Omegance: A Single Parameter for Various Granularities in Diffusion-Based Synthesis

Supplementary Material

This supplemental material mainly contains:

- Representations of δ_t and ζ_t in Equ. (2) for different schedulers in Sec. A.
- Alternative implementation of Omegance for flow matching scheduler in Sec. B.
- Step-by-step derivations of the modified SNR in Equ. (5) in Sec. C.
- Implementation of example omega schedules in Fig. 6 in Sec. D.
- More image-to-image generation applications in Sec. E.
- More text-to-video results in Sec. F.
- Calculation of High-Frequency Energy (HFE) in Sec. G.
- A detailed description of user study setup in Sec. H.
- More discussions in Sec. I.
- More implementation details in Sec. J.
- More qualitative results in Sec. K.

A. Representations of δ_t and ζ_t

Equation 2 (recapped below) provides a general representation of one denoising step in the diffusion reverse process. We provide detailed formulas of how to convert different denoising schedulers to the general representation.

$$z_{t-1} = \delta_t \cdot z_t + \underbrace{\zeta_t \cdot \epsilon_\theta(z_t, t)}_{\text{“direction pointing to } z_0\text{”}}$$

(1) DDIM scheduler [13]:

$$\begin{aligned} z_{t-1} &= \sqrt{\alpha_{t-1}} \underbrace{\frac{z_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(z_t)}{\sqrt{\alpha_t}}}_{\text{“predicted } z_0\text{”}} + \underbrace{\sqrt{1-\alpha_{t-1}} \cdot \epsilon_\theta^{(t)}(z_t)}_{\text{“direction pointing to } z_t\text{”}} \\ &= \delta_t \cdot z_t + \zeta_t \cdot \epsilon_\theta(z_t, t) \end{aligned}$$

where

$$\begin{aligned} \delta_t &= \frac{\sqrt{\alpha_{t-1}}}{\sqrt{\alpha_t}} \\ \zeta_t &= -\sqrt{\alpha_{t-1}} \cdot \frac{\sqrt{1-\alpha_t}}{\sqrt{\alpha_t}} + \sqrt{1-\alpha_{t-1}} \end{aligned}$$

(2) Euler discrete scheduler [4]:

$$\begin{aligned} z_{t-1} &= z_t + (\sigma_{t+1} - \hat{\sigma}) \cdot \epsilon_\theta(z_t, t) \\ &= \delta_t \cdot z_t + \zeta_t \cdot \epsilon_\theta(z_t, t) \end{aligned}$$

where

$$\begin{aligned} \delta_t &= 1 \\ \zeta_t &= \sigma_{t+1} - \hat{\sigma} \end{aligned}$$

(3) Flow-matching scheduler [1]:

$$\begin{aligned} z_{t-dt} &= z_t + dt \cdot v_\theta(z_t, t) \\ &= \delta_t \cdot z_t + \zeta_t \cdot \epsilon_\theta(z_t, t) \end{aligned}$$

where

$$\begin{aligned} \delta_t &= 1 \\ \zeta_t \cdot \epsilon_\theta(z_t, t) &\approx dt \cdot v_\theta(z_t, t) \end{aligned}$$

B. Alternative Implementation of Omegance in Flow Matching Scheduler

We additionally explored an alternative implementation of Omegance for flow-matching-based sampling. Given the clean latents z_0 and Gaussian noise $\epsilon \sim N(0, 1)$, noisy latents in flow-matching scheduling is $z_t = (1-t)z_0 + t \cdot \epsilon$. Therefore, $v_t = \frac{dz_t}{dt} = \epsilon - z_0$.

Given the model predicted velocity $v_\theta(z_t, t)$ at time t, the estimated clean latents would be:

$$\hat{z_0} = z_t + t \cdot v_\theta$$

Then the estimated Gaussian noise is

$$\hat{\epsilon} = v_\theta + \hat{z_0} = z_t + (t+1) \cdot v_\theta$$

We then apply Omegance by scaling the estimated noise:

$$\hat{\epsilon}' = \omega \cdot \hat{\epsilon}$$

The updated velocity becomes

$$\begin{aligned} v'_\theta &= \hat{\epsilon}' - \hat{z_0} \\ &= \omega \cdot (v_\theta + \hat{z_0}) - \hat{z_0} \\ &= \omega \cdot v_\theta + (\omega - 1)\hat{z_0} \\ &= \omega \cdot v_\theta + (\omega - 1)(z_t + t \cdot v_\theta) \\ &= (\omega - 1)z_t + (\omega + t \cdot \omega - t)v_\theta \end{aligned}$$

Compared to our main implementation, which directly modulates the velocity as $v'_\theta = \omega \cdot v_\theta$, this formulation introduces a dependence on the estimated Gaussian noise. Empirically, both approaches produce comparable outputs. Conceptually, our method better aligns with the continuous nature of flow-matching, treating ϵ as a fixed endpoint rather than a step-wise quantity.

C. Modified SNR Derivations

In DDIM denoising process,

$$z_{t-1} = \sqrt{\alpha_{t-1}} \underbrace{\frac{z_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(z_t)}{\sqrt{\alpha_t}}}_{\text{“predicted } z_0\text{”}} + \underbrace{\sqrt{1-\alpha_{t-1}} \cdot \epsilon_\theta^{(t)}(z_t)}_{\text{“direction pointing to } z_t\text{”}}$$
$$= \sqrt{\alpha_{t-1}}\hat{z}_0 + \sqrt{1-\alpha_{t-1}}\epsilon_\theta$$
$$\approx \sqrt{\alpha_{t-1}}\hat{z}_0 + \sqrt{1-\alpha_{t-1}}\epsilon$$

So the original SNR during the reverse process is still: $\text{SNR}(t-1) = \frac{\alpha_{t-1}}{1-\alpha_{t-1}}$. In the derivation above, we model the noise prediction ϵ_θ as an ideal approximation of the actual noise $\epsilon \sim N(0,1)$. While ϵ_θ is technically the minimum MSE estimator of ϵ and may not follow the exact same distribution, this approximation allows for a tractable and interpretable formulation of how the effective SNR evolves under different ω settings. Such ideal approximation is used in modified SNR derivations as well.

Equation 5 shows the modified SNR based on DDIM denoising equation. The step-by-step derivations are provided below:

$$z'_{t-1} = \sqrt{\alpha_{t-1}}\frac{z_t - \sqrt{1-\alpha_t}\epsilon_\theta \cdot \omega}{\sqrt{\alpha_t}} + \sqrt{1-\alpha_{t-1}}\epsilon_\theta \cdot \omega$$
$$\approx \sqrt{\alpha_{t-1}}\frac{z_t - \sqrt{1-\alpha_t}\epsilon \cdot \omega}{\sqrt{\alpha_t}} + \sqrt{1-\alpha_{t-1}}\epsilon \cdot \omega$$

Substitute $z_t = \sqrt{\alpha_t}z_0 + \sqrt{1-\alpha_t}\epsilon$ into the equation and isolate z_0 and ϵ terms, we get:

$$z'_{t-1} = Az_0 + B\epsilon$$

where

$$A = \sqrt{\alpha_{t-1}}$$
$$B = \frac{\sqrt{\alpha_{t-1}}\sqrt{1-\alpha_t}(1-\omega) + \sqrt{\alpha_t}\sqrt{1-\alpha_{t-1}}\omega}{\sqrt{\alpha_t}}$$

Therefore, the modified SNR is:

$$\text{SNR}' = \frac{A^2}{B^2}$$
$$= \frac{\alpha_{t-1}}{\frac{(\sqrt{\alpha_{t-1}}\sqrt{1-\alpha_t}(1-\omega)+\sqrt{\alpha_t}\sqrt{1-\alpha_{t-1}}\omega)^2}{\alpha_t}}$$
$$= \frac{\alpha_{t-1}}{\left(\frac{\sqrt{\alpha_{t-1}}\sqrt{1-\alpha_t}}{\sqrt{\alpha_t}} + \frac{\sqrt{\alpha_t}\sqrt{1-\alpha_{t-1}}-\sqrt{\alpha_{t-1}}\sqrt{1-\alpha_t}}{\sqrt{\alpha_t}}\omega\right)^2}$$
$$\propto \omega$$

Since α_t monotonically decreases in diffusion model, so $\alpha_{t-1} > \alpha_t$ and $1-\alpha_t > 1-\alpha_{t-1}$. Therefore, $\sqrt{\alpha_t}\sqrt{1-\alpha_{t-1}} - \sqrt{\alpha_{t-1}}\sqrt{1-\alpha_t}$ is always negative and SNR' increases as ω increases.

D. Omega Schedule Examples Implementation

Detailed implementations of four example omega schedules shown in Fig. 6(a) are listed below. Please take notes that these examples are only for illustration. Users are free to design their own schedules following the effects in Fig. 4.

```python
import numpy as np

def exponential_scheduler(initial_lr, min_lr, gamma, steps):
    return [(initial_lr - min_lr) * (gamma ** step) + min_lr for step in range(steps)]

def cosine_scheduler(initial_lr, min_lr, steps, alpha=1.0):
    lr_schedule = []
    for step in range(steps):
        lr = min_lr + 0.5 * (initial_lr - min_lr) * (1 + np.cos(alpha * np.pi * step / steps))
        lr = min(lr, initial_lr if step == 0 else lr_schedule[-1])
        lr_schedule.append(lr)
    return lr_schedule

# Mirrored functions w.r.t. y = 1.0
def exponential_scheduler_mirrored(initial_lr, min_lr, gamma, steps):
    original = exponential_scheduler(initial_lr, min_lr, gamma, steps)
    return [2 - lr for lr in original]

def cosine_scheduler_mirrored(initial_lr, min_lr, steps, alpha=1.0):
    original = cosine_scheduler(initial_lr, min_lr, steps, alpha)
    return [2 - lr for lr in original]

# Parameters
steps = 50

# Omega schedulers
exp1 = exponential_scheduler_mirrored(1.05, 1.0, 0.9, steps)
exp2 = exponential_scheduler_mirrored(1.1, 0.9, 0.9, steps)
cos1 = cosine_scheduler_mirrored(1.0, 0.95, steps, 1.5)
cos2 = cosine_scheduler(1.1, 0.9, steps, 0.9)
```

Listing 1. Implementation details of omega schedule examples shown in Fig. 6(a).

E. More Image-to-Image Generation

SDEdit. For image-to-image editing using SDEdit [7], we use the off-the-shelf semantic segmentation tool SAM2 [9] to generate the segmentation masks for the input images, enabling the application of omega masks to specific segmentation regions. Results are demonstrated in Fig. A.

Real Image Editing. Omegance can also be applied spatially in real-image inversion tasks to achieve granularity editing of specific objects. The masks can be obtained from

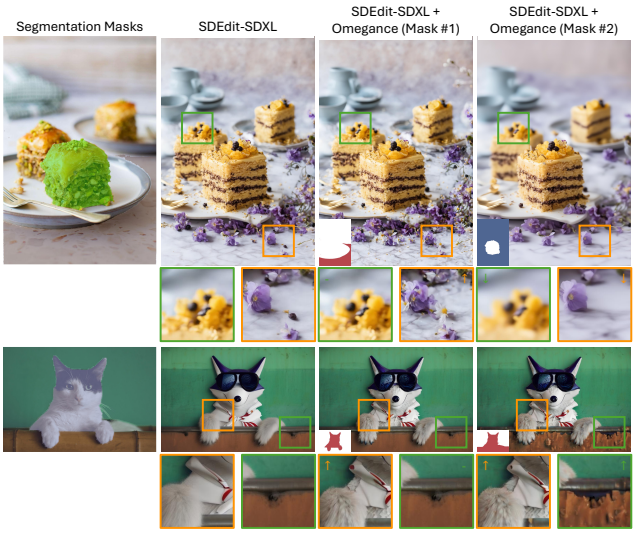


Figure A. Spatial effects of mask-based Omegance in SDEdit results. Segmentation masks are obtained using SAM2 [9]. Omegance can achieve precise spatial granularity control without altering untouched regions.

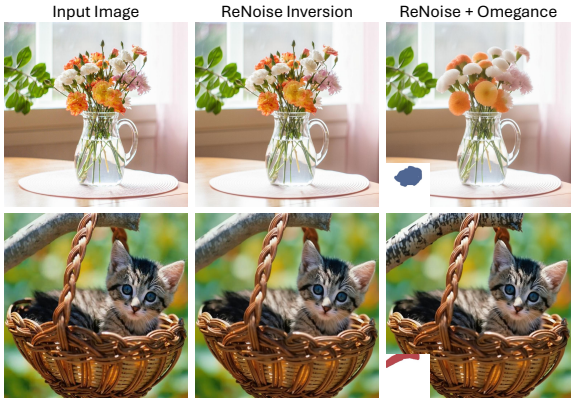


Figure B. Spatial effects of mask-based Omegance in ReNoise inversion results. By inverting an image and applying Omegance with desired masks, the users can easily modify the granularity of any real images.

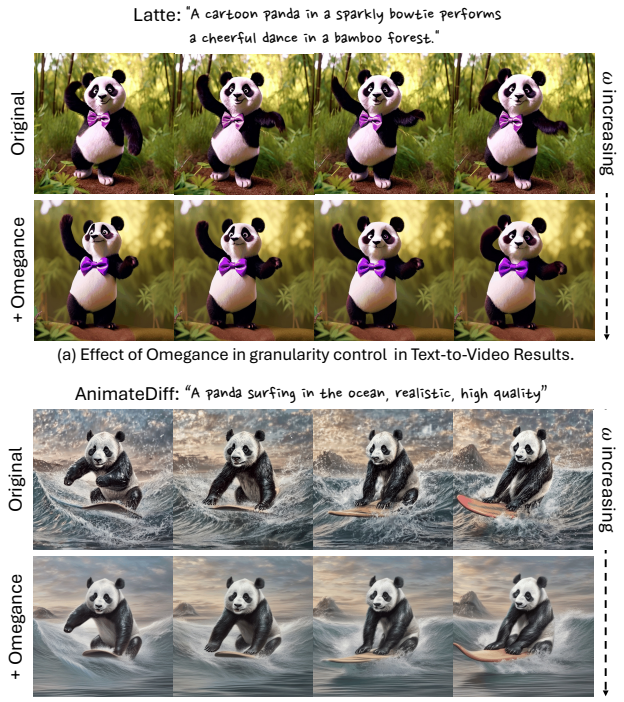
either segmentation tools or user-provided strokes. The results using ReNoise inversion [2] are presented in Fig. B.

Image Inpainting. We further generalize the use of Omegance to image inpainting task. Figure C shows the results of performing image inpainting using SDXL [8].

F. More Text-to-Video Generation

Omegance's granularity control ability in more text-to-video applications, *e.g.*, Latte [6] and AnimateDiff [3], are demonstrated in Fig. D. In Fig. D (a), "ω increasing" (detail suppression) leads to a less complex background and smoother texture, which highlights the main character and presents a more visually pleasing result. Additionally, current text-to-video generation techniques often suffer from

Figure C. Effects of Omegance in image inpainting task. Omegance can effectively control the granularity of infilling region, making it more flexible for the users to get their desired results.



(a) Effect of Omegance in granularity control in Text-to-Video Results.

(b) Effect of Omegance in fixing visual artifacts in Text-to-Video Results.

Figure D. Effects of Omegance in Text-to-Video results. The employed T2V models and prompts are shown above, with ω variations on the right. Omegance enables granularity control in generated videos while preserving temporal coherence and can occasionally fix artifacts.

visual artifacts in the output, as illustrated in the first row of Fig. D (b), where an unintended flow effect appears in the sky. Applying Omegance can effectively remove the above artifact as demonstrated in the second row of Fig. D (b).

G. High-Frequency Energy Calculation

A detailed calculation of the high-frequency energy (HFE) of an image is shown below. The final reported Mean HFE is averaged across all generated images.

```
def high_frequency_energy(image_tensor):
    """
```

```
    Measure the high-frequency energy of an image

    Args:
    image_tensor (torch.Tensor): Input image
    tensor of shape (C, H, W) for RGB or (H, W)
    for grayscale. Values should be normalized to
     [0, 1] or [0, 255].

    Returns:
    float: Mean high-frequency energy of the
    image.
    """
    # Convert RGB to grayscale
    if len(image_tensor.shape) == 3 and
    image_tensor.shape[0] == 3:  # RGB image
        image_tensor = rgb_to_grayscale(
    image_tensor)

    # Ensure the image is grayscale
    if len(image_tensor.shape) != 2:
        raise ValueError("Input must be a
    grayscale image of shape (H, W).")

    # Compute the Fourier Transform
    dft = torch.fft.fft2(image_tensor)

    # Shift the zero-frequency component to the
    center
    dft_shifted = torch.fft.fftshift(dft)

    # Compute the magnitude spectrum
    magnitude = torch.abs(dft_shifted)

    # Define a center region to exclude low
    frequencies
    H, W = magnitude.shape
    center_x, center_y = H // 2, W // 2
    radius = min(H, W) // 8  # Exclude the lowest
     1/8th frequency

    # Create a high-pass mask
    y, x = torch.meshgrid(torch.arange(H), torch.
    arange(W), indexing="ij")
    mask = ((x - center_x)**2 + (y - center_y)
    **2) > radius**2

    # Apply the mask to extract high frequencies
    high_freq_magnitude = magnitude * mask

    # Compute the mean high-frequency energy
    mean_high_freq_energy = torch.mean(
    high_freq_magnitude**2).item()

    return mean_high_freq_energy
```

Listing 2. Implementation details of High-Frequency Energy as measurement for detail level.

H. User Study

To demonstrate the effectiveness of Omegance in controlling image granularity and preserving image quality, we design a two-part user study.

In Part 1, for each question, the participants are given three images, one generated by the base model, and the other two by the base model with Omegance. The participants are asked to select the rank of image granularity (from high to low) that best matches their inspection. Instructions about granularity are given: *Granularity in the context of image generation refers to the level of detail and texture richness in a visual output. High granularity corresponds to intricate textures, complex patterns, and rich visual density, while low granularity is associated with smoother transitions, minimal detail, and simpler compositions*. The motivation behind such design is that if the granularity rank can be correctly distinguished by the users, it proves the effectiveness of Omegance in controlling image granularity. The images used in Part 1 are generated by SDXL [8] and FLUX [5] using global Omegance. For quantitative evaluation, we consider each rank position separately and calculate the accuracy independently. The final average rank accuracy is the average accuracy of all rank positions.

In Part 2, our objective is to illustrate that Omegance does not harm the base model's quality and can occasionally improve it by fixing artifacts and enhancing realism. We present two contents generated by the base model with and without Omegance and ask the users to select the one with better quality. The instructions on visual quality definition are given as: *Visual quality refers to the overall perceptual appeal and coherence of a generated content, encompassing aspects such as sharpness, realism, fidelity to given prompt, and freedom from artifacts*. We believe that by achieving over 50% votes in quality competition, it is sufficient to prove that Omegance can achieve at least equal quality outcomes as the base model. The results used in Part 2 are generated by SDXL [7], FLUX [5], RealVisXL-5.0 [11], AnimateDiff [3], and Latte [6]. We report the overall percentage of votes indicating that our method achieves equal or higher quality over the base model as our quality evaluation in Tab. 3 of the main paper. Detailed percentages of separate vote rates indicating our method achieves better, equal or worse quality are shown below:

Table A. Detailed vote rate of our method showcasing better, equal, or worse quality. Most users chose our method as the one with better quality.

Better	Equal	Worse
67.62%	13.76%	18.61%

I. Discussion

In this section, we discuss several settings that achieve similar effects in granularity or motivate our design choices.

I.1. Change Inference Steps

An intuitive approach to influencing the granularity of generative outputs is to adjust the number of inference steps.

Figure E. Change of omega (left) vs. Change of number of inference steps (right). Example 1. The orange box indicates default results when $\omega = 1.0$ and the number of inference steps $= 50$.

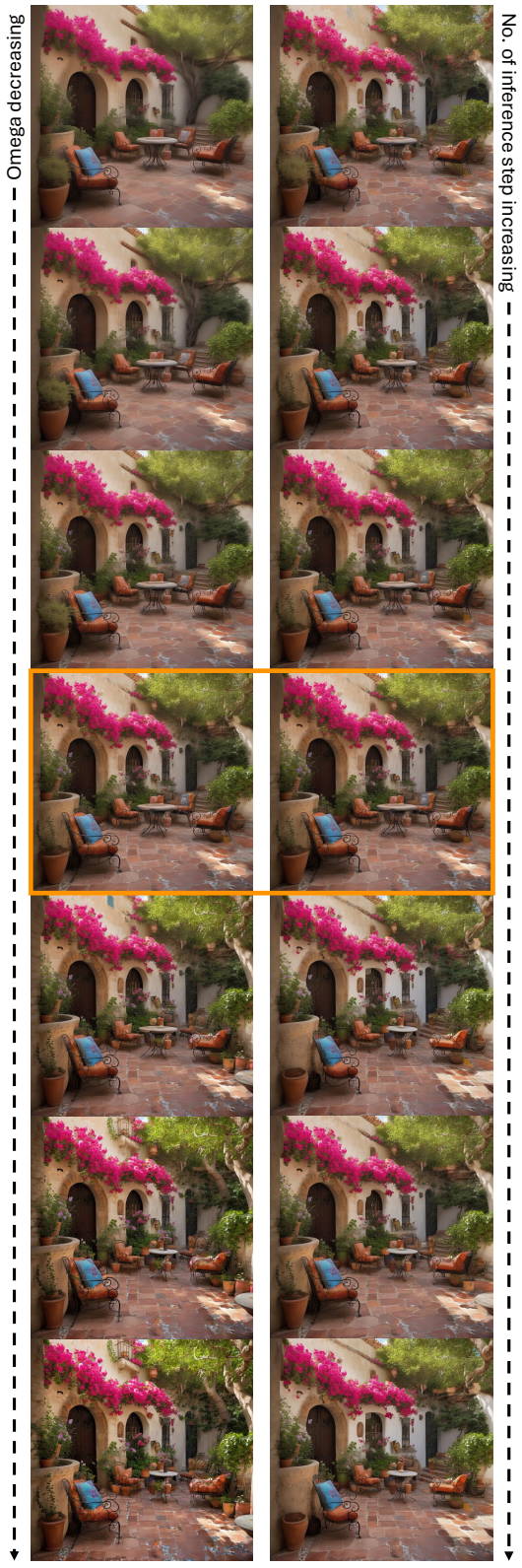


Figure F. Change of omega (left) vs. Change of number of inference steps (right). Example 2. The orange box indicates default results when $\omega = 1.0$ and the number of inference steps $= 50$.

However, as observed in our experiments, linearly increasing the number of steps **does not provide consistent granularity control**. Figure E and F demonstrate this using SDXL [8] with the Euler discrete scheduler [4], where changes in layout complexity and texture richness caused by increasing steps show irregular patterns compared to the consistent effects achieved with Omegance. Moreover, modifying the number of steps impacts the entire image globally, offering **no capacity for region-specific adjustments**. In contrast, Omegance enables localized control through omega masks and schedules. Omega masks allow region-specific granularity adjustments, enhancing or suppressing detail in specific regions while preserving unselected areas, and omega schedules can refine layout or texture granularity at different denoising stages. This flexibility is especially valuable for real-world design tasks, where different regions or elements require varying levels of detail. Besides, increasing steps **adds computational overhead** without guaranteeing significant improvements in granularity control, while Omegance provides granularity manipulation within a fixed inference schedule with no extra cost, maintaining computational efficiency and output fidelity.

In summary, adjusting inference steps is a coarse mechanism for granularity control, as it alters the overall denoising dynamics without offering fine-tuned control. Omegance, by operating directly within the existing denoising framework, delivers precise, user-driven adjustments without additional computational cost.

I.2. Change Latent Mean

In Omegance, we carefully preserve the mean of z_t while altering its variance. This design choice is motivated by our observation that changes in the latent mean lead to color shifts in the final decoded outputs. This phenomenon can be verified by directly modifying the mean of VAE-encoded latents in a latent diffusion model [10]. Given an original image in Fig. G(a) as x_0, we encode the image to latent z_0 using the VAE used in SD3 [1] and modify the mean of z_0 directly. The results of increasing and decreasing latent mean are shown in Fig. G(b) and (c), respectively. These results reveal that the green channel is particularly sensitive to mean changes, while the red channel is the least affected. Consequently, increasing the mean causes the green color to dominate, while decreasing it leads to a dominance of red tones, producing undesirable color shifts. To avoid such artifacts, we ensure mean preservation in the implementation of Omegance.

J. More Implementation Details

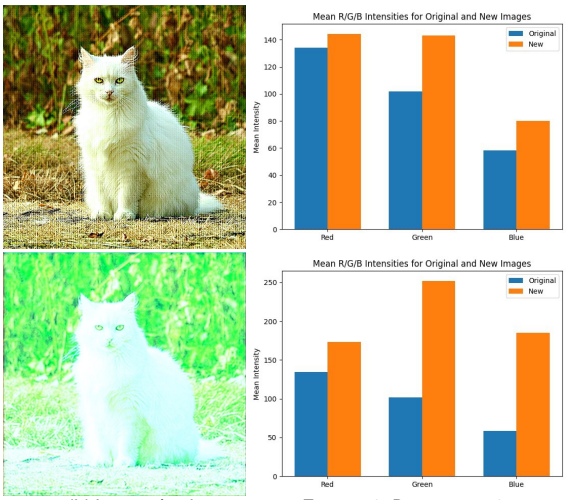
In this section, the implementation details for producing the demo contents are explained.

J.1. Prompts for Demo

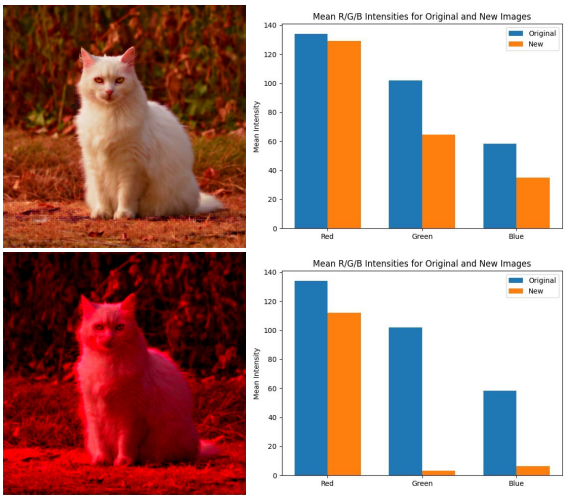
We use Large Language Model to help generate prompts.

Figure 1(a): *A serene lake at dawn, with crystal-clear water reflecting snow-capped mountains and a soft pink sky. In the foreground, delicate wildflowers bloom along the shoreline, while a small wooden rowboat drifts quietly. The smooth, mirror-like surface of the lake contrasts with the*

(a) Original image.



(b) Increasing latent mean. Top: +1.0; Bottom: +5.0



(c) Decreasing latent mean. Top: -1.0; Bottom: -5.0

Figure G. Effects of latent mean changes on image RGB mean with analysis.

rich details of the flowers, boat, and distant mountains, all softly illuminated by the gentle morning light.

Figure 1(b): *A wise wizard in a long, starry cloak stands in an enchanted forest clearing, with towering trees and vibrant mushrooms glowing softly. Around him, floating magical orbs and sparkling fireflies create a mystical ambiance, while wildflowers and ivy-covered stones add texture to the scene.*

Figure 1(c): *A bustling medieval marketplace filled with colorful tents and stalls, where merchants display spices, textiles, and jewelry. Cobblestone streets wind between the booths, and in the distance, a towering castle rises against the horizon.*

Figure 3 SDXL (Left): *A gentle healer dressed in flowing robes stands beside a calm forest pond, her hand extended over the water as soft, glowing light surrounds her. The pond's smooth surface reflects her figure and the towering trees around, while small, vibrant wildflowers dot the mossy shore.*

Figure 3 SDXL (Middle): *A peaceful Japanese zen garden at dusk, with smooth raked sand patterns surrounding moss-covered stones and a gentle, flowing stream reflecting the warm glow of lantern light.*

Figure 3 SDXL (Right): *A futuristic space station interior with rounded, smooth walls and control panels filled with colorful buttons and screens. Astronauts in sleek spacesuits float gently in zero gravity, surrounded by floating tools and shimmering holographic displays showing star maps and distant planets.*

Figure 3 SDXL + FreeU: *A bustling riverside café painted in impressionist style, where figures in soft, muted tones gather under glowing lanterns. The river reflects the lights, creating smooth, rippling patterns as the colors blend seamlessly with the surrounding trees and buildings.*

Figure 3 RealVisXL-5.0: *A serene meadow at sunrise, with a vintage picnic blanket spread out under a large oak tree. Wildflowers in pastel colors bloom across the soft grass, and a wicker picnic basket filled with freshly baked bread and fruit adds to the idyllic, pastoral scene.*

Figure 3 SD3: *A celestial observatory with smooth, polished floors and a massive domed ceiling painted with constellations in deep blues and silvers. A large telescope stands at the center, and intricate star maps and charts are scattered around, bathed in the soft, ambient light of glowing stars.*

Figure 3 FLUX: *A cozy anime café on a rainy afternoon, where customers sit at tables enjoying warm drinks as raindrops patter against the large windows. Soft, warm lighting gives the café a welcoming feel, with framed pictures, books, and cushions scattered around, adding rich detail to the snug interior.*

Figure 5: *A boy is playing Pokemon.*

Figure 6: *A quaint alpine village market in winter, with stalls selling handmade crafts, baked goods, and hot drinks, nestled between snow-covered wooden chalets. Pine trees and mountain peaks frame the background, while a soft snowfall adds charm to the festive, bustling atmosphere.*

Figure 7 SDXL Pose (Left): *Darth vander dancing in a desert, high quality.*

Figure 7 SDXL Pose (Right): *An elven warrior princess stands in a lush forest glade, her armor decorated with delicate leaf patterns and shining gemstones. Her cape is embroidered with nature motifs, and her ornate bow and quiver add a touch of elegance, creating a striking blend of nature and nobility.*

Figure 7 SDXL Depth (Left): *A topiary plant decorated by flowers.*

Figure 7 SDXL Depth (Right): *A sleek, futuristic robot with a polished metallic body and glowing blue eyes, standing upright with articulated limbs and fine details in its joints and circuitry. Soft lights reflect off its surface, highlighting its advanced design and smooth, streamlined form.*

Figure 7 SDXL Canny (Left): *Aerial view, a futuristic research complex in a bright foggy jungle, hard lighting.*

Figure 7 SDXL Canny (Right): *A stunning piece of jewelry featuring a delicate, silver pendant encrusted with sparkling diamonds and a large, brilliant-cut sapphire at its center, suspended from a fine, intricately detailed chain that glimmers in soft light.*

Figure 7 FLUX Canny (Right): *A sparkling crystal mansion stands in the middle of a blooming meadow, its walls refracting sunlight into brilliant rainbows. Delicate cherry blossom trees line the cobblestone path leading to the ornate glass doors, while butterflies flutter in the gentle breeze.*

Figure 7 FLUX Canny (Left): *A whimsical, pastel-colored cottage with a gently sloping roof and rounded windows sits amidst a vibrant garden of oversized flowers. A cobblestone path leads to the arched wooden door, and sparkling lights hang from the eaves, glowing softly under the golden afternoon sun.*

Figure A (Top): *A creamy white marble table sprinkled with tiny violet and daisy blooms supports a classic white plate with thin, golden edges. The chocolate cake, layered with dark chocolate curls and topped with vibrant berries, sits beautifully on the plate, creating a striking contrast between the lush texture of the cake and the refined elegance of the setting.*

Figure A (Bottom): *A cool fox wearing sunglasses leaning on a rusted water pipe.*

Figure B (Top): *A glass vase filled with flowers.*

Figure B (Bottom): *A kitten in a basket.*

Figure C: *A loosely arranged bouquet of wildflowers in a glass jar sits in the center of the bench, with petals and leaves spilling over the edge. The soft textures of the flowers balance the structure of the jar and bench for a touch of*

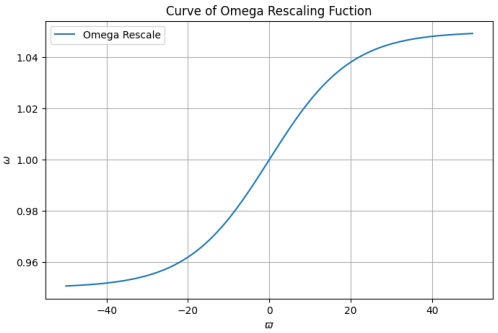


Figure H. Visualization of omega rescale function.

elegance.

J.2. Negative Prompt

We use negative prompts where applicable to help improve generation quality: *"distorted lines, warped shapes, uneven grid patterns, irregular geometry, misaligned symmetry, low quality, bad quality"*.

J.3. Omega Rescale

As mentioned in Sec. 3.2, we rescale omega to allow finer-grained control within $(-\infty, \infty)$ input range by:

$$\omega = \mathcal{R}(\varpi) = L + \frac{U - L}{1 + e^{-k \cdot \varpi}} \tag{1}$$

The default rescaling function for SDXL [8] model is visualized in Fig. H with $k = 0.1, L = 0.95, U = 1.05$. In this case, ϖ within $[-10.0, 10.0]$ would present visible effects. The sensitivity of visual changes to different ϖ values are illustrated in Fig. I. In SD3 [1] and FLUX [5], the range of ω should be larger around $[0.8, 1.2]$.

J.4. Number of Inference Steps

The default inference steps are set to 50 for SDXL [8], 4 for FLUX-Schnell [5], and 28 for SD3 [1]. However, Omegance remains compatible with other inference step configurations, provided that the base model produces valid results.

K. More Qualitative Results

In this section, we show more examples of Omegance in SDXL [8], SDXL+FreeU [12], RealVisXL-V5.0 [11], SD3 [1], FLUX [5], and ControlNet [14]. Control signals are indicated in the top left corner of the original results. Omega masks used are indicated in the bottom left corner of the corresponding Omegance-edited results. Red for detail enhancement, blue for detail suppression.

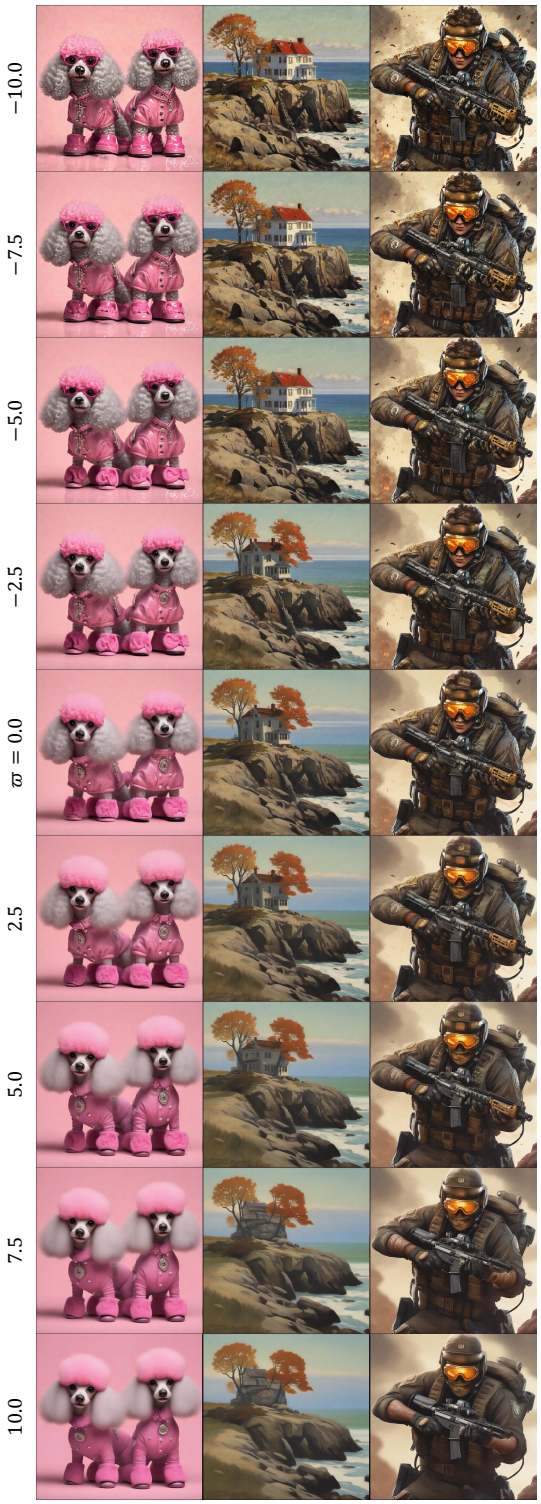


Figure I. Sensitivity of SDXL [8] granularity changes w.r.t. ϖ values with $[-10.0, 10.0]$.

References

[1] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024. 1, 6, 8

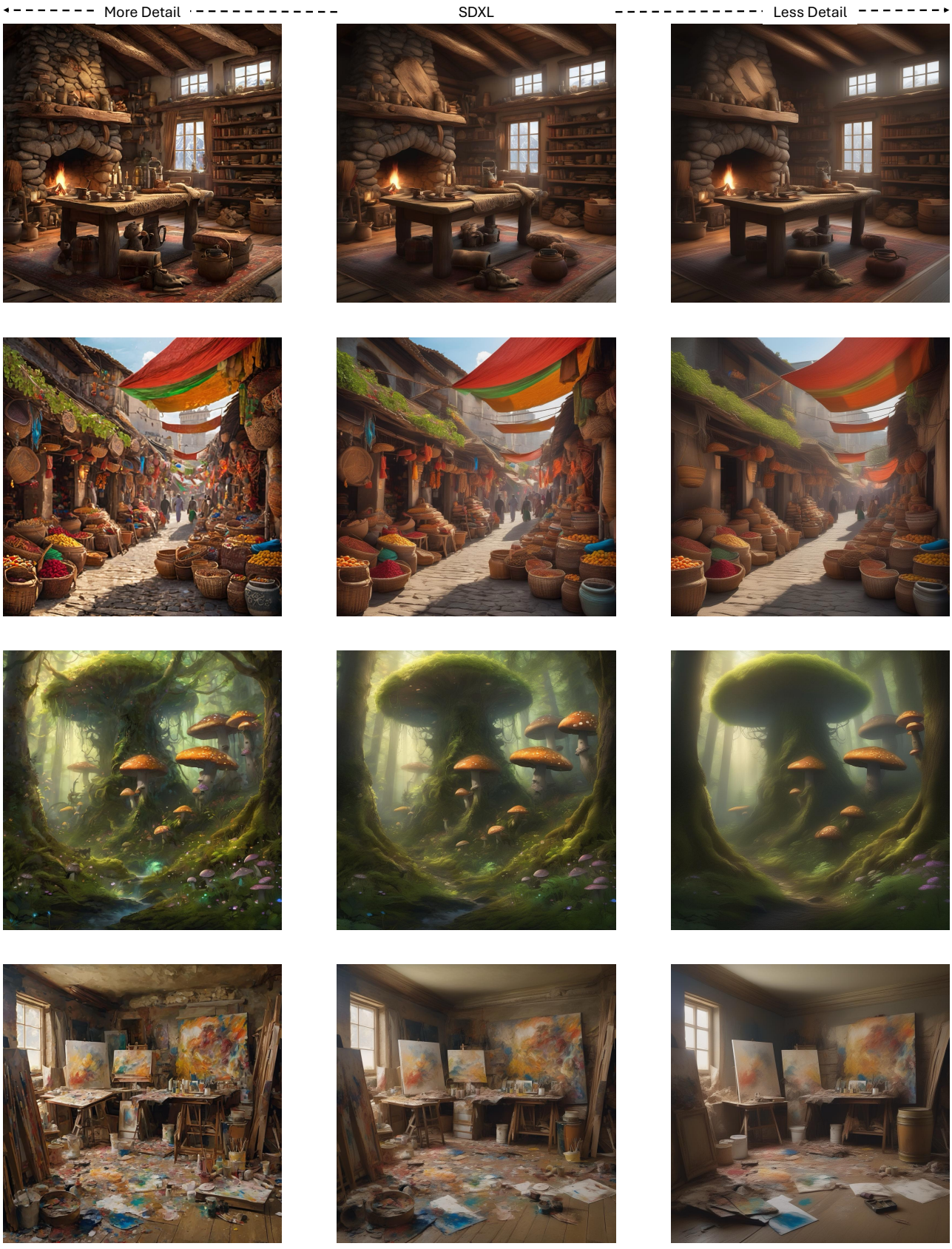


Figure J. More global effects of Omegance in SDXL.

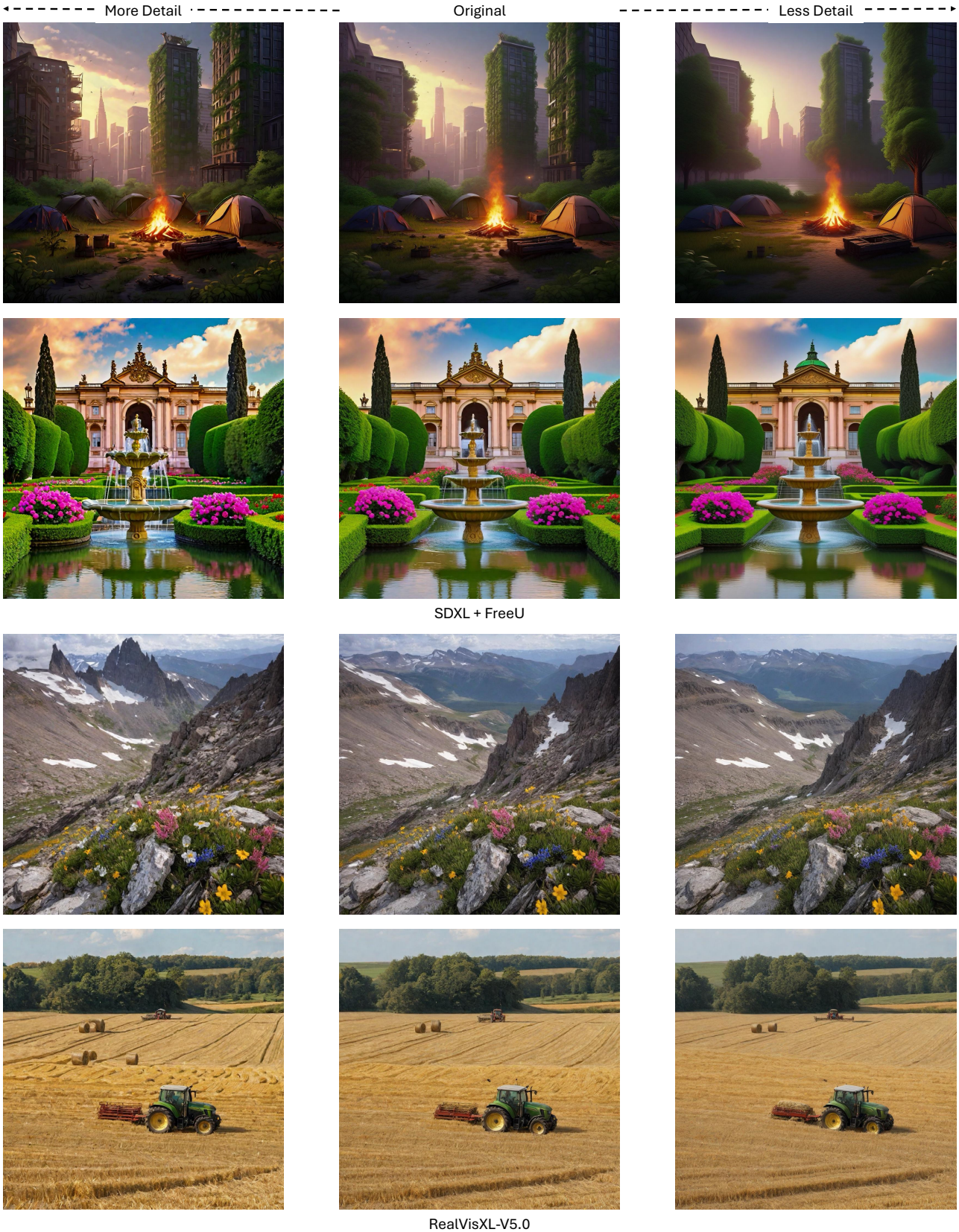


Figure K. More global effects of Omegance in SDXL+FreeU and RealVisXL-V5.0.

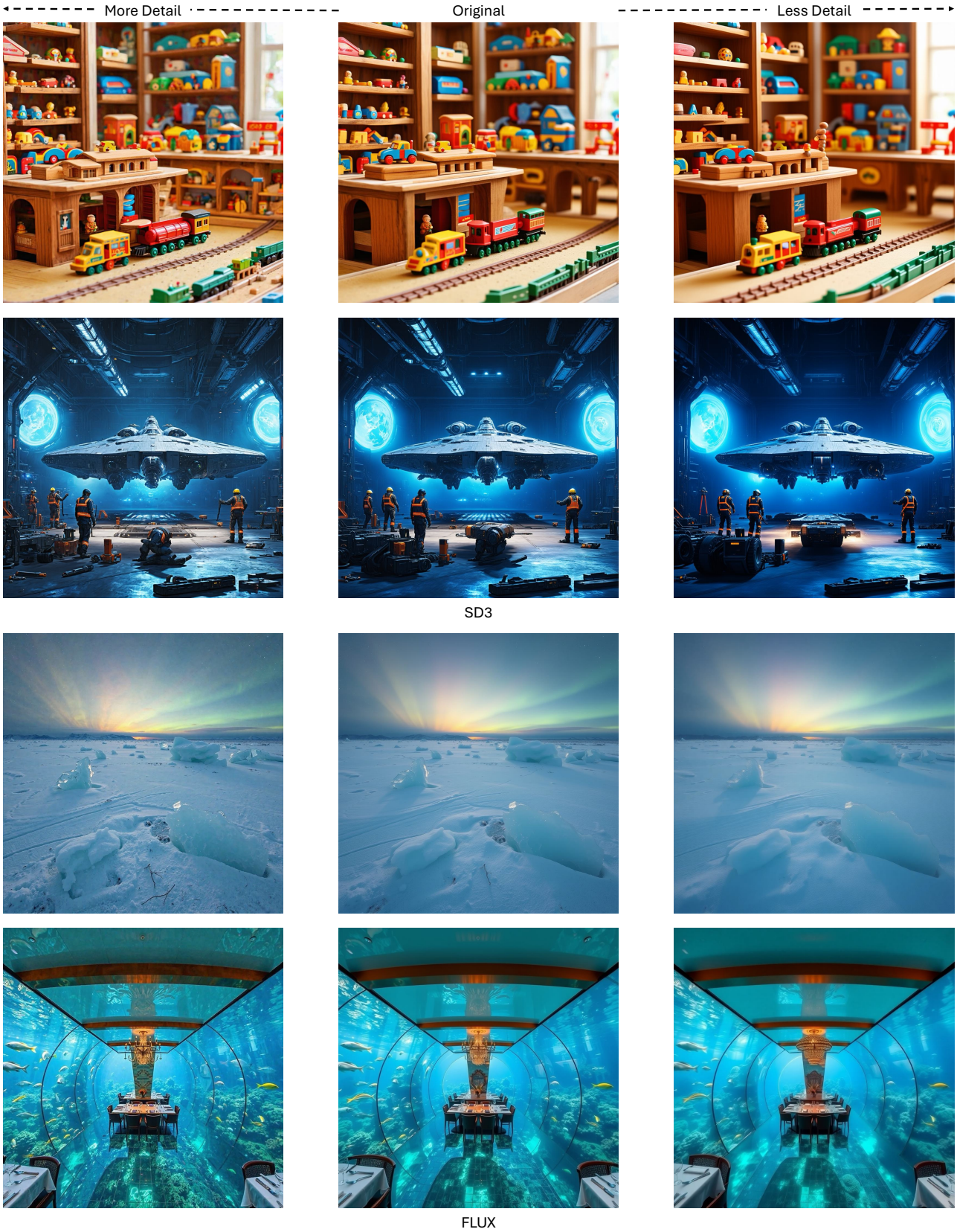


Figure L. More global effects of Omegance in SD3 and FLUX.

Original

EXP1 Schedule

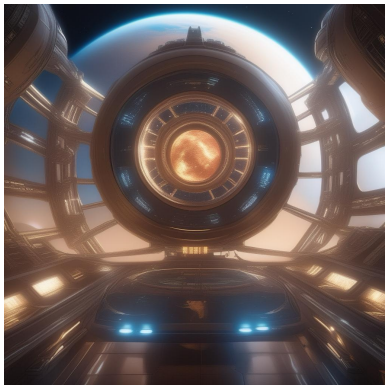
EXP2 Schedule

COS1 Schedule

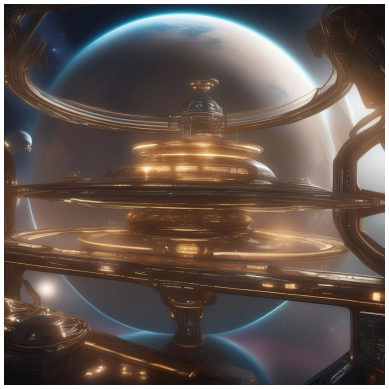
COS2 Schedule

Original

EXP1 Schedule



EXP2 Schedule



COS1 Schedule

COS2 Schedule

Figure M. More temporal effects of schedule-based Omegance follow schedules defined in Fig. 6.

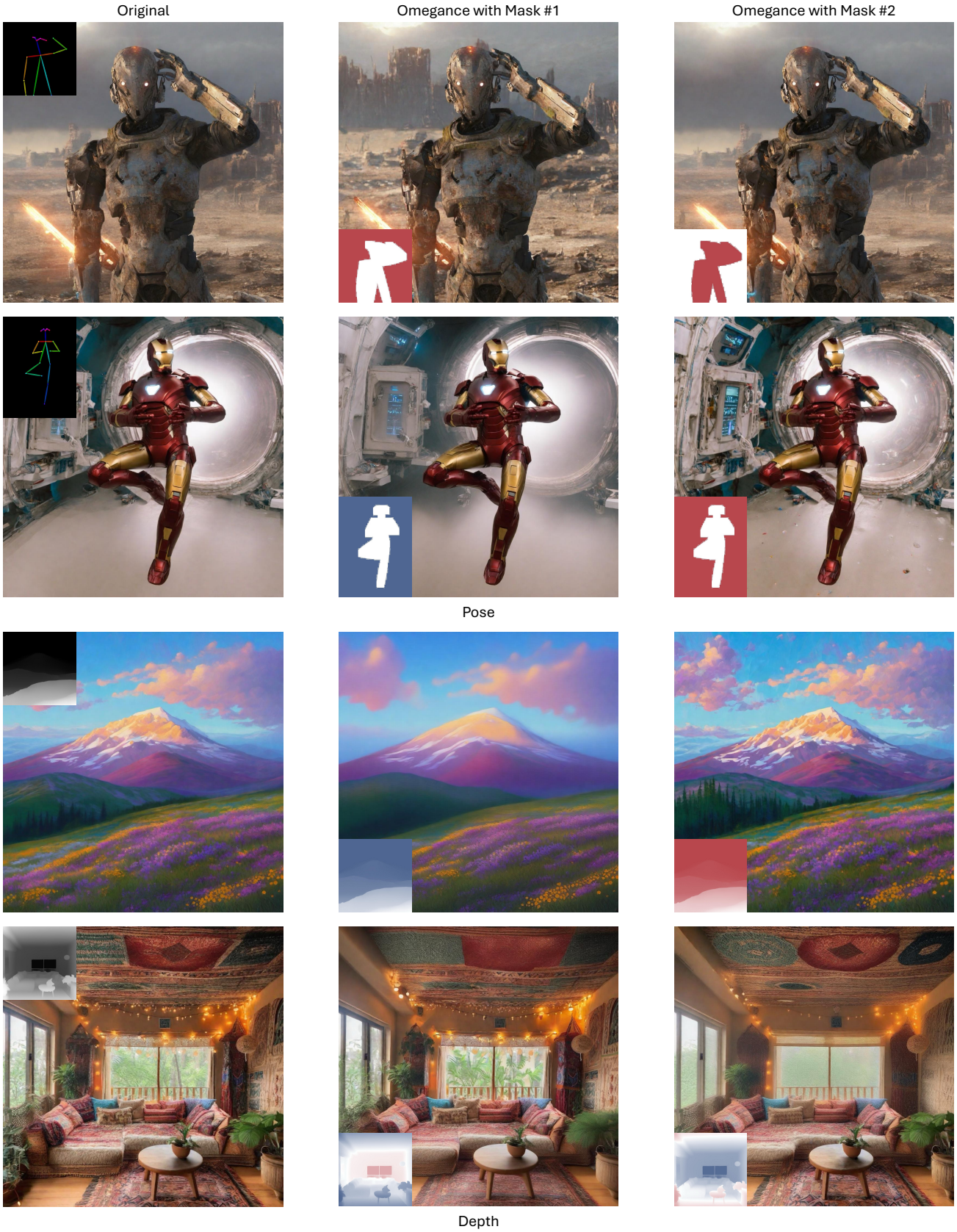


Figure N. More spatial effects of mask-based Omegance.

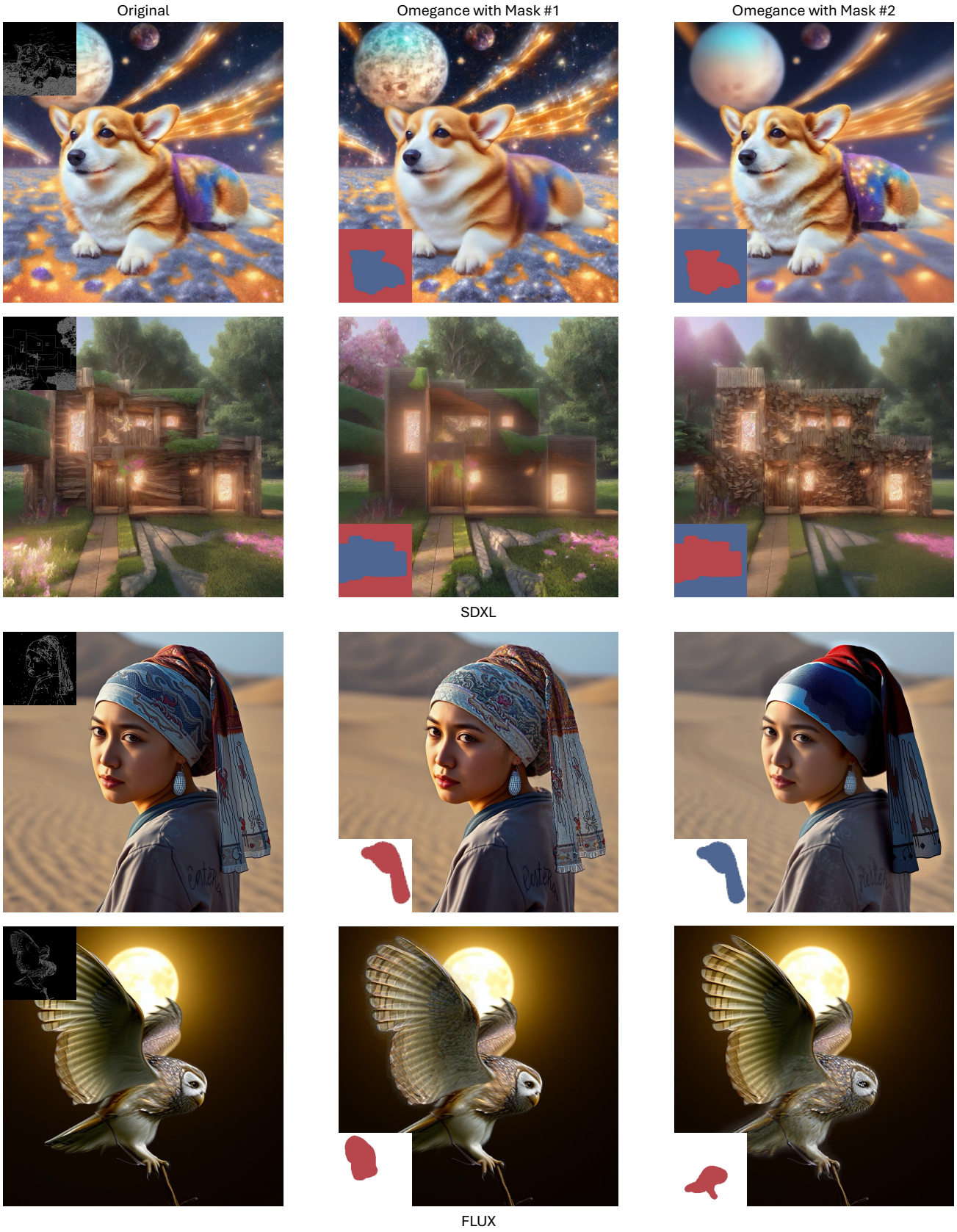


Figure O. More spatial effects of mask-based Omegance using ControlNet with canny signal.

[2] Daniel Garibi, Or Patashnik, Andrey Voynov, Hadar Averbuch-Elor, and Daniel Cohen-Or. ReNoise: Real image inversion through iterative noising. In *ECCV*, 2024. 3

[3] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. AnimateDiff: Animate your personalized text-to-image diffusion models without specific tuning. In *ICLR*, 2024. 3, 4

[4] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022. 1, 6

[5] Black Forest Labs. FLUX. https://github.com/black-forest-labs/flux, 2023. 4, 8

[6] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024. 3, 4

[7] 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. In *ICLR*, 2022. 2, 4

[8] 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. In *ICLR*, 2024. 3, 4, 6, 8

[9] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 2, 3

[10] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2021. 6

[11] SG161222. RealVisXL V5.0. https://civitai.com/models/139562/realvisxl-v50, 2024. 4, 8

[12] Chenyang Si, Ziqi Huang, Yuming Jiang, and Ziwei Liu. FreeU: Free lunch in diffusion U-Net. In *CVPR*, 2024. 8

[13] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 1

[14] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 8