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

Supplementary Material

A. Discussion on Related Works

A.1. Off-set Noise

Off-set Noise [4] only embarks on an investigation into low-frequency (LF) information. The author actively disrupts these LF elements and fine-tunes the models. It will lead deviating from the mean of the actual image.

A.2. Rescale Schedule

Rescale Schedule [5] only focuses on the discrepancy between the distribution of training data and the sampling. There is no clear connection between the $N(0,1)$ distribution and the x_0 manifold during the diffusion process. Thus this method needs finetuning the whole model and force the prediction type as v-prediction.

A.3. Signal-leak

Signal-leak [2] mentions that the signal leaked during training is essentially LF components, as well as the cause of distribution mismatch. Therefore, the author statistically analyze the LF components of a limited scale dataset(˜300 images) to obtain the terminal distribution for training. This method overlooks the intrinsic attributes of diffusion models, focusing solely on LF components. This results in a constraint: it is only effective when the terminal distribution can be statistically estimated on limited size dataset, e.g. specific style, which is impractical for large models. Thus it lacks versatility as different training data have distinct terminal distributions, requires calculations for each model. And it is unviable with new acceleration tools, such as consistency models.

In contrast, we not only found LF information is the main cause of the issue but also approach the problem from the perspective of diffusion prediction at time x_T. We can obtain the average of the true distribution using a text-conditional module. This makes our model versatile: regardless of any LDM structure models or any LoRA, it can be solved with a single OMS module (only 3.1M parameters).

B. High Dimensional Gaussian

In our section, we delve into the geometric and probabilistic features of high-dimensional Gaussian distributions, which are not as evident in their low-dimensional counterparts. These characteristics are pivotal for the analysis of latent spaces within denoising models, given that each intermediate latent space follows a Gaussian distribution during denoising. Our statement is anchored on the seminal work by [1, 10, 11]. These works establish a connection between the high-dimensional Gaussian distribution and the latent variables inherent in the diffusion model.

Property B.1 **([10])** *For a unit-radius sphere in high dimensions, as the dimension* d *increases, the volume of the sphere goes to 0, and the maximum possible distance between two points stays at 2.*

Lemma B.2 **([10])** *The surface area* $A(d)$ *and the volume* $V(d)$ *of a unit-radius sphere in* d*-dimensions can be obtained by:*

$$A(d) = \frac{2\pi^{d/2}}{\Gamma(d/2)}, V(d) = \frac{\pi^{d/2}}{\frac{d}{2}\Gamma(d/2)}, \tag{1}$$

where $\Gamma(x)$ represents an extension of the factorial function to accommodate non-integer values of x, the aforementioned Property B.1 and Lemma B.2 constitute universal geometric characteristics pertinent to spheres in higher-dimensional spaces. These principles are not only inherently relevant to the geometry of such spheres but also have significant implications for the study of high-dimensional Gaussians, particularly within the framework of diffusion models during denoising process.

Property B.3 **([10])** *The volume of a high-dimensional sphere is essentially all contained in a thin slice at the equator and is simultaneously contained in a narrow annulus at the surface, with essentially no interior volume. Similarly, the surface area is essentially all at the equator.*

The Property B.3 implies that samples from $\mathbf{x}_T^S$ are falling into a narrow annulus.

Lemma B.4 **([10])** *For any* $c > 0$*, the fraction of the volume of the hemisphere above the plane* $x_1 = \frac{c}{\sqrt{d-1}}$ *is less than* $\frac{2}{c}e^{-\frac{c^2}{2}}$.

Lemma B.5 **([11])** *For a* d*-dimensional spherical Gaussian of variance 1, all but* $\frac{4}{c^2}e^{-c^2/4}$ *fraction of its mass is within the annulus* $\sqrt{d-1} - c \le r \le \sqrt{d-1} + c$ *for any* $c > 0$.

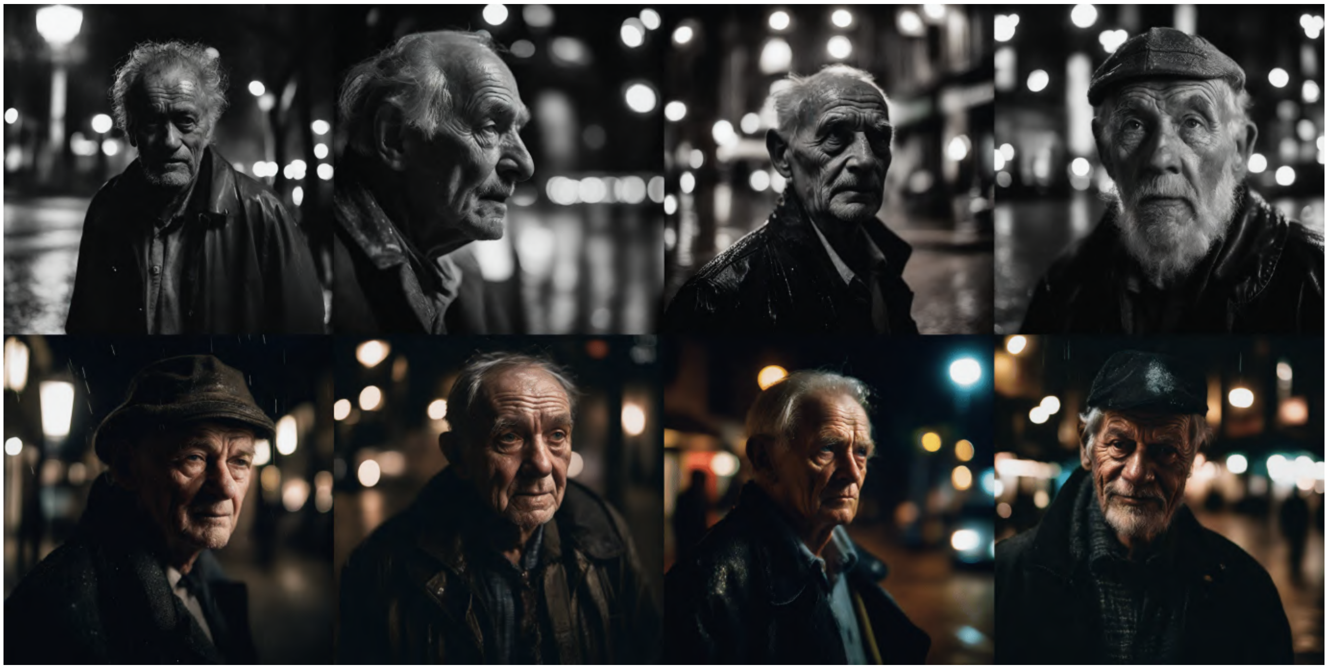


Figure 1. The same set of configurations (SDXL w/ LCM-LoRA with $4(+1)$ Steps) as Fig. 5 but with different random seeds. SDXL with LCM-LoRA leans towards black-and-white images, but OMS produces more colorful images. It is worth noting the mean value of all SDXL with LCM-LoRA results is 0.24 while the average value of OMS results is **0.17**. We hypothesize the tendency of SDXL to produce black-and-white images is a direct result of flaws in its scheduler for training.

Lemmas B.4 & B.5 imply the volume range of the concentration mass above the equator is in the order of $O(\frac{r}{\sqrt{d}})$, also within an annulus of constant width and radius $\sqrt{d-1}$. Figs.2 & 4 in main paper illustrates the geometric properties of the ideal sampling space $\mathbf{x}_T^{\mathcal{S}}$ compared to the practical sampling spaces $\mathbf{x}_T^{\mathcal{T}}$ derived from various schedules, which should share an identical radius ideally.

Property B.6 ([11]) *The maximum likelihood spherical Gaussian for a set of samples is the one over center equal to the sample mean and standard deviation equal to the standard deviation of the sample.*

The above Property B.6 provides the theoretical foundation whereby the mean of squared distances serves as a robust statistical measure for approximating the radius of high-dimensional Gaussian distributions.

C. Expression of DDIM in angular parameterization

The following covers derivation that was originally presented in [7], with some corrections. We can simplify the DDIM update rule by expressing it in terms of $\phi_t = \arctan(\sigma_t/\alpha_t)$, rather than in terms of time t or log-SNR λ_t, as we show here.

Given our definition of ϕ, and assuming a variance preserving diffusion process, we have $\alpha_\phi = \cos(\phi)$, $\sigma_\phi = \sin(\phi)$, and hence $\mathbf{z}_\phi = \cos(\phi)\mathbf{x} + \sin(\phi)\epsilon$. We can now define the velocity of $\mathbf{z}_\phi$ as

$$\mathbf{v}_\phi \equiv \frac{d\mathbf{z}_\phi}{d\phi} = \frac{d\cos(\phi)}{d\phi}\mathbf{x} + \frac{d\sin(\phi)}{d\phi}\epsilon = \cos(\phi)\epsilon - \sin(\phi)\mathbf{x}. \tag{2}$$

Rearranging $\epsilon, \mathbf{x}, \mathbf{v}$, we then get:

$$\begin{aligned} \sin(\phi)\mathbf{x} &= \cos(\phi)\epsilon - \mathbf{v}_\phi \\ &= \frac{\cos(\phi)}{\sin(\phi)}(\mathbf{z} - \cos(\phi)\mathbf{x}) - \mathbf{v}_\phi \end{aligned} \tag{3}$$

$$\sin^2(\phi)\mathbf{x} = \cos(\phi)\mathbf{z} - \cos^2(\phi)\mathbf{x} - \sin(\phi)\mathbf{v}_\phi \tag{4}$$

$$(\sin^2(\phi) + \cos^2(\phi))\mathbf{x} = \mathbf{x} = \cos(\phi)\mathbf{z} - \sin(\phi)\mathbf{v}_\phi, \tag{5}$$

and similarly we get $\epsilon = \sin(\phi)\mathbf{z}_\phi + \cos(\phi)\mathbf{v}_\phi$.

Furthermore, we define the predicted velocity as:

$$\hat{\mathbf{v}}_\theta(\mathbf{z}_\phi) \equiv \cos(\phi)\hat{\epsilon}_\theta(\mathbf{z}_\phi) - \sin(\phi)\hat{\mathbf{x}}_\theta(\mathbf{z}_\phi), \tag{6}$$

where $\hat{\epsilon}_\theta(\mathbf{z}_\phi) = (\mathbf{z}_\phi - \cos(\phi)\hat{\mathbf{x}}_\theta(\mathbf{z}_\phi))/\sin(\phi)$.

Rewriting the DDIM update rule in the introduced terms then gives:

$$\begin{aligned}\mathbf{z}_{\phi_s} &= \cos(\phi_s)\hat{\mathbf{x}}_\theta(\mathbf{z}_{\phi_t}) + \sin(\phi_s)\hat{\epsilon}_\theta(\mathbf{z}_{\phi_t})\\ &= \cos(\phi_s)(\cos(\phi_t)\mathbf{z}_{\phi_t} - \sin(\phi_t)\hat{\mathbf{v}}_\theta(\mathbf{z}_{\phi_t}))+\\ &\quad \sin(\phi_s)(\sin(\phi_t)\mathbf{z}_{\phi_t} + \cos(\phi_t)\hat{\mathbf{v}}_\theta(\mathbf{z}_{\phi_t}))\\ &= [\cos(\phi_s)\cos(\phi_t) + \sin(\phi_s)\sin(\phi_t)]\mathbf{z}_{\phi_t} +\\ &\quad [\sin(\phi_s)\cos(\phi_t) - \cos(\phi_s)\sin(\phi_t)]\hat{\mathbf{v}}_\theta(\mathbf{z}_{\phi_t}).\end{aligned} \tag{7}$$

Finally, we use the trigonometric identities

$$\begin{aligned}\cos(\phi_s)\cos(\phi_t) + \sin(\phi_s)\sin(\phi_t) &= \cos(\phi_s - \phi_t)\\ \sin(\phi_s)\cos(\phi_t) - \cos(\phi_s)\sin(\phi_t) &= \sin(\phi_s - \phi_t),\end{aligned} \tag{8}$$

to find that[1]

$$\mathbf{z}_{\phi_s} = \cos(\phi_s - \phi_t)\mathbf{z}_{\phi_t} + \sin(\phi_s - \phi_t)\hat{\mathbf{v}}_\theta(\mathbf{z}_{\phi_t}). \tag{9}$$

or equivalently

$$\mathbf{z}_{\phi_t - \delta} = \cos(\delta)\mathbf{z}_{\phi_t} - \sin(\delta)\hat{\mathbf{v}}_\theta(\mathbf{z}_{\phi_t}). \tag{10}$$

Viewed from this perspective, DDIM thus evolves $\mathbf{z}_{\phi_s}$ by moving it on a circle in the $(\mathbf{z}_{\phi_t}, \hat{\mathbf{v}}_{\phi_t})$ basis, along the $-\hat{\mathbf{v}}_{\phi_t}$ direction. When SNR is set to zero, the v-prediction effectively reduces to the $\mathbf{x}_0$-prediction. The relationship between $\mathbf{z}_{\phi_t}, \mathbf{v}_t, \alpha_t, \sigma_t, \mathbf{x}, \epsilon$ is visualized in Fig. 2.

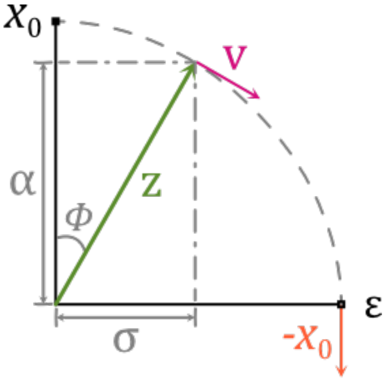


Figure 2. Visualization of reparameterizing the diffusion process in terms of ϕ and $\mathbf{v}_\phi$. We highlight the scenario where SNR is equal to zero in orange.

D. More Empirical Details

D.1. Detailed Algorithm

Due to space limitations, we omitted some implementation details in the main body, but we provided a detailed version of the OMS based on DDIM sampling in Alg. 1. This example implementation utilizes $\mathbf{v}$-prediction for the OMS and ϵ-prediction for the pre-trained model.

Algorithm 1: DDIM Sampling with OMS

```
Require: Pre-trained Diffusion Pipeline with a
         model $\theta$ to perform $\epsilon$-prediction.
Require: One More Step module $\psi(\cdot)$
Input: OMS Text Prompt $\mathcal{C}_\psi$, OMS CFG weight $\omega_\psi$
Input: Text Prompt $\mathcal{C}_\theta$, Guidance weight $\omega_\theta$, Eta $\sigma$
# Introduce One More Step
$\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ ;
$\mathbf{x}_T^{\mathcal{S}} \sim \mathcal{N}(0, \mathbf{I})$;
# Classifier Free Guidance at One More Step Phase
if $\omega_\psi > 1$ then
    $\tilde{\mathbf{x}}_0^{\mathcal{S}} = -\psi_{\text{cfg}}(\mathbf{x}_T^{\mathcal{S}}, \mathcal{C}_\psi, \emptyset, \omega_\psi)$ ;
else
    $\tilde{\mathbf{x}}_0^{\mathcal{S}} = -\psi(\mathbf{x}_T^{\mathcal{S}}, \mathcal{C}_\psi)$ ;
end
$\tilde{\mathbf{x}}_T^{\mathcal{T}} = \sqrt{\bar{\alpha}_T^{\mathcal{T}}}\tilde{\mathbf{x}}_0^{\mathcal{S}} + \sqrt{1 - \bar{\alpha}_T^{\mathcal{T}} - \sigma^2}\mathbf{x}_T^{\mathcal{S}} + \sigma\mathbf{z}$ ;
# Sampling from Pre-trained Diffusion Model
for $t = T, \ldots, 1$ do
    $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = 0$;
    if $t$=$T$ then
        if $\omega_\theta > 1$ then
            $\tilde{\epsilon}_T = \theta_{\text{cfg}}(\tilde{\mathbf{x}}_T^{\mathcal{T}}, \mathcal{C}_\theta, \emptyset, \omega_\theta)$ ;
        else
            $\tilde{\epsilon}_T = \theta(\tilde{\mathbf{x}}_t^{\mathcal{T}}, \mathcal{C}_\theta)$ ;
        end
        $\tilde{\mathbf{x}}_{T-1} = \sqrt{\bar{\alpha}_{T-1}}\left(\frac{\tilde{\mathbf{x}}_T^{\mathcal{T}} - \sqrt{1 - \bar{\alpha}_T^{\mathcal{T}}}\tilde{\epsilon}_T}{\sqrt{\bar{\alpha}_T^{\mathcal{T}}}}\right) +$
          $\sqrt{1 - \bar{\alpha}_{T-1} - \sigma^2}\tilde{\epsilon}_T + \sigma\mathbf{z}$ ;
    else
        if $\omega_\theta > 1$ then
            $\tilde{\epsilon}_t = \theta_{\text{cfg}}(\tilde{\mathbf{x}}_t, \mathcal{C}_\theta, \emptyset, \omega_\theta)$ ;
        else
            $\tilde{\epsilon}_t = \theta(\mathbf{x}_t^{\mathcal{T}}, \mathcal{C}_\theta)$ ;
        end
        $\tilde{\mathbf{x}}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\frac{\tilde{\mathbf{x}}_t - \sqrt{1 - \bar{\alpha}_t}\tilde{\epsilon}_t}{\sqrt{\bar{\alpha}_t}}\right) +$
          $\sqrt{1 - \bar{\alpha}_{t-1} - \sigma^2}\tilde{\epsilon}_t + \sigma\mathbf{z}$
    end
end
return $\tilde{\mathbf{x}}_0$
```

The derivation related to prediction of $\tilde{\mathbf{x}}_T^{\mathcal{T}}$ in Eq.20 can be obtained from Eq.12 in [9]. Given $\mathbf{x}_t$, one can generate $\mathbf{x}_0$:

$$\tilde{\mathbf{x}}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\frac{\tilde{\mathbf{x}}_t - \sqrt{\bar{\alpha}_t}\tilde{\epsilon}_t}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2}\tilde{\epsilon}_t + \sigma_t\mathbf{z}, \tag{11}$$

where $\tilde{\mathbf{x}}_0^t$ is parameterised by $\frac{\tilde{\mathbf{x}}_t - \sqrt{\bar{\alpha}_t}\tilde{\epsilon}_t}{\sqrt{\bar{\alpha}_t}}$. In OMS phase, $\bar{\alpha}_T^{\mathcal{S}} = 0$ and $\bar{\alpha}_{T-1}^{\mathcal{S}} = \bar{\alpha}_T^{\mathcal{T}}$. According to Eq.9, the OMS

[1] The highlighted part corrects minor errors that occurred in Eqs 34 and 35 from [7]

module $\psi(\cdot)$ directly predict the direction $\mathbf{v}$ of the data, which is equal to $-\tilde{\mathbf{x}}_0^{\mathcal{S}}$:

$$\tilde{\mathbf{x}}_0^{\mathcal{S}} := -\mathbf{v}_{\psi}(\mathbf{x}_T^{\mathcal{S}}, \mathcal{C}). \tag{12}$$

Applying these conditions to Eq. 11 yields the following:

$$\tilde{\mathbf{x}}_T^{\mathcal{T}} = \sqrt{\bar{\alpha}_T^{\mathcal{T}}}\tilde{\mathbf{x}}_0^{\mathcal{S}} + \sqrt{1 - \bar{\alpha}_T^{\mathcal{T}} - \sigma^2}\mathbf{x}_T^{\mathcal{S}} + \sigma\mathbf{z} \tag{13}$$

D.2. Additional Comments

Alternative training targets for OMS As we discussed in Sec 3.2, the objective of $\mathbf{v}$-prediction at SNR=0 scenario is exactly the same as negative $\mathbf{x}_0$-prediction. Thus we can also train the OMS module under the L2 loss between $\|\mathbf{x}_0 - \tilde{\mathbf{x}}_0\|_2^2$, where the OMS module directly predict $\tilde{\mathbf{x}}_0 = \psi(\mathbf{x}_T^{\mathcal{S}}, \mathcal{C})$.

Reasons behind versatility The key point is revealed in Eq.20. The target prediction of OMS module is only focused on the conditional mean value $\tilde{\mathbf{x}}_0$, which is only related to the training data. $\mathbf{x}_T^{\mathcal{S}}$ is directly sampled from normal distribution, which is independent. Only $\bar{\alpha}_T$ is unique to other pre-defined diffusion pipelines, but it is non-parametric. Therefore, given an $\mathbf{x}_T^{\mathcal{S}}$ and an OMS module ψ, we can calculate any $\mathbf{x}_T^{\mathcal{T}}$ that aligns with the pre-trained model schedule according to Eq.20.

Consistent generation Additionally, our study demonstrates that the OMS can significantly enhance the coherence and continuity between the generated images, which aligns with the discoveries presented in recent research [3] to improve the coherence between frames in the video generation process.

D.3. Implementation Details

Dataset The proposed OMS module and its variants were trained on the LAION 2B dataset [8] without employing any specific filtering operation. All the training images are first resized to 512 pixels by the shorter side and then randomly cropped to dimensions of 512 × 512, along with a random flip. Notably, for the model trained on the pre-trained SDXL, we utilize a resolution of 1024. Additionally, we conducted experiments on LAION-HR images with an aesthetic score greater than 5.8. However, we observed that the high-quality dataset did not yield any improvement. This suggests that the effectiveness of our model is independent of data quality, as OMS predicts the mean of training data conditioned on the prompt.

OMS scale variants We experiment with OMS modules at three different scales, and the detailed settings for each variants are shown in Table 1. Combining these with three different text encoders results in a total of nine OMS modules with different parameters. As demonstrated in Table 2, we found that OMS is not sensitive to the number of parameters and the choice of text encoder used to extract text embeddings for the OMS network.

Model	OMS-S	OMS-B	OMS-L
Layer num.	2	2	2
Transformer blocks	1	1	1
Channels	[32, 64, 64]	[160, 320, 640]	[320, 640, 1280, 1280]
Attention heads	[2, 4, 4]	8	[5, 10, 20, 20]
Cross Attn dim.	768/1024/4096	768/1024/4096	768/1024/4096
# of OMS params	3.3M/3.7M/8.1M	151M/154M/187M	831M/838M/915M

Table 1. Model scaling variants of OMS.

OMS Scale	CLIP ViT-L	OpenCLIP ViT-H	T5-XXL
OMS-S	45.87	45.30	45.35
OMS-B	46.85	45.74	45.77
OMS-L	46.68	45.65	45.19

(a) ImageReward results among different OMS scales and text encoders

OMS Scale	CLIP ViT-L	OpenCLIP ViT-H	T5-XXL
OMS-S	21.82	21.82	21.80
OMS-B	21.83	21.82	21.81
OMS-L	21.82	21.82	21.80

(b) PickScore results among different OMS scales and text encoders

Table 2. Experiment results among different OMS scales and text encoders on pre-trained SD2.1.

	FID	CLIP	ImageReward	PickScore
zsnr [5]	**12.17**	0.2586	0.3668	21.79
Ours	15.72	**0.2628**	**0.4565**	**21.82**

Table 3. Quantitative Comparison between OMS and [5].

Hyper-parameters In our experiments, we employed the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 0.01. The batch size and learning rate are adjusted based on the model scale, text encoder, and pre-trained model, as detailed in Tab. 4. Notably, our observations indicate that our model consistently converges within a relatively low number of iterations, typically around 2,000 iterations being sufficient.

Hardware and speed All our models were trained using eight 80G A800 units, and the training speeds are provided in Tab. 4. It is evident that our model was trained with high efficiency, with OMS-S using CLIP ViT-L requiring only about an hour for training.

Model	Batch size	Learning rate	Training time
OMS-S/CLIP (SD2.1)	512	5.0e-5	1.21h
OMS-B/CLIP (SD2.1)	512	5.0e-5	1.37h
OMS-L/CLIP (SD2.1)	512	5.0e-5	1.98h
OMS-S/OpenCLIP (SD2.1)	512	5.0e-5	1.21h
OMS-B/OpenCLIP (SD2.1)	512	5.0e-5	1.37h
OMS-L/OpenCLIP (SD2.1)	512	5.0e-5	2.00h
OMS-S/T5 (SD2.1)	256	3.5e-5	1.49h
OMS-B/T5 (SD2.1)	256	3.5e-5	1.56h
OMS-L/T5 (SD2.1)	256	3.5e-5	2.07h
OMS-S/OpenCLIP (SDXL)	128	2.5e-5	1.46h
OMS-B/OpenCLIP (SDXL)	128	2.5e-5	1.65h
OMS-L/OpenCLIP (SDXL)	128	2.5e-5	2.68h

Table 4. Distinct hyper-parameters and training speed on different model. All models are trained for 2k iterations using 8 80G A800.

D.4. OMS Versatility and VAE Latents Domain

The output of the OMS model is related to the training data of the diffusion phase. If the diffusion model is trained in the image domain, then our image domain-based OMS can be widely applied to these pre-trained models. However, the more popular LDM model has a VAE as the first stage that compresses the pixel domain into a latent space. For different LDM models, their latent spaces are not identical. In such cases, the training data for OMS is actually the latent compressed by the VAE Encoder. Therefore, our OMS model is versatile for pre-trained LDM models within the same VAE latent domain, *e.g.*, SD1.5, SD2.1 and LCM.

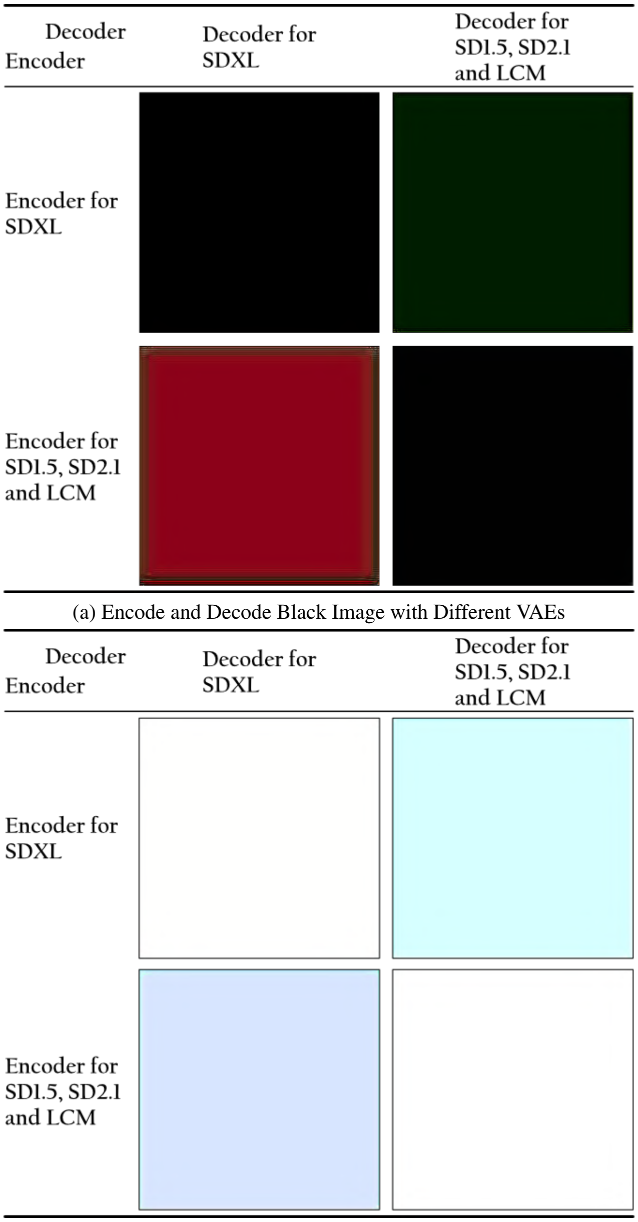
Our analysis reveals that the VAEs in SD1.5, SD2.1, and LCM exhibit a parameter discrepancy of less than 1e-4 and are capable of accurately restoring images. Therefore, we consider that these three are trained diffusion models in the same latent domain and can share the same OMS module. However, for SDXL, our experiments found significant deviations in the reconstruction process, especially in more extreme cases as shown in Fig. 3. Therefore, the OMS module for SDXL needs to be trained separately. But it can still be compatible with other models in the community based on SDXL.

If we forcibly use the OMS trained with the VAE of the SD1.5 series on the base model of SDXL, severe color distortion will occur whether we employ latents with unit variance. We demonstrate some practical distortion case with the rescaled unit variance space in Fig. 4. The observed color shift aligns with the effect shown in Fig. 3, *e.g.*, Black → Red.

E. More Experimental Results

E.1. Comparison with [5]

We further evaluate the released model [2] of [5] on the COCO 10k as shown in 3.


Decoder
Encoder
Decoder for SDXL
Decoder for SD1.5, SD2.1 and LCM
Encoder for SDXL
Encoder for SD1.5, SD2.1 and LCM
Decoder
Encoder
Decoder for SDXL
Decoder for SD1.5, SD2.1 and LCM
Encoder for SDXL
Encoder for SD1.5, SD2.1 and LCM

(a) Encode and Decode Black Image with Different VAEs

(b) Encode and Decode White Image with Different VAEs

Figure 3. The offset in compression and reconstruction of different series of VAEs.

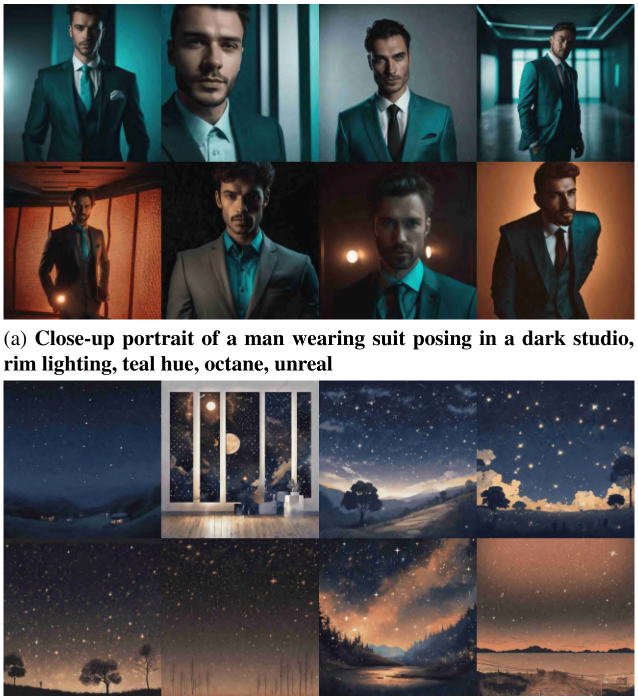
E.2. LoRA and Community Models

In this experiment, we selected a popular community model *GhostMix 2.0 BakedVAE* [3] and a LoRA *MoXin 1.0* [4]. In Fig. 6 & Fig. 7, we see that the OMS module can be applied to many scenarios with obvious effects. LoRA scale is set as 0.75 in the experiments. We encourage readers to adopt our method in a variety of well-established open-

[2] The model can be download from here

[3] *GhostMix* can be found at `https://civitai.com/models/36520`

[4] *MoXin* can be found at `https://civitai.com/models/12597`



(a) **Close-up portrait of a man wearing suit posing in a dark studio, rim lighting, teal hue, octane, unreal**

(b) **A starry sky**

Figure 4. Examples of distortion due to incompatible VAEs. Use the OMS model trained on SD1.5 VAE to forcibly conduct inference on SDXL base model. The upper layer of each subfigure shows the results sampled using the original model, while the lower layer shows the results of inference using the biased OMS model.

source models to enhance the light and shadow effects in generated images.

We also do some experiment on LCM-LoRA [6] with SDXL for fast inference. The OMS module is the same as we used for SDXL.

E.3. Additional Results

Here we demonstrate more examples based on SD1.5 Fig. 8, SD2.1 Fig. 9 and LCM Fig. 10 with OMS. In each subfigure, top row are the images directly sampled from raw pretrained model, while bottom row are the results with OMS. In this experiment, all three pre-trained base model *share the same OMS module*.

Limitations

We believe that the OMS module can be integrated into the student model through distillation, thereby reducing the cost of the additional step. Similarly, in the process of training from scratch or fine-tuning, we can also incorporate the OMS module into the backbone model, only needing to assign a pseudo-t condition to the OMS. However, doing so would lead to changes in the pre-trained model parameters, and thus is not included in the scope of discussion of this work.

References

[1] Avrim Blum, John Hopcroft, and Ravindran Kannan. *Foundations of data science*. Cambridge University Press, 2020. 1

[2] Martin Nicolas Everaert, Athanasios Fitsios, Marco Bocchio, Sami Arpa, Sabine Süsstrunk, and Radhakrishna Achanta. Exploiting the signal-leak bias in diffusion models. In *WACV*, pages 4025–4034, 2024. 1

[3] Rohit Girdhar, Mannat Singh, Andrew Brown, Quentin Duval, Samaneh Azadi, Sai Saketh Rambhatla, Akbar Shah, Xi Yin, Devi Parikh, and Ishan Misra. Emu video: Factorizing text-to-video generation by explicit image conditioning, 2023. 4

[4] Nicholas Guttenberg and CrossLabs. Diffusion with offset noise. Online Webpage, 2023. 1

[5] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. In *WACV*, pages 5404–5411, 2024. 1, 4, 5

[6] Simian Luo, Yiqin Tan, Suraj Patil, Daniel Gu, Patrick von Platen, Apolinário Passos, Longbo Huang, Jian Li, and Hang Zhao. LCM-LoRA: A universal stable-diffusion acceleration module. *arXiv preprint arXiv:2311.05556*, 2023. 6

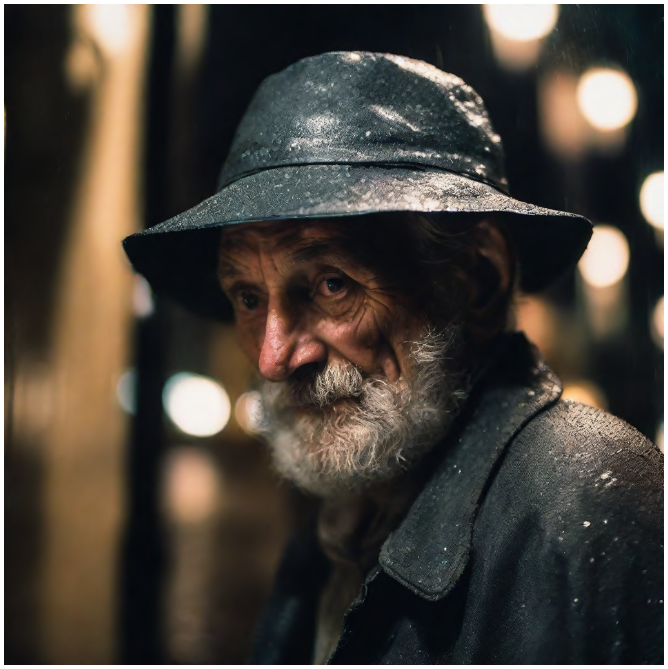
[7] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 2, 3

[8] 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. 4

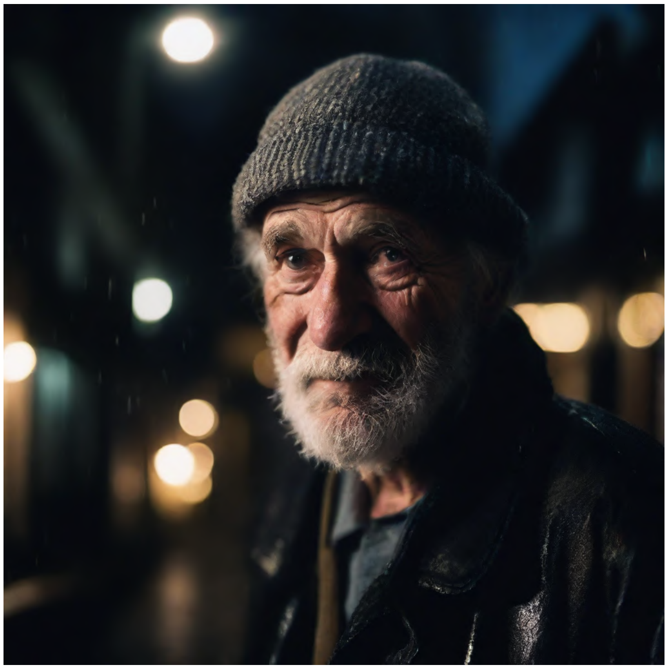
[9] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 3

[10] 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. 1

[11] Ye Zhu, Yu Wu, Zhiwei Deng, Olga Russakovsky, and Yan Yan. Unseen image synthesis with diffusion models. *arXiv preprint arXiv:2310.09213*, 2023. 1, 2

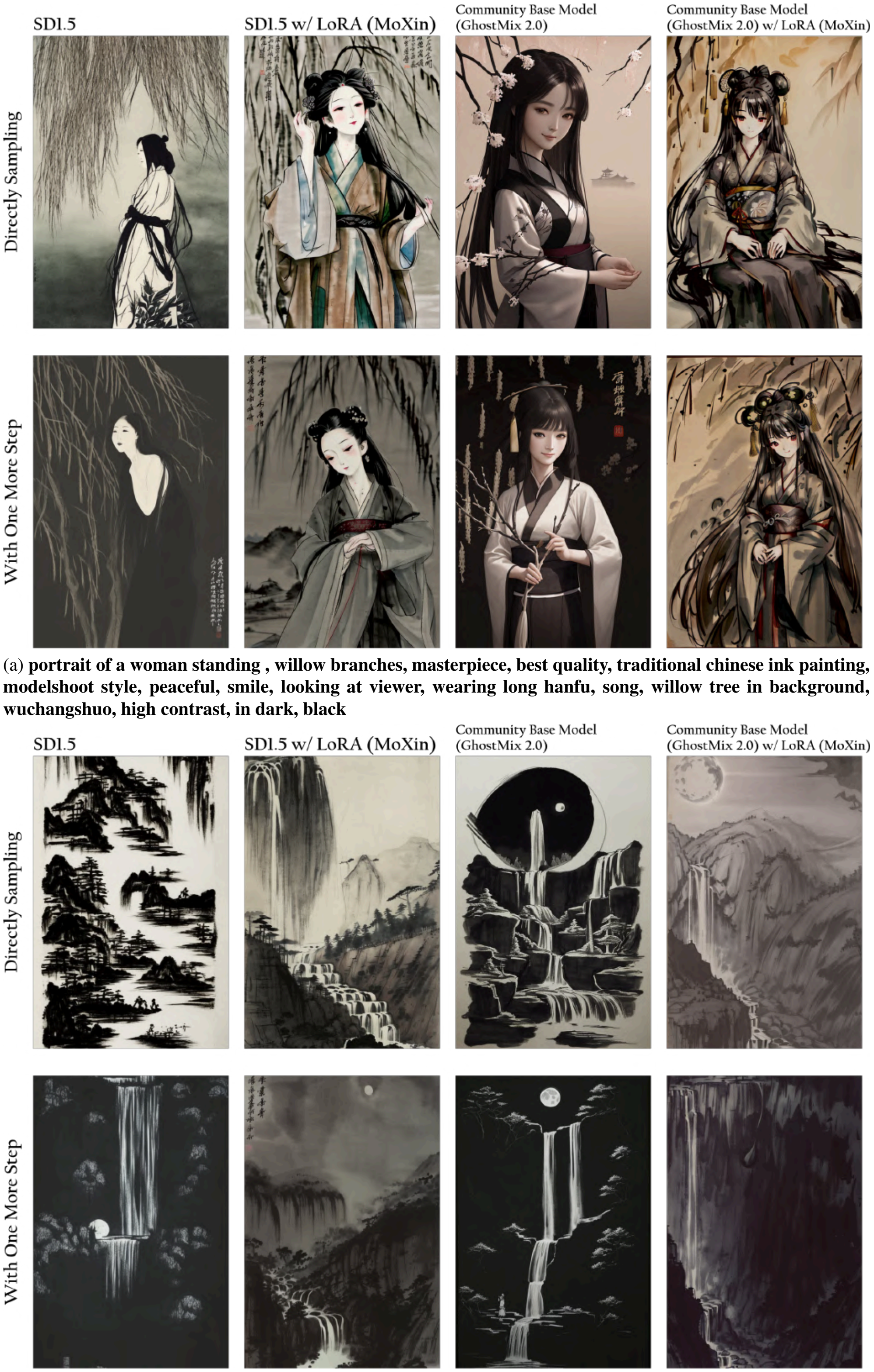


(a) **close-up photography of old man standing in the rain at night, in a street lit by lamps, leica 35mm summilux**, SDXL with LCM-LoRA, LCM Scheduler with 4 Steps. CFG weight is 1 (no CFG), Seed 1337. Mean value is 0.24.



(b) **close-up photography of old man standing in the rain at night, in a street lit by lamps, leica 35mm summilux**, SDXL with LCM-LoRA, LCM Scheduler with 4 + 1 (OMS) Steps. Base model CFG weight is 1 and OMS CFG weight is 2. Seed 1337. Mean value is **0.14**.

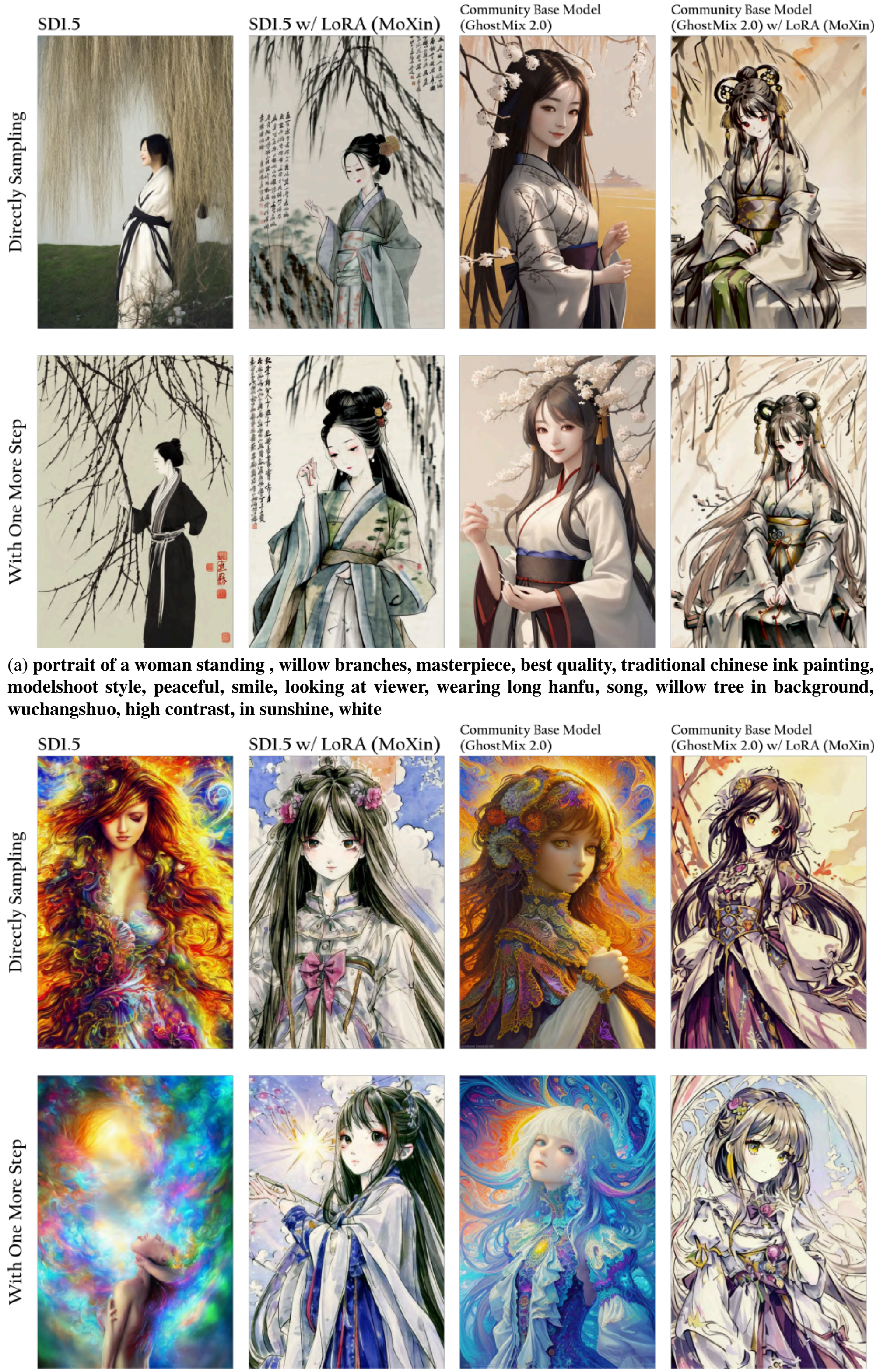
Figure 5. LCM-LoRA on SDXL for the reproduced result.


SD1.5
SD1.5 w/ LoRA (MoXin)
Community Base Model (GhostMix 2.0)
Community Base Model (GhostMix 2.0) w/ LoRA (MoXin)
Directly Sampling
With One More Step
SD1.5
SD1.5 w/ LoRA (MoXin)
Community Base Model (GhostMix 2.0)
Community Base Model (GhostMix 2.0) w/ LoRA (MoXin)
Directly Sampling
With One More Step

(a) **portrait of a woman standing , willow branches, masterpiece, best quality, traditional chinese ink painting, modelshoot style, peaceful, smile, looking at viewer, wearing long hanfu, song, willow tree in background, wuchangshuo, high contrast, in dark, black**

(b) **The moon and the waterfalls, night, traditional chinese ink painting, modelshoot style, masterpiece, high contrast, in dark, black**

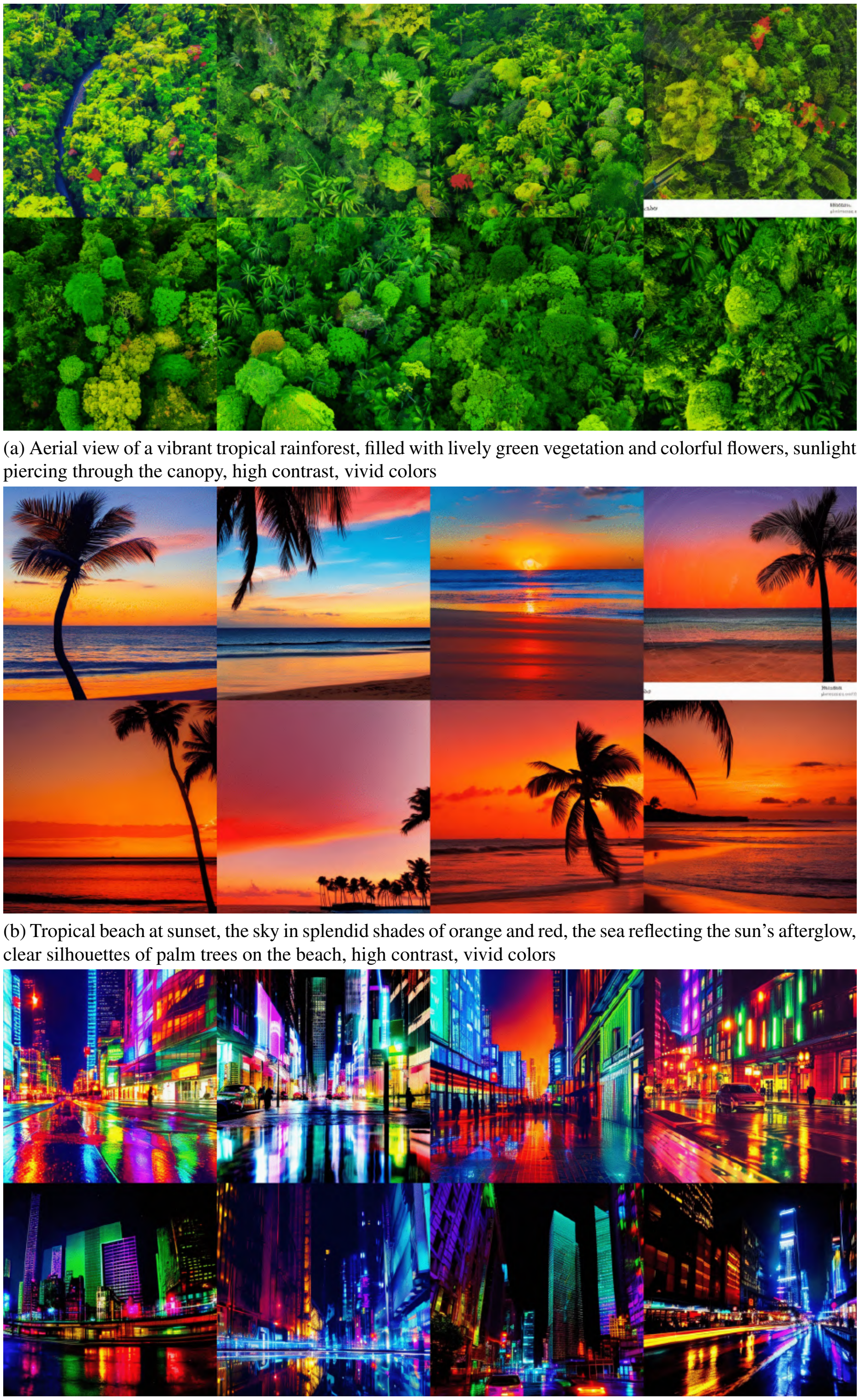
Figure 6. Examples of SD1.5, Community Base Model *GhostMix* and LoRA *MoXin* with OMS leading to darker images.


SD1.5
SD1.5 w/ LoRA (MoXin)
Community Base Model (GhostMix 2.0)
Community Base Model (GhostMix 2.0) w/ LoRA (MoXin)
Directly Sampling
With One More Step
SD1.5
SD1.5 w/ LoRA (MoXin)
Community Base Model (GhostMix 2.0)
Community Base Model (GhostMix 2.0) w/ LoRA (MoXin)
Directly Sampling
With One More Step

(a) **portrait of a woman standing , willow branches, masterpiece, best quality, traditional chinese ink painting, modelshoot style, peaceful, smile, looking at viewer, wearing long hanfu, song, willow tree in background, wuchangshuo, high contrast, in sunshine, white**

(b) **(masterpiece, top quality, best quality, official art, beautiful and aesthetic:1.2), (1girl), extreme detailed,(fractal art:1.3),colorful,highest detailed, high contrast, in sunshine, white**

Figure 7. Examples of SD1.5, Community Base Model *GhostMix* and LoRA *MoXin* with OMS leading to brighter images.

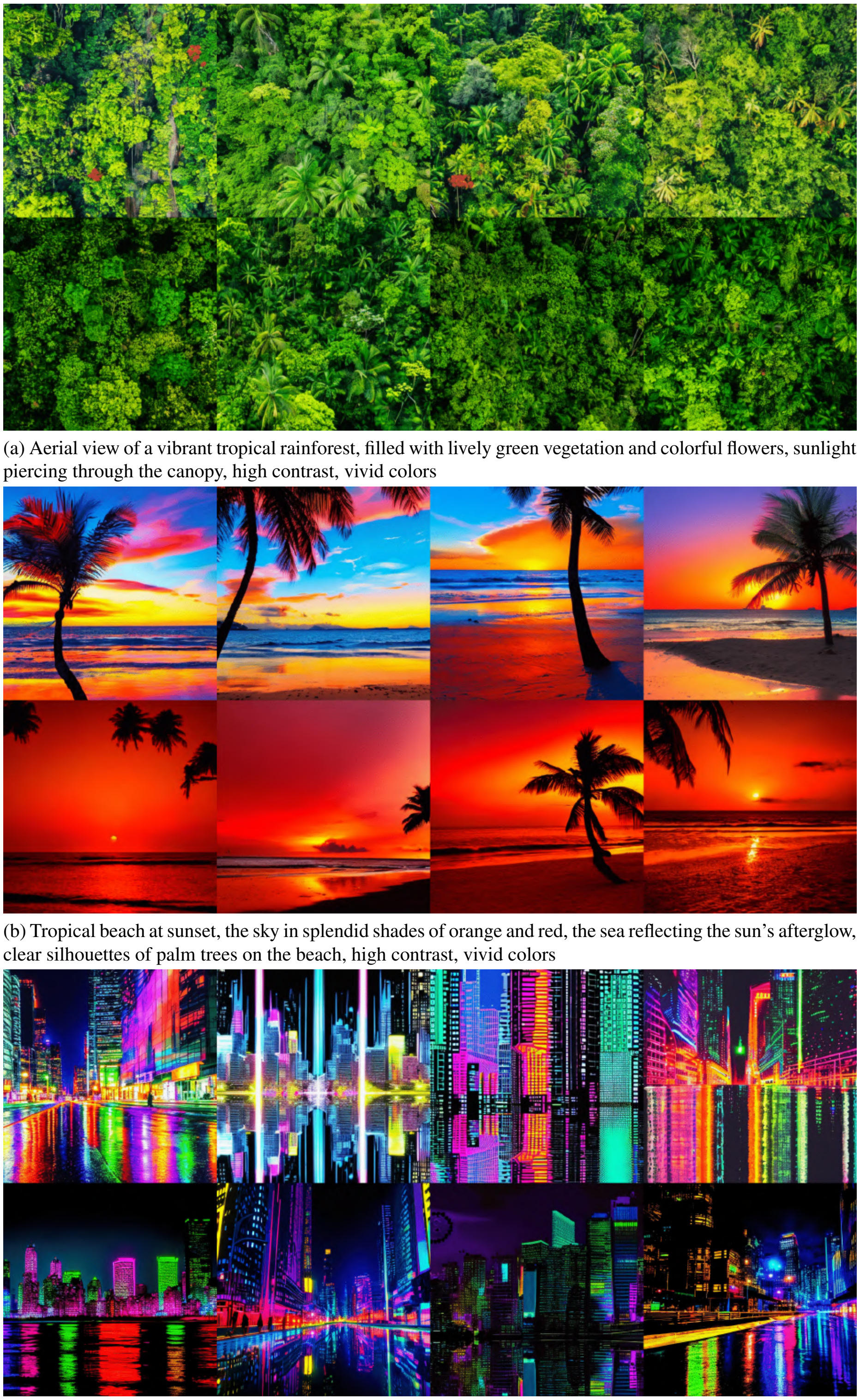


(a) Aerial view of a vibrant tropical rainforest, filled with lively green vegetation and colorful flowers, sunlight piercing through the canopy, high contrast, vivid colors

(b) Tropical beach at sunset, the sky in splendid shades of orange and red, the sea reflecting the sun's afterglow, clear silhouettes of palm trees on the beach, high contrast, vivid colors

(c) A cityscape at night with neon lights reflecting off wet streets, towering skyscrapers illuminated in a kaleidoscope of colors, high contrast between the bright lights and dark shadows

Figure 8. Additional Samples from SD1.5, top row from original model and bottom row with OMS.

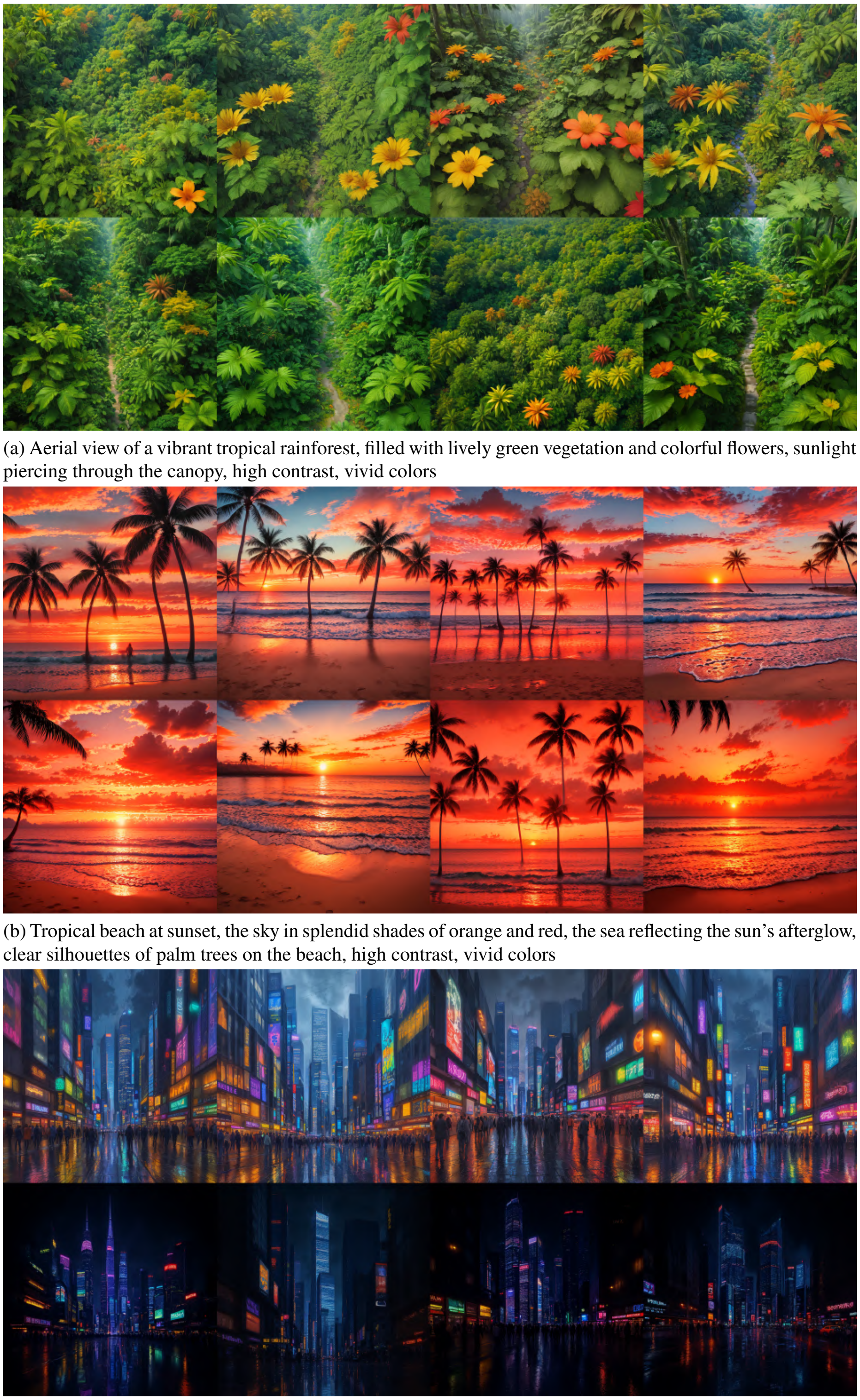


(a) Aerial view of a vibrant tropical rainforest, filled with lively green vegetation and colorful flowers, sunlight piercing through the canopy, high contrast, vivid colors

(b) Tropical beach at sunset, the sky in splendid shades of orange and red, the sea reflecting the sun's afterglow, clear silhouettes of palm trees on the beach, high contrast, vivid colors

(c) A cityscape at night with neon lights reflecting off wet streets, towering skyscrapers illuminated in a kaleidoscope of colors, high contrast between the bright lights and dark shadows

Figure 9. Additional Samples from SD2.1, top row from original model and bottom row with OMS.



(a) Aerial view of a vibrant tropical rainforest, filled with lively green vegetation and colorful flowers, sunlight piercing through the canopy, high contrast, vivid colors

(b) Tropical beach at sunset, the sky in splendid shades of orange and red, the sea reflecting the sun's afterglow, clear silhouettes of palm trees on the beach, high contrast, vivid colors

(c) A cityscape at night with neon lights reflecting off wet streets, towering skyscrapers illuminated in a kaleidoscope of colors, high contrast between the bright lights and dark shadows

Figure 10. Additional Samples from LCM, top row from original model and bottom row with OMS.