation of the Gaussian radius during the sampling and inference phases under different beta schedules. Here, we randomly sampled 20,000 points to calculate the radius.

3.2. Prediction at Terminal Timestep

According to Eq. 5 & 7, we can obtain the sampling process under the text-conditional DDPM pipeline with ϵ-prediction at timestep T:

$$\mathbf{x}_{T-1} = \frac{1}{\sqrt{\alpha_T}}\left(\mathbf{x}_T - \frac{1-\alpha_T}{\sqrt{1-\bar{\alpha}_T}}\epsilon_\theta\right) + \sigma_T\mathbf{z}, \tag{15}$$

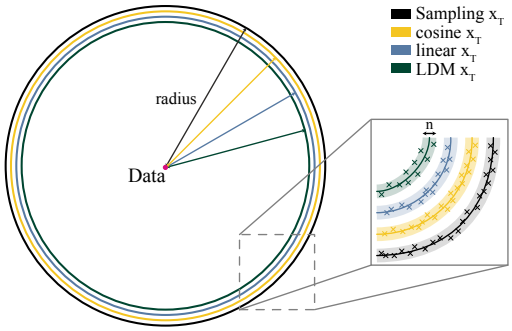


Figure 2. The geometric illustration of concentration mass in the equatorial cross-section of high-dimensional Gaussians, where its mass concentrates in a very small annular band around the radius. Different colors represent the results sampled based on different schedules. It can be seen that as the SNR increases, the distribution tends to be more data-centric, thus the radius of the distribution is gradually decreasing.

where $\mathbf{z}, \mathbf{x}_T \sim \mathcal{N}(0, \mathrm{I})$. In this particular scenario, it is obvious that the ideal SNR(T) = 0 setting (with $\alpha_T = 0$) will lead to numerical issues, and any predictions made by the network at time T with an SNR(T) = 0 are arduous and lack meaningful interpretation. This also elucidates the necessity for the linear schedule to define its start and end values [6] and for the cosine schedule to incorporate an offset s [17].

Utilizing SNR-independent $\mathbf{v}$-prediction can address this issue. By substituting Eq. 9 into Eq. 5, we can derive:

$$\mathbf{x}_{T-1} = \sqrt{\alpha_T}\mathbf{x}_T - \frac{\sqrt{\bar{\alpha}_{T-1}}(1-\alpha_T)}{\sqrt{1-\bar{\alpha}_T}}\mathbf{v}_\theta + \sigma_T\mathbf{z}, \tag{16}$$

which the assumption of SNR(T) = 0 can be satisfied: when SNR(T) = 0, the reverse process of calculating $\mathbf{x}_{T-1}$ depends only on the prediction of $\mathbf{v}_\theta(\mathbf{x}_T, T)$,

$$\mathbf{x}_{T-1} = -\sqrt{\bar{\alpha}_{T-1}}\mathbf{v}_\theta + \sigma_T\mathbf{z}, \tag{17}$$

which can essentially be interpreted as predicting the direction of $\mathbf{x}_0$ according to Eq. 8:

$$\mathbf{x}_{T-1} = \sqrt{\bar{\alpha}_{T-1}}\mathbf{x}_0 + \sigma_T\mathbf{z}. \tag{18}$$

This is also consistent with the conclusions of angular parameterisation[1] [23]. To conclude, under the ideal condition of SNR = 0, the model is essentially forecasting the L2 mean of the data, hence the objective of the $\mathbf{v}$-prediction at this stage aligns closely with that of the direct $\mathbf{x}_0$-prediction. Furthermore, this prediction by the network at this step is independent of the pipeline schedule, implying that the prediction remain consistent irrespective of the variations in noise input.

[1]Details about $\mathbf{v}$-prediction and angular parametersation can be found in the Appendix. C

3.3. Adding One More Step

Holding the assumption that $\mathbf{x}_T$ belongs to a standard Gaussian distribution, the model actually has no parameters to be trained with pre-defined beta schedule, so the objective L_T should be the constant:

$$L_T = D_{\mathrm{KL}}\left(q(\mathbf{x}_T|\mathbf{x}_0)\|p(\mathbf{x}_T)\right). \tag{19}$$

In the present architecture, the model conditioned on $\mathbf{x}_T$ actually does not participate in the training. However, existing models have been trained to predict based on $\mathbf{x}_T^{\mathcal{T}}$, which indeed carries some data information.

Drawing upon prior discussions, we know that the model's prediction conditioned on $\mathbf{x}_T^{\mathcal{S}}$ should be the average of the data, which is also independent of the beta schedule. This understanding brings a new perspective to the problem: retaining the whole pipeline of the current model, encompassing both its parameters and the beta schedule. In contrast, we can reverse $\mathbf{x}_T^{\mathcal{S}}$ to $\mathbf{x}_T^{\mathcal{T}}$ by introducing **O**ne **M**ore **S**tep (**OMS**). In this step, we first train a network $\psi(\mathbf{x}_T^{\mathcal{S}}, \mathcal{C})$ to perform $\mathbf{v}$-prediction conditioned on $\mathbf{x}_T^{\mathcal{S}} \sim \mathcal{N}(0, \mathbf{I})$ with L2 loss $\|\mathbf{v}_T^{\mathcal{S}} - \tilde{\mathbf{v}}_T^{\mathcal{S}}\|_2^2$, where $\mathbf{v}_T^{\mathcal{S}} = -\mathbf{x}_0$ and $\tilde{\mathbf{v}}_T^{\mathcal{S}}$ is the prediction from the model. Next, we reconstruct $\tilde{\mathbf{x}}_T^{\mathcal{T}}$ based on the output of ψ with different solvers. In addition to the SDE Solver delineated in Eq. 17, we can also leverage prevalent ODE Solvers, *e.g.*, DDIM [26]:

$$\tilde{\mathbf{x}}_T^{\mathcal{T}} = \sqrt{\bar{\alpha}_T^{\mathcal{T}}}\tilde{\mathbf{x}}_0 + \sqrt{1 - \bar{\alpha}_T^{\mathcal{T}} - \sigma_T^2}\mathbf{x}_T^{\mathcal{S}} + \sigma_T \mathbf{z}, \tag{20}$$

where $\tilde{\mathbf{x}}_0$ is obtained based on $\psi(\mathbf{x}_T^{\mathcal{S}}, \mathcal{C})$. Subsequently, $\tilde{\mathbf{x}}_T^{\mathcal{T}}$ can be utilized as the initial noise and incorporated into various pre-trained models. From a geometrical viewpoint, we employ a model conditioned on $\mathbf{x}_T^{\mathcal{S}}$ to predict $\tilde{\mathbf{x}}_T^{\mathcal{T}}$ that aligns more closely with $\mathcal{N}\left(\sqrt{\bar{\alpha}_T^{\mathcal{T}}}\mathbf{x}_0, (1 - \bar{\alpha}_T^{\mathcal{T}})\mathbf{I}\right)$, which has a smaller radius and inherits to the training phase of the pre-trained model at timestep T. The whole pipeline and geometric explanation is demonstrated in Figs. 3 & 4, and the detailed algorithm and derivation can be referred to Alg. 1 in Appendix D.1.

Notably, the prompt $\mathcal{C}_\psi$ in OMS phase $\psi(\cdot)$ can be different from the conditional information $\mathcal{C}_\theta$ for the pre-trained diffusion model $\theta(\cdot)$. Modifying the prompt in OMS phase allows for additional manipulation of low-frequency aspects of the generated image, such as color and luminance. Besides, OMS module also support classifier free guidance [5] to strength the text condition:

$$\psi_{\mathrm{cfg}}(\mathbf{x}_T^{\mathcal{S}}, \mathcal{C}_\psi, \emptyset, \omega_\psi) = \psi(\mathbf{x}_T^{\mathcal{S}}, \emptyset) + \omega_\psi\left(\psi(\mathbf{x}_T^{\mathcal{S}}, \mathcal{C}_\psi) - \psi(\mathbf{x}_T^{\mathcal{S}}, \emptyset)\right), \tag{21}$$

where ω_ψ is the CFG weights for OMS. Experimental results for inconsistent prompt and OMS CFG can be found in Sec. 4.3.

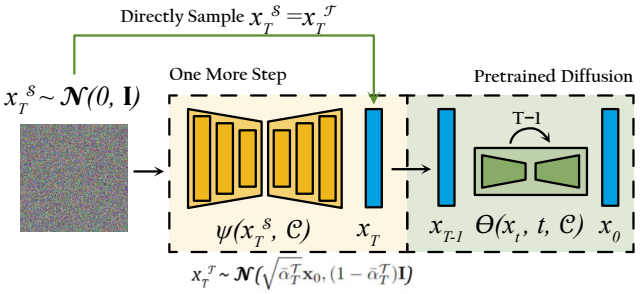


Figure 3. The pipeline of ***One More Step***. The section highlighted in **yellow** signifies our introduced OMS module, with ψ being the only trainable component. The segments in **blue** represents latent vectors, and **green** represents the pre-trained model used only for the inference.

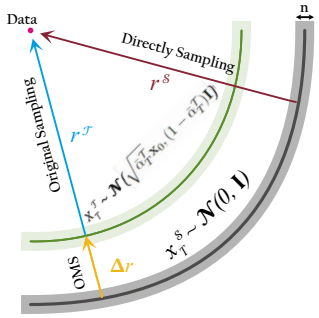


Figure 4. Geometric explanation of ***One More Step***. While directly sampling method requires sampling from a Gaussian distribution with a radius of $r^{\mathcal{T}}$, yet it samples from the standard Gaussian with $r^{\mathcal{S}}$ in practice. OMS bridges the gap Δr between $r^{\mathcal{S}}$ and the required $r^{\mathcal{T}}$ through an additional inference step. Here n is the width of the narrow band where the distribution mass is concentrated.

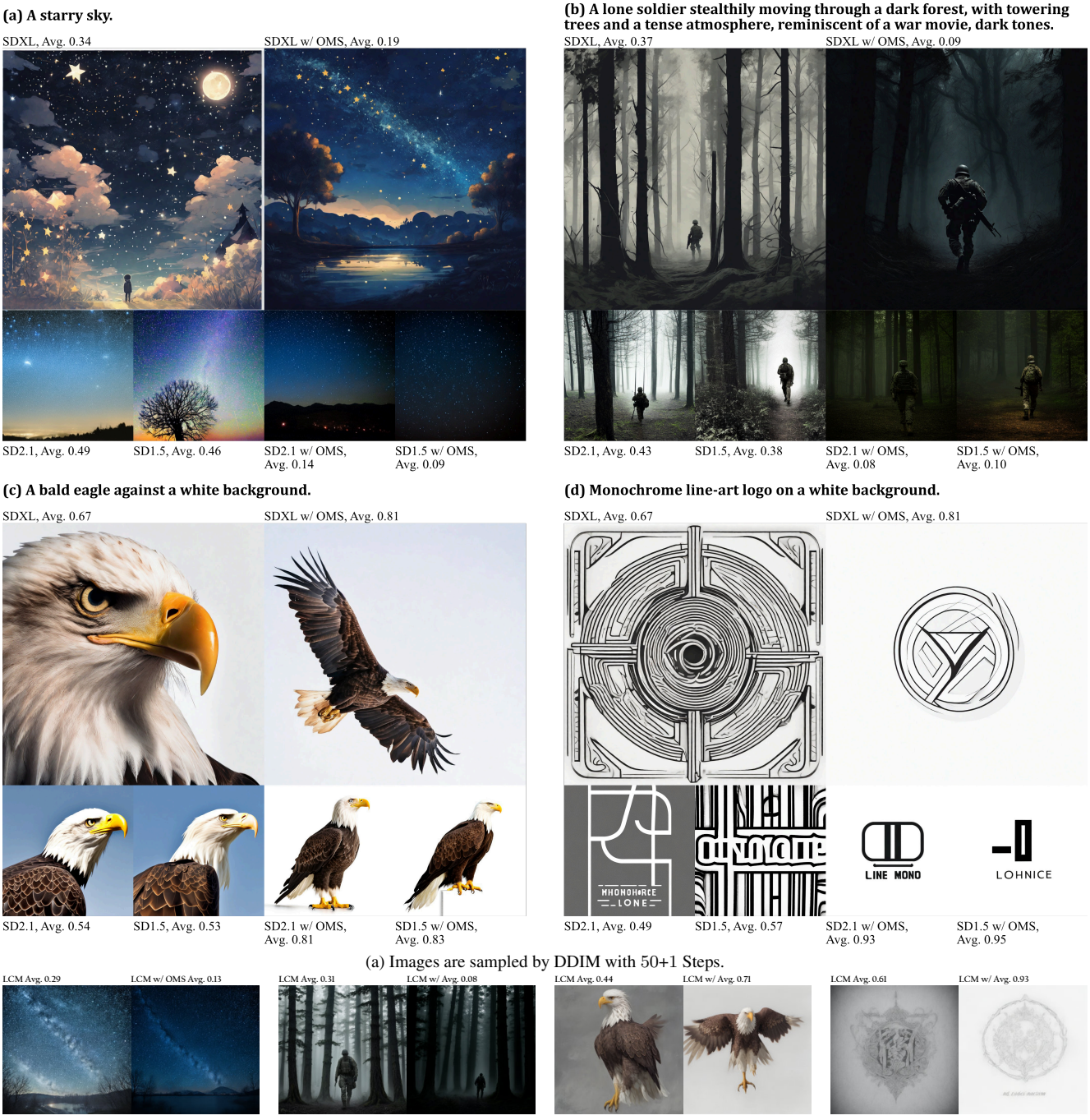
It is worth noting that OMS can be adapted to any pre-trained model within the same space. Simply put, our OMS module trained in the same VAE latent domain can adapt to any other model that has been trained within the same latent space and data distribution. Details of the OMS and its versatility can be found in Appendix D.2 & D.4.

4. Experiments

This section begins with an evaluation of the enhancements provided by the proposed OMS module to pre-trained generative models, examining both qualitative and quantitative aspects, and its adaptability to a variety of diffusion models. Subsequently, we conducted ablation studies on pivotal designs and dive into several interesting occurrences.

4.1. Implementation Details

We trained our OMS module on LAION 2B dataset [24]. OMS module architecture follows the widely used UNet [6, 21] in diffusion, and we evaluated different configurations,



(a) Images are sampled by DDIM with 50+1 Steps.

(b) Images are sampled by LCM with 4+1 Steps and the same prompts sets.

Figure 5. Qualitative comparison. For each image pair, the left shows results from original pre-trained diffusion models, whereas the right demonstrates the output from these same models enhanced with the OMS under identical prompts. It is worth noting that SD1.5, SD2.1 [20] and LCM [15] in this experiment *share the same OMS module, rather than training an exclusive module for each one.* .

e.g., number of layers. By default we employ OpenCLIP ViT-H to encode text for the OMS module and trained the model for 2,000 steps. For detailed implementation information, please refer to the Appendix.

4.2. Performance

Qualitative Figs. 1 and 5 illustrate that our approach is capable of producing images across a large spectrum of brightness levels. Among these, SD1.5, SD2.1 and

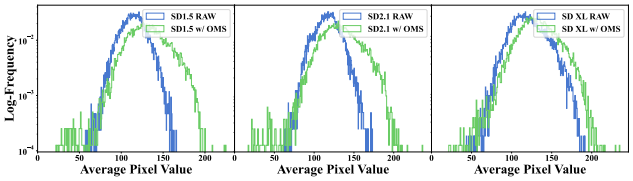


Figure 6. Log-frequency histogram of image mean values.

LCM [15] use the *same OMS module*, whereas SDXL employs a separately trained OMS module[2]. As shown in the Fig. 5 left, existing models invariably yield samples of medium brightness and are *not* able to generate accurate images when provided with explicit prompts. In contrast, our model generates a distribution of images that is more broadly covered based on the prompts. In addition to further qualifying the result, we also show some integration of the widely popular customized LoRA [8] and base models in the community with our module in Appendix E, which also ascertains the versatility of OMS.

Quantitative For the quantitative evaluation, we randomly selected 10k captions from MS COCO [14] for zero-shot generation of images. We used Fréchet Inception Distance (FID), CLIP Score [19], Image Reward [28], and PickScore [11] to assess the quality, text-image alignment, and human preference of generated images. Tab. 2 presents a comparison of these metrics across various models, either with or without the integration of the OMS module. It is worth noting that Kirstain et al. [11] demonstrated that the FID score for COCO zero-shot image generation has a *negative correlation* with visual aesthetics, thus the FID metric is not congruent with the goals of our study. Instead, we have further computed the Precision-Recall (PR) [12] and Density-Coverage (DC) [16] between the ground truth images and those generated, as detailed in the Tab. 2. Additionally, we calculate the mean of images and the Wasserstein distance [22], and visualize the log-frequency distribution in Fig. 6. It is evident that our proposed OMS module promotes a more broadly covered distribution.

4.3. Ablation

Module Scale Initially, we conducted some research on the impact of model size. The aim is to explore whether variations in the parameter count of the OMS model would influence the enhancements in image quality. We experimented with OMS networks of three different sizes and discovered that the amelioration of image quality is *not* sensitive to the number of OMS parameters. From Appendix, we found that even with only 3.7M parameters, the model was still able to successfully improve the distribution of generated images. This result offers us an insight: it is conceivable that during the entire denoising process, certain timesteps encounter relatively trivial challenges, hence the model scale of specific timestep might be minimal and using a Mixture of Experts strategy [1] but with different scale models at diverse timesteps may effectively reduce the time required for inference.

[2]The VAE latent domain of the SDXL model differs considerably from those of SD1.5, SD2.1 and LCM. For more detailed information, please refer to the Appendix D.4.

(a) Modifying the prompts in the OMS module can adjust the brightness in the generated images.

(b) Modifying the prompts in the OMS module can change the object color in the generated images.

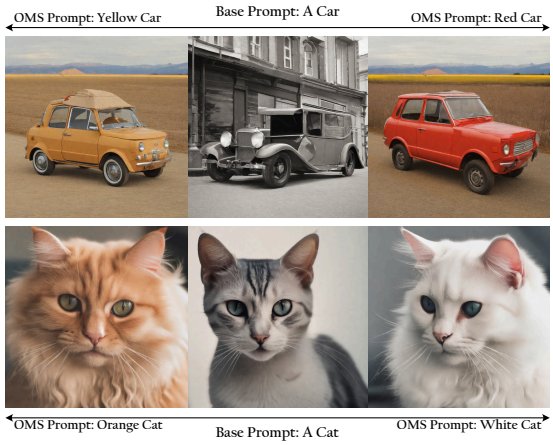


Figure 7. Altering the prompts in the OMS module, while keeping the text prompts in the diffusion backbone model constant, can notably affect the characteristics of the images generated.

Text Encoder Another critical component in OMS is the text encoder. Given that the OMS model's predictions can be interpreted as the mean of the data informed by the prompt, it stands to reason that a more potent text encoder would enhance the conditional information fed into the OMS module. However, experiments show that the improvement brought by different encoders is also limited. We believe that the main reason is that OMS is only effective for

Model		FID ↓	CLIP Score↑	ImageReward↑	PickScore↑	Precision↑	Recall↑	Density↑	Coverage↑	Wasserstein↓
SD1.5	RAW	**12.52**	0.2641	0.1991	21.49	0.60	**0.55**	0.56	0.54	22.47
	OMS	14.74	**0.2645**	**0.2289**	**21.55**	**0.64**	0.46	**0.64**	**0.57**	**7.84**
SD2.1	RAW	**14.10**	0.2624	0.4501	21.80	0.58	**0.55**	0.52	0.50	21.63
	OMS	15.72	**0.2628**	**0.4565**	**21.82**	**0.61**	0.48	**0.58**	**0.54**	**7.70**
SD XL	RAW	**13.14**	0.2669	0.8246	22.51	0.64	**0.52**	0.67	0.63	11.08
	OMS	13.29	**0.2679**	**0.8730**	**22.52**	**0.65**	0.49	**0.70**	**0.64**	**7.25**

Table 2. Quantitative evaluation. All models use DDIM sampler with 50 steps, guidance weight $\omega_\theta = 7.5$ and negative prompts are $\emptyset$. For OMS module, there is no OMS CFG $\omega_\psi = 1$ and no inconsistent prompt $\mathcal{C}_\psi = \mathcal{C}_\theta$. Better results are highlighted in **bold**.

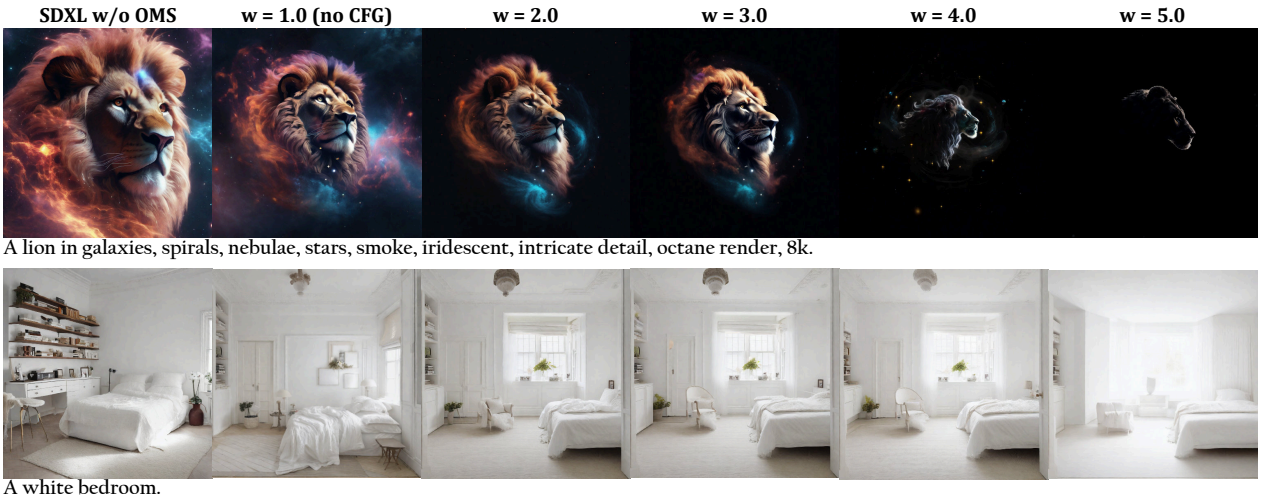


Figure 8. Images under the same prompt but with different OMS CFG weights applied in OMS module. Notably, CFG weight of the pre-trained diffusion model remains 7.5.

low-frequency information in the generation process, and these components are unlikely to affect the explicit representation of the image. The diverse results can be found in Tab. 4 in Appendix D.3.

Modified Prompts In addition to providing coherent prompts, we also conducted experiments to examine the impact of the low-frequency information during the OMS step with different prompts, mathematically $\mathcal{C}_\psi \neq \mathcal{C}_\theta$. We discovered that the brightness level of the generated images can be easily controlled with terms like $\mathcal{C}_\psi$ is "dark" or "light" in the OMS phase, as can be seen from Fig. 7a. Additionally, our observations indicate that the modified prompts used in the OMS are capable of influencing other semantic aspects of the generated content, including color variations as shown in Fig. 7b.

Classifier-free guidance Classifier-free guidance (CFG) is well-established for enhancing the quality of generated content and is a common practice [5]. CFG still can play a key component in OMS, effectively influencing the low-frequency characteristics of the image in response to the given prompts. Due to the unique nature of our OMS target for generation, the average value under $\emptyset$ is close to that of conditioned ones $\mathcal{C}_\psi$. As a result, even minor applications of CFG can lead to considerable changes. Our experiments show that a CFG weight $\omega_\psi = 2$ can create distinctly visible alterations. In Fig. 8, we can observe the performance of generated images under different CFG weights for OMS module. It worth noting that CFG weights of OMS and the pre-trained model are imposed independently.

5. Conclusion

In summary, our observations indicate a discrepancy in the terminal noise between the training and sampling stages of diffusion models due to the schedules, resulting in a distribution of generated images that is centered around the mean. To address this issue, we introduced ***One More Step***, which adjusts for the training and inference distribution discrepancy by integrating an additional module while preserving the original parameters. Furthermore, we discovered that the initial stages of the denoising process with low SNR largely determine the low-frequency traits of the images, particularly the distribution of brightness, and this phase does not demand an extensive parameter set for accurate model fitting.

References

[1] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. eDiff-I: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 7

[2] Avrim Blum, John Hopcroft, and Ravindran Kannan. *Foundations of data science*. Cambridge University Press, 2020. 4

[3] Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023. 2

[4] Nicholas Guttenberg and CrossLabs. Diffusion with offset noise. Online Webpage, 2023. 2, 3

[5] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 5, 8

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2, 3, 4, 5

[7] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. *arXiv preprint arXiv:2301.11093*, 2023. 2

[8] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2, 7

[9] Minghui Hu, Jianbin Zheng, Daqing Liu, Chuanxia Zheng, Chaoyue Wang, Dacheng Tao, and Tat-Jen Cham. Cocktail: Mixing multi-modality controls for text-conditional image generation. *arXiv preprint arXiv:2306.00964*, 2023. 2

[10] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022. 2

[11] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *arXiv preprint arXiv:2305.01569*, 2023. 7

[12] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32, 2019. 7

[13] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. In *WACV*, pages 5404–5411, 2024. 2, 3

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 7

[15] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023. 2, 6, 7

[16] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *International Conference on Machine Learning*, pages 7176–7185. PMLR, 2020. 7

[17] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 2, 4

[18] 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. *arXiv preprint arXiv:2307.01952*, 2023. 2

[19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 7

[20] 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*, pages 10684–10695, 2022. 2, 3, 6

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 5

[22] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40:99–121, 2000. 7

[23] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 2, 3, 4

[24] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 5

[25] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 2

[26] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 5

[27] 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. 2

[28] 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. *arXiv preprint arXiv:2304.05977*, 2023. 7

[29] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In

Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 3836–3847, 2023. 2

[30] Ye Zhu, Yu Wu, Zhiwei Deng, Olga Russakovsky, and Yan Yan. Boundary guided mixing trajectory for semantic control with diffusion models. *arXiv preprint arXiv:2302.08357*, 2023. 4