PUMA: Empowering Unified MLLM with Multi-granular Visual Generation

Supplementary Material

7. PUMA Training

7.1. Visual Decoding Training

7.1.1. Dataset Details

For training the image decoding process, we leverage three large-scale datasets: Laion-2B [43], Laion-Aesthetics [5], and JourneyDB [45]. To ensure high-quality generation capabilities, we apply a resolution-based filtering criterion, selecting only images with resolutions of 512×512 pixels or larger. We only use center crop as the data augmentation method.

7.1.2. Training Settings

We train five dedicated image decoders for the f_0, f_1, f_2, f_3, and f_4 scale features respectively. The image encoder is the frozen CLIP-L image encoder [40]. Each image decoder is initialized from the SDXL model. The VAE [21] remains frozen throughout the training process. The corresponding image features are input to the diffusion model through the cross-attention mechanism, replacing the original text embedding input. We train the decoders using AdamW optimizer [34] with a maximum learning rate of 8e-5, using linear learning rate decay and a gradient clipping value of 1.0. The training batch size is $1,024$. The training steps for the five features are $40,000$, $30,000$, $20,000$, $15,000$, and $10,000$ respectively, with features containing more visual tokens using longer training steps. We use noise off value of 0.1 and random drop of 10% of the input image to blank image for classifier-free guidance.

7.2. MLLM Training

7.2.1. Training Objective

PUMA employs a unified framework with supervision on both text tokens and image features. For text tokens, we use cross-entropy classification loss, while for image features, we adopt MSE regression loss. To balance the contribution of text and image outputs, we apply a loss ratio of 0.02 for text and 1.0 for image features. Within the image feature regression loss, we use different ratios for the progressively generated 5 scales of image features (f_4, f_3, f_2, f_1, and f_0), with ratios of $1024.0, 512.0, 64.0, 8.0$, and 1.0 respectively. This scaling compensates for the varying number of tokens at each feature scale, with larger ratios for scales with fewer tokens. The training loss objective remains consistent across both the pretraining and instruction tuning phases.

7.2.2. Pretraining Dataset Details

During PUMA's pretraining phase, we utilize a diverse set of datasets including Laion-2B [43], Laion-Aesthetics [5], GRIT [37], The Pile [14], OCR-VQA-200K [36], and LLaVAR [64]. For the image-text pair data in Laion-2B, Laion-Aesthetics, and GRIT, we randomly assign 50% of the samples to text-to-image training and 50% to image-to-text training, fostering both image generation and understanding capabilities. We employ center crop as the primary image augmentation technique. To train on the GRIT dataset for object grounding, we append 224 additional position tokens to the MLLM's codebook, representing object positions with bounding box coordinates `[x_min, y_min, x_max, y_max]`. We construct the training sequences by appending the tokens `<s>` and `</s>` to denote the beginning and end of each sequence. At the beginning and end of each image feature sequence, we include the special tokens `[IMG]` and `[/IMG]` to indicate the visual position.

7.2.3. Pretraining Settings

We conduct pretraining for 100K steps using the AdamW optimizer with a batch size of 2048. The maximum learning rates are set to 1e-4 for the projector and 3e-5 for the LLaMA backbone. We employ a 2,000-step warm-up period, cosine learning rate decay, and gradient clipping at 5.0 during pretraining. To optimize memory usage and computational efficiency, training is accelerated using DeepSpeed ZeRO Stage 3. The entire pretraining process is carried out on 256 NVIDIA V100 GPUs over a period of 10 days.

7.2.4. Unified Training Settings

We train a single unified model PUMA on a combined dataset encompassing multiple tasks rather than training separate models. Our training strategy includes:

Multi-Task Data Composition: We merge data from high-quality text-to-image generation (Laion-Aesthetics [5] and JourneyDB [45] at 1:1 ratio), precise image manipulation (SEED-Edit [16] covering seven operations), conditional image generation (subset of MultiGen-20M [39] including canny-to-image, inpainting, and colorization), and image understanding tasks from LLaVA-OneVision [24] and Cambrian [51], excluding math/reasoning and cross-duplicated data.

PUMA: We train our unified model with batch size 2048 for $30,000$ steps. We use a max learning rate of 1e-5 with 1000 warm-up steps and cosine learning rate decay. Appropriate image augmentations are applied for each task type within the batch. For image understanding tasks, we apply resizing as the image augmentation and supervise only the text output tokens. This single model handles all tasks including text-to-image generation, image manipulation, conditional generation, and visual chat without requiring separate checkpoints.

8. Evaluation Details

8.1. Image Reconstruction Evaluation

To evaluate the reconstruction performance of different scales of features and our baselines, we use the ImageNet validation set, comprising 50,000 images. Each image is resized to a rectangular shape before being input into each image encoder. We assess reconstruction precision by computing PSNR and LPIPS, which measure the difference between the reconstructed image and the original image.

For PSNR and LPIPS evaluations, we use a resolution of 256×256 to align with the evaluation settings in previous works. For LPIPS evaluation specifically, we employ AlexNet as the feature extractor.

8.2. Text-to-image Generation Evaluation

We evaluate text-to-image generation on the COCO 30K validation set [29]. We use CLIP-I and CLIP-T scores to measure the consistency between the generated image and the ground truth image and caption, respectively. CLIP-Base-32 serves as the feature extractor for these metrics. To assess generation diversity, we calculate LPIPS_d between two images generated using the same input prompt but different random seeds. The LPIPS_d measurement details are consistent with those described in Sec. 8.1.

8.3. Image Editing Evaluation

We evaluate image editing performance on the Emu-Edit benchmark [44]. To assess editing quality, we adopt CLIP-I, CLIP-T, and DINO scores. CLIP-I and DINO [7] scores measure the model's ability to preserve elements from the source image, while CLIP-T reflects the consistency between the output image and the target caption. For the DINO score, we employ DINO-Small-16 as the feature extractor.

8.4. Image Understanding Evaluation

For image understanding tasks, we use the same evaluation setting as LLaVA-v1.5 [30]. During evaluation, we use the system message "`A chat between a curious user and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the user's questions.`"

9. Ablation of Different Scale Features as Input on Image Understanding Task

PUMA employs a unified multi-granular image feature representation as both input and output for the MLLM backbone. To investigate the influence of different scales of image feature input on image understanding tasks, we conducted an ablation study. Specifically, we followed the standard LLaVA-1.5-7B pretraining and fine-tuning protocol, using a CLIP-Large-224 image encoder. For this study, we trained separate models, each using image encodings derived from a single granularity f_4, f_3, f_2, f_1, or f_0, as well as a unified multi-granular model using all scales (f_4-f_0).

Token type	Token num.	MMB↑	MME↑	GQA↑	VQAv2$_{(test)}$↑
f_4	1	56.8	1252.6	0.0	64.1
f_3	4	58.3	1285.5	0.0	67.0
f_2	16	61.5	1403.0	46.6	71.1
f_1	64	63.6	1400.8	58.4	74.4
f_0	256	**65.4**	**1464.9**	<u>58.8</u>	**76.9**
f_4-f_0	341	<u>65.1</u>	<u>1445.5</u>	**61.0**	**76.9**

Table 7. Ablation of different visual token input on image understanding. The experiments are conducted on LLaVA-v1.5 setting with CLIP-Large-224 visual encoder.

Tab. 7 summarizes the results. As expected, finer-grained features generally lead to better performance on image understanding tasks due to their higher spatial resolution and detailed information. For example, the finest scale, f_0, achieves the best performance on most metrics. Notably, utilizing all image features from f_4 to f_0 (the PUMA setting) achieves comparable performance to using all 256 visual tokens of the finest scale (f_0). The limited performance improvement with multi-granularity features can be attributed to their derivation process, as all five feature scales (f_4 to f_0) are generated from f_0 using average pooling. In theory, the f_0 features already encapsulate all the information present in coarser scales, which reduces the potential gains from incorporating multi-granular features. However, adopting multi-granular features as both input and output is essential for unifying the model's design, enabling PUMA to handle diverse visual tasks seamlessly. Additionally, the characteristics of different benchmarks play a crucial role in the observed results. The new benchmarks like MME and MMB, which focus more on local details and fine-grained information, benefit significantly from high-resolution features (f_0). Conversely, tasks in GQA that emphasize global attributes, such as color, texture, or object type, benefit from the diverse representations offered by multi-granular features. This highlights the importance of multi-granular modeling in supporting a broad range of visual tasks while balancing fine-grained and coarse-grained requirements.

10. Visualization of Text-to-image Generation on Five Scale Features

In the text-to-image generation task, if without the use of the router, PUMA produces five distinct images corresponding to the five feature scales, all derived from a single input generation prompt. Fig. 9 presents samples of outputs across these five scales for given generation prompts.

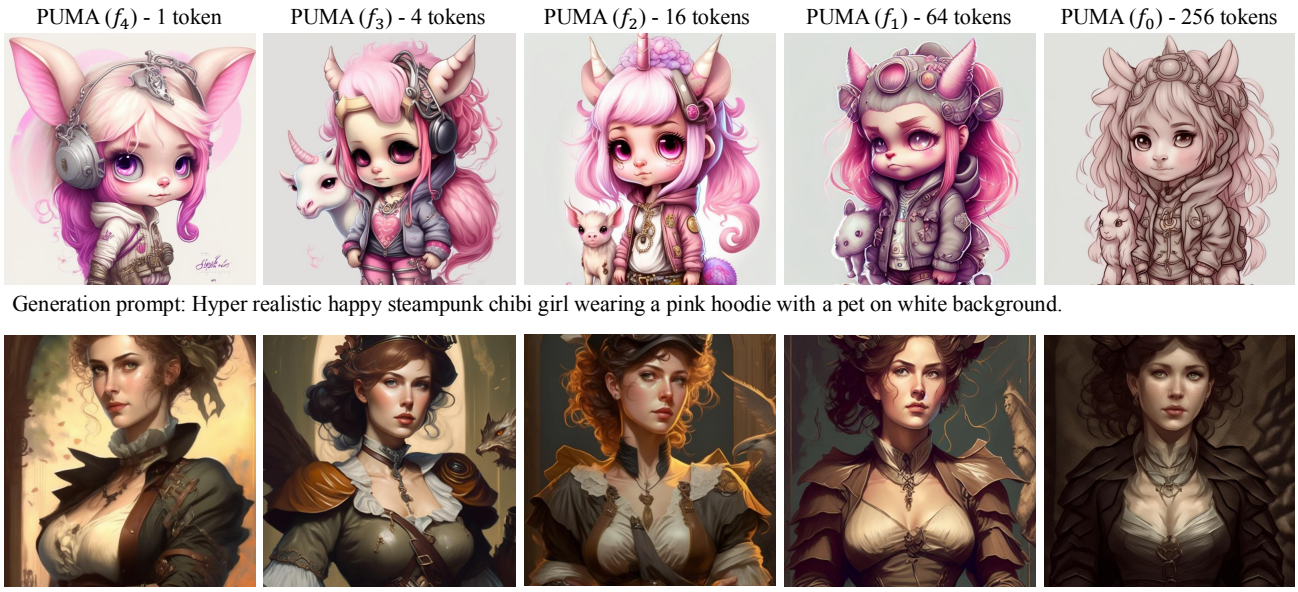

Figure 9. Visualization of PUMA text-to-image outputs across five scale features given the generation prompt.

11. More Qualitative Results

We present more qualitative cases for image reconstruction, diverse text-to-image generation, editing, and conditional image generation, as shown in Figures 10 to 14.

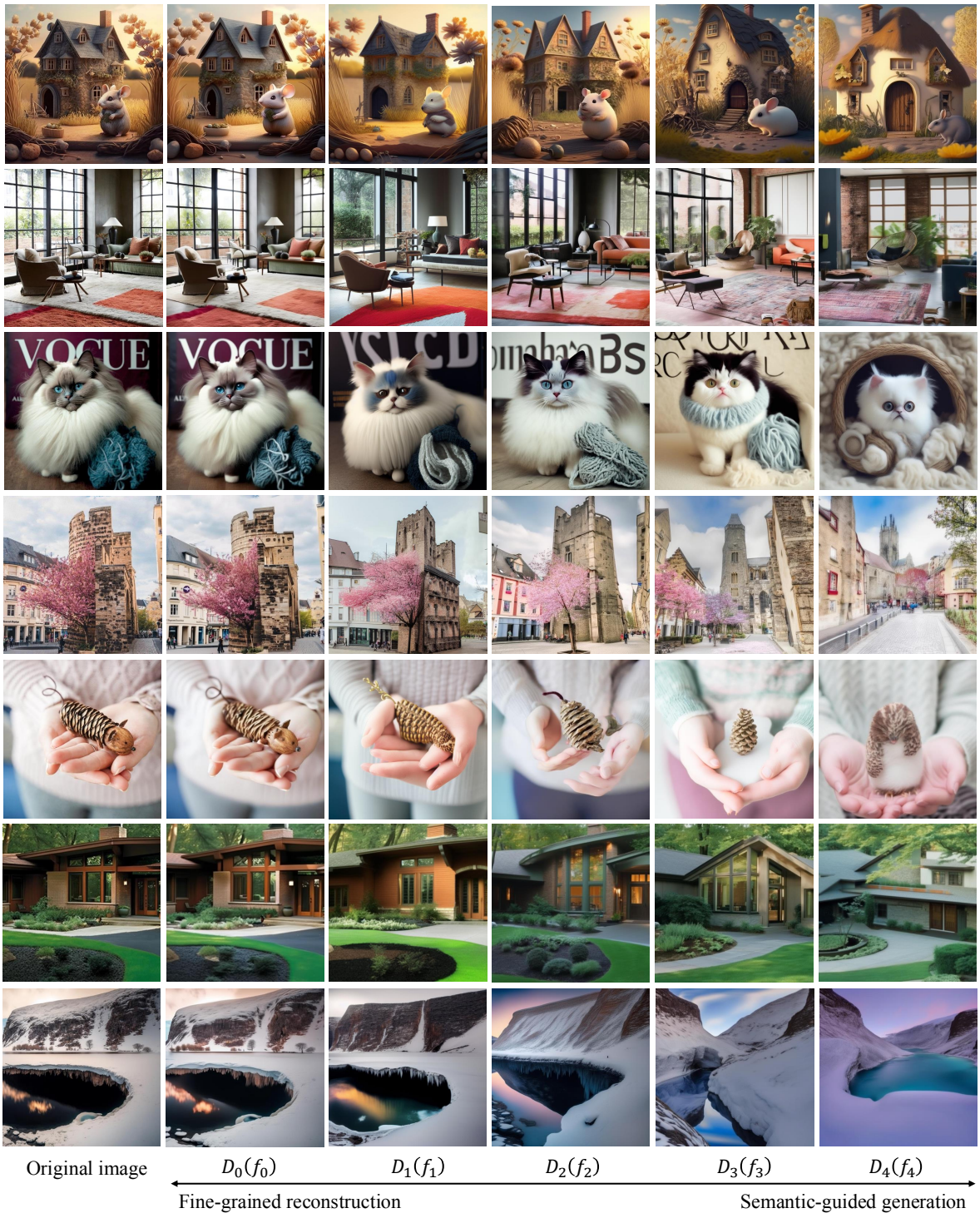

Figure 10. More visualizations on multi-granular visual decoding from fine-grained to coarse-grained granularity.

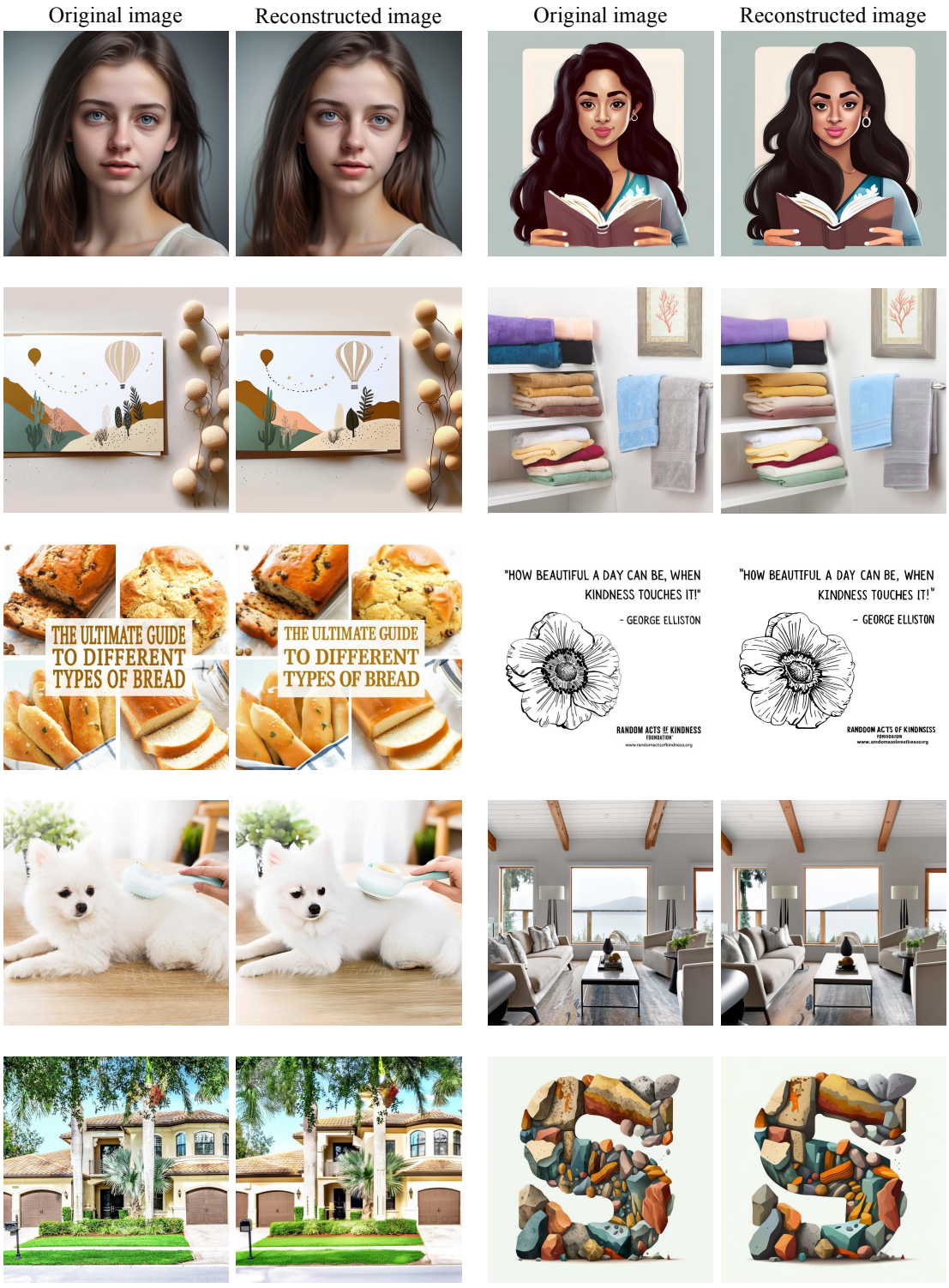

Figure 11. More visualizations on fine-grained image reconstruction with f_0 scale feature.

prompt1: Winter Queen princess baby girl, blond hair, blue eyes, pink lips, Walt Disney beautiful smiling, beautiful character sweet and delicate, stickers, lovely frame, beautiful face, big eyes beautiful.

Prompt2: 1972 porsche, ginza, bright light, hyper realistic, magazine quality, cinematic lighting, neon ads in background, vertical Japanese signs.

prompt3: a cute totoro like tortoise character, bold colors, amiga game, isometric, pixel art, 8K.

prompt4: Film still of rabbit sitting at the counter of an art-deco loungebar, drinking whisky from a tumbler glass, in the style of "Blade Runner", velvety, soft lights, long shot, high quality photo.

prompt5: a container designed compound built for a group home styled living space. 6000 SQ ft with 7 bedrooms and 1 adult suite. give the view landscape style with a smilling pool in the front.

prompt6: Open valley from mountains, aspen, hyper-realistic.

prompt7: Cartoon, pixar style, the planet hamburger, line art drawing, magical scene, highly detailed, soft orange, soft blue, soft pink, soft red, sharp outlines, sharp brush strokes.

prompt8: Beautiful colorful flower motif graphic, in the shape of an elegant flamingo in the style of Hayao Miyazaki, front view.

Figure 12. More visualizations on text-to-image generation utilizing f_4 and f_3 scales.

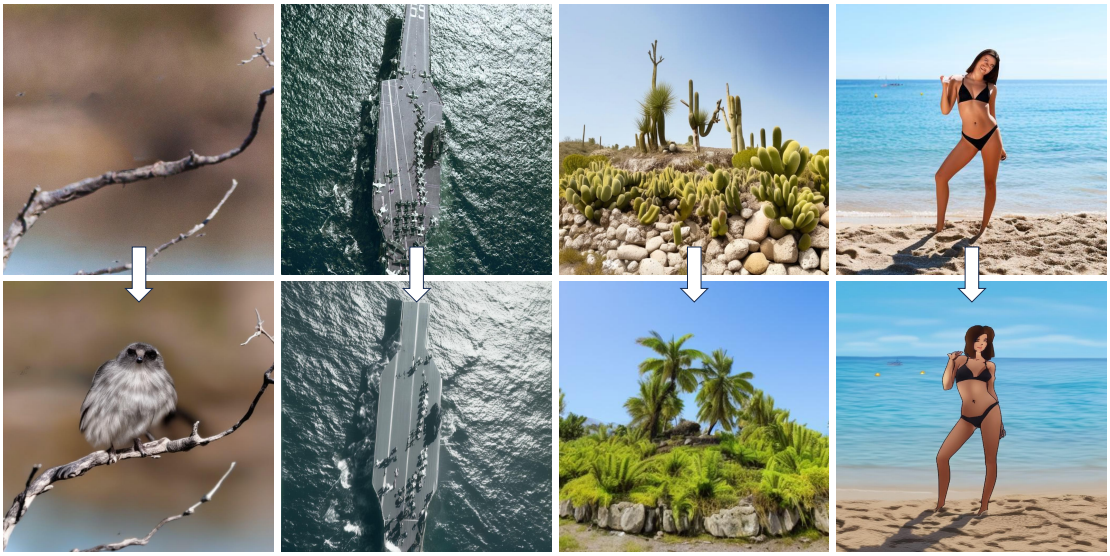

Figure 13. More visualizations on image editing.

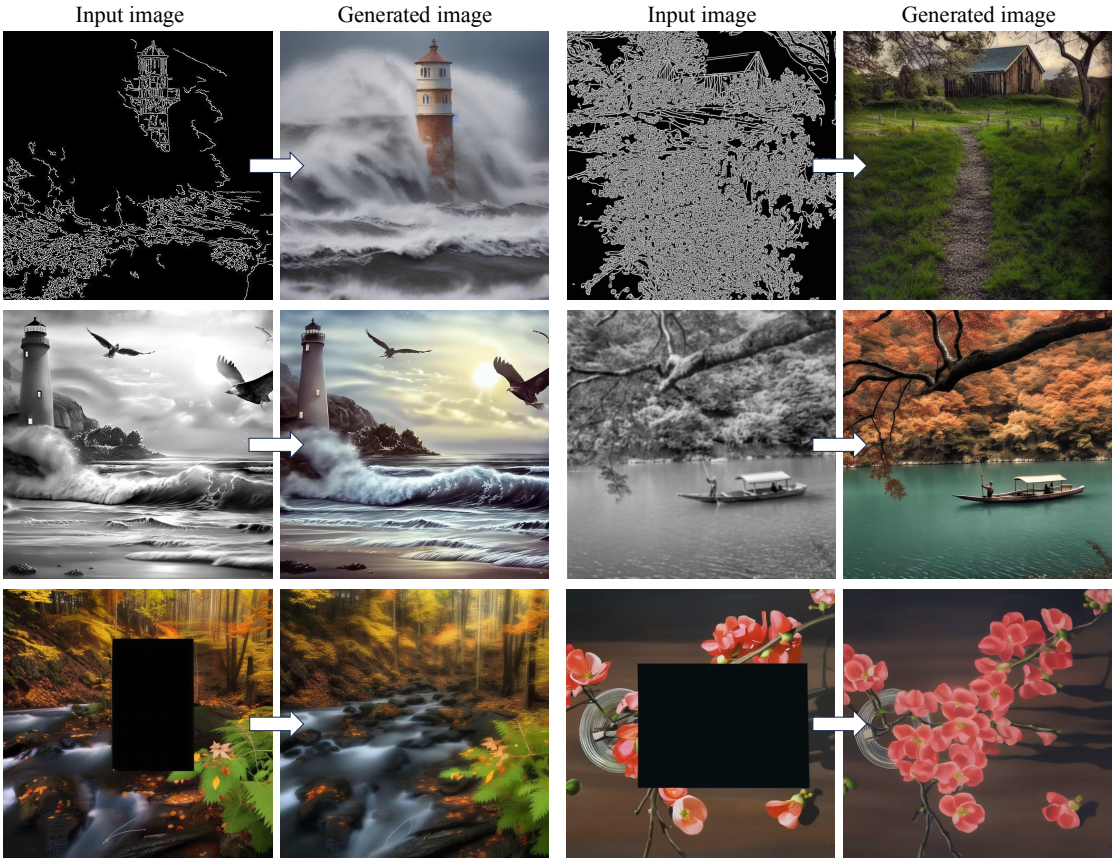

Figure 14. More visualizations on conditional image generation.