Supplemental Material: LLMGA: Multimodal Large Language Model based Generation Assistant

Bin Xia[1], Shiyin Wang[2], Yingfan Tao[2], Yitong Wang[2], and Jiaya Jia[1]

[1] The Chinese University of Hong Kong
[2] ByteDance Inc
https://llmga.github.io/

1 Overview

The overview of the supplementary materials:

(1) We have provided implementation details for the LLMGA (Sec. 2).

(2) We offer a comprehensive introduction to the creation of the training dataset (Sec. 3).

(3) We discussed the differences between BLIP-diffusion and LLMGA (Sec. 4).

(4) We provided the VQA performance of LLMGA (Sec. 5).

(5) We have provided a detailed introduction on how to use LLMGA to implement instruction-based image editing. (Sec. 6)

(6) Additional details of the training and evaluation for LLMGA are elaborated in this part. Furthermore, a detailed training process for Stable Diffusion XL (SDXL) is also included (Sec. 7).

(7) We present more comparison details on LLMGA and LLMGA Embedding, along with corresponding analyses (Sec. 8).

(8) More visual results are showcased to highlight LLMGA's performance in T2I generation (Sec. 9).

(9) More visual results are provided in instruction-based image editing (Sec. 10).

(10) More visual results are presented for LLMGA on inpainting and outpainting (Sec. 11).

(11) More Visual results effectively demonstrate the prowess of DiffRIR in addressing brightness and contrast disparities between newly generated and retained regions during image editing, along with its ability to enhance texture details (Sec. 12).

(12) We provide additional examples showcasing the interactive generation and editing capabilities of LLMGA (with SDXL) (Sec. 13).

2 Implementation Details

For the first stage of training, we employ the pretrained LLaVA-1.5-7B or LLaVA-1.5-13B as the initial MLLM. We utilize the AdamW optimizer, setting the learning rate to 2×10^{-5}. Moreover, we adopt CosineLR as the learning rate scheduler.

The batch size per device and epochs are set to 16 and 1, respectively. Besides, the training ratios for VQA (LLaVA v1.5 mix665k), QA (including general QA, all kinds of design, picture book generation, and illustration generation), prompt refinement, similar image generation, inpainting & outpainting, and instruction-based editing are specified as 1, 0.3, 0.3, 0.3, 0.3 and 0.3, respectively.

For the second stage of training, we adopt the Stable Diffusion 1.5 (SD1.5) as the initial image generation or inpainting & outpainting model. We train these models with the AdamW optimizer, setting the learning rate to 1×10^{-5}. The batch size is set to 32. We train SD1.5 by 1×10^5 iterations.

For the restoration network, we train DiffRIR on DIV2K [1] and Flickr2K [7] datasets with the same GAN-based loss function as DiffIR. The batch sizes are set to 64, and the LQ patch sizes are set to 64×64. We use Adam optimizer, setting the learning rate to 2×10^{-4}. We train this model by 4×10^5 iterations.

3 Data

For the first stage of training, we constructed a training dataset that requires detailed descriptions of images to assist the LLM in better understanding the compositional details of images and supporting image generation and editing. Specifically, we utilized the MSCOCO [5] datasets, encompassing rich real-world scenarios. Then, we employed GPT4-V to generate detailed and visually compelling descriptions for these datasets. The prompt format for GPT4-V to generate a detailed description is as follows: *"The original caption of this image is 'ORIGINAL CAPTION'. Describe this image."*, where *'ORIGINAL CAPTION'* is a short caption provided by the dataset. An example is shown in Fig. 1.

In order to better adapt the LLMGA to instruction-based image editing, we finetuned Mixtral7Bx8 using short instruction editing training data from InstructPix2Pix. This enabled it to learn how to generate corresponding editing data and target descriptions in JSON format based on detailed descriptions. Then, we used the finetuned Mixtral7Bx8 to create corresponding editing data for detailed descriptions of MSCOCO images made with GPT4-V. An example is shown in Fig. 2. After that, we filtered out data with incorrect output formats to obtain the final instruction-based editing training dataset.

Furthermore, we have gathered a vast amount of design books and graphic story datasets for creating parts QA data, which has been used to train the LLMGA. This enables the LLMGA to excel in various design tasks, including logo design, fashion design, infographic creation, and poster design. Additionally, it can generate illustrated storybooks that intertwine text and images, tailored to user specifications.

4 Discuss BLIP-diffusion and LLMGA.

(1) Unlike works as BLIP-diffusion [4], which uses MLLMs primarily as image and text encoders, our LLMGA aims to retain the multi-turn conversation and general cognitive ability of LLMs, acting as a designer and assistant to produce

The original caption of this image is "A tall clock tower sitting under sky." Describe this image.

The image depicts a tall clock tower with a large clock face prominently displayed on its side. The clock face is white, and the tower itself is made of stone. The tower stands against a grey sky, which adds a sense of depth and contrast to the scene. The clock is situated near the top of the tower, making it easily visible from a distance.

Fig. 1: An example of generating detailed descriptions for description data.

The image depicts a tall clock tower with a large clock face prominently displayed on its side. The clock face is white, and the tower itself is made of stone. The tower stands against a grey sky, which adds a sense of depth and contrast to the scene. The clock is situated near the top of the tower, making it easily visible from a distance. Please help me create instructions editing data based on the above input and output it in JSON format.

{"instruction": "change the clock to a digital display", "new_caption": "The image depicts a tall clock tower with a large digital clock display prominently displayed on its side. The digital clock display is white, and the tower itself is made of stone. The tower stands against a grey sky, which adds a sense of depth and contrast to the scene. The digital clock is situated near the top of the tower, making it easily visible from a distance}

Fig. 2: An example of generating detailed descriptions for instruction-based editing data.

satisfactory images through conversation. **(2)** Subject-driven generation can be implemented with IP-Adapters, easily integrated into our SD-ft. Fig. 3 shows our LLMGA achieves better outcomes. **(3)** We will add the discussion of these methods in our paper.

5 VQA Evaluation.

In Tab. 1, compared to other MLLMs with image generation ability, our LLMGA has a much better performance on both VQA and image generation.

6 More Details on Instruction-based Editing

Given an image and corresponding editing instructions, our LLMGA provides a description of the image and a target description after editing based on the given editing instructions. We then perform a Direct Inversion [3] on the image

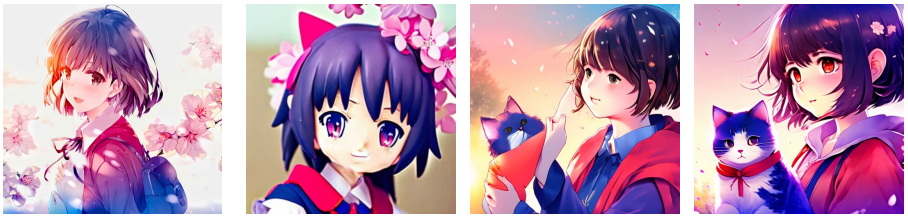


Fig. 3: The comparison on subject-driven generation.

Table 1: VQA comparison.

Method	Can image generation?	MMB↑	VQA^{V2}↑	POPE↑
LLaVA1.5-7b	✗	64.3	78.5	85.9
GILL	✓	4.8	20.4	50.1
LLMGA-embedding	✓	35.3	56.3	67.4
LLMGA-7b (Ours)	✓	**62.9**	**77.2**	**83.8**

to obtain an initial noise map. Utilizing this noise map, along with the original image's description and the edited target description, we employ the straightforward prompt-to-prompt [2] method to achieve the desired edited outcome.

7 More Training and Evaluation Details

In addition to the LLMGA (SD1.5) described in the paper, we have also conducted training for LLMGA (SDXL). LLMGA (SDXL) follows a training process and configuration similar to that of LLMGA (SD1.5) but demonstrates even higher-quality generation. Specifically, for the first training stage, we employ the same MLLM as utilized in LLMGA (SD1.5). Regarding MLLM, we utilize the pretrained LLaVA-1.5-7B or LLaVA-1.5-13B as the initial MLLM. Our optimization approach involves the use of the AdamW optimizer, with the learning rate set at 2×10^{-5} and weight decay at 0, respectively. Additionally, we adopt the CosineLR learning rate scheduler. The total batch size and number of epochs are configured at 128 and 1, respectively. MLLM training is carried out on our constructed datasets as described in the paper. Furthermore, the training ratios for VQA (LLaVA v1.5 mix665k), QA (Alpaca), prompt refinement, similar image generation, inpainting & outpainting, and instruction-based editing are specified as 1, 0.3, 0.3, 0.3, 0.3 and 0.3, respectively.

For the second stage of training, we adopt the Stable Diffusion XL (SDXL) [6] as the initial image generation or editing model. We train these models with AdamW optimizer, setting the learning rate to 1×10^{-5}. The total batch size is

set to 32. We train SDXL by 5×10^4 iterations. For both T2I generation and inpainting & outpainting, we conduct training on the LAION-Aesthetic dataset, using the generation prompts generated by the first-stage pretrained MLLM as guidance. The input patch sizes are set to 1024×1024. Moreover, similar to SD1.5, we randomly generate masks for SDXL inpainting & outpainting, including box masks, irregular masks, and boundary masks.

8 More Details on Control Scheme

The form in which to establish a control link between MLLM and SD is a question that requires careful consideration. In this paper, as illustrated in Fig. 4, we explore two control schemes: namely, our adopted language-based generation prompt control (Fig.4 (a)) and visual embedding-based control scheme (Fig.4 (b)). Here, we will provide a detailed overview of the training approach for LLMGA Embedding, followed by a comparison and analysis of these schemes.

For the training of LLMGA Embedding, we employ a comprehensive three-stage training scheme. **(1)** In the first stage, for T_R, we use the same auto-regressive cross-entropy loss ($\mathcal{L}_{MLLM}$, Eq. 1) as LLMGA, and for visual embedding, we utilize the visual embedding loss ($\mathcal{L}_{embed}$, Eq. 2). We simultaneously apply the same training settings as LLMGA. **(2)** In the second stage, we initiate joint optimization of MLLM and SD. Unlike LLMGA, in this phase, we optimize only the projection layer for LLMGA embedding, using the SD loss ($\mathcal{L}_{SD}$, Eq. 3), and then freeze the parameters of Unet and MLLM. **(3)** In the third training stage, we freeze the parameters of MLLM and projection, and then optimize the SD Unet using the SD loss ($\mathcal{L}_{SD}$, Eq. 3).

$$\mathcal{L}_{MLLM} = \mathbf{CE}\left(\mathbf{T}_R, \mathbf{T}_{GT}\right), \tag{1}$$

$$\mathcal{L}_{embed} = \|\phi_{proj}(\mathbf{V}_E) - \phi_{CLIP}(\mathbf{T}_{caption})\|_2^2, \tag{2}$$

$$\mathcal{L}_{SD} = \mathbb{E}_{\mathbf{Z}_t, \mathbf{C}, \epsilon, t}\left(\|\epsilon - \epsilon_\theta\left(\mathbf{Z}_t, \mathbf{C}\right)\|_2^2\right), \tag{3}$$

where $\mathbf{T}_R$ denotes the generated text response, and $\mathbf{T}_{GT}$ represents the ground-truth target. $\mathbf{CE}(.)$ signifies auto-regressive cross-entropy. $\phi_{proj}(.)$ denotes the projection involving three linear layers. $\phi_{CLIP}(.)$ is the CLIP text encoder. $\mathbf{V}_E$ stands for visual embedding, and $\mathbf{T}caption$ corresponds to the original caption from the datasets. $\mathcal{L}_{SD}$ represents the diffusion loss as described in the paper.

The results are presented in Fig. 6 in the main paper. Despite comprehensive training, we find that the embedding-based approaches still lags behind LLMGA. Moreover, with an increase in the number of dialogue turns, there is a noticeable decrease for embedding-based approaches in both the accuracy and quality of generation. This observation can be well comprehended through the processing mechanism of LLM. Specifically, LLM predicts an embedding based on previous input images and texts (Eq. 4). After that, embedding is refined through a linear layer, categorizing it into a fixed language domain (Eq. 5). This is because

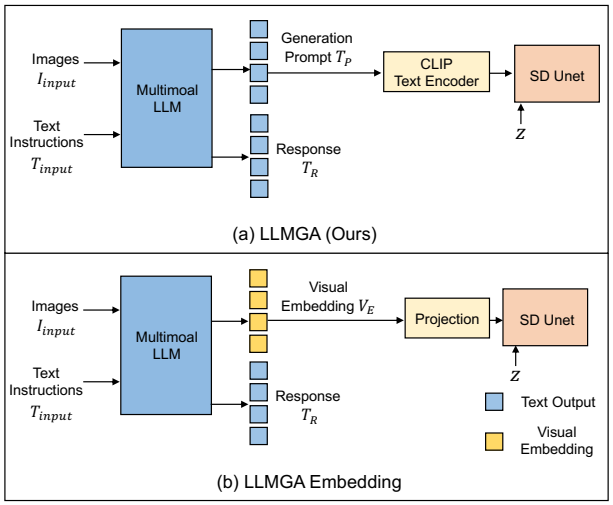


Fig. 4: The illustration of LLMGA and LLMGA Embedding. LLMGA uses language generation prompts as a control scheme, while LLMGA Embedding uses visual embedding as a control scheme.

predicted embeddings exist in a continuous space, inherently imprecise and filled with noise. For instance, the same embedding, depending on the sampling probabilities, may generate different semantic words, indicating that embeddings are filled with various forms of noise. Mapping embeddings to a fixed and specific language domain by classification effectively eliminates this noise, enabling precise control over SD generation. The decline in performance of embedding-based methods with an increase in conversation turns is attributed to the introduction of additional noise into the predicted embedding as more prior conversation information is incorporated, thereby affecting accuracy.

$$\mathbf{E} = \Psi_{body}(\mathbf{I}_{input}, \mathbf{T}_{input}), \tag{4}$$

$$\mathbf{T} = \Psi_{linear}(\mathbf{E}), \tag{5}$$

where $\mathbf{I}_{input}$ and $\mathbf{T}_{input}$ represent all preceding input images and texts. Ψ_{body} signifies the network body of LLM, producing an embedding $\mathbf{E}$. Ψ_{linear} constitutes the final linear layer in LLM, tasked with categorizing the embedding into a predetermined text domain, resulting in the generation of text $\mathbf{T}$.

In summary, compared with embedding based methods, our LLMGA using detailed language prompts for control generation has the following advantages:

- The embeddings predicted by the MLLM are often filled with noise. This can be filtered out by mapping them to a fixed language domain, enabling precise control of SD.
- Detailed language prompts can make the network more transparent and interactive, allowing users to understand MLLM's thoughts for generating images.

- MLLM is pre-trained on vast textual datasets. Explicit language prompts rather than implicit embeddings are more advantageous for MLLM to generate prompts and comprehend context.
- Dynamic-sized language prompt facilitates the addition of generation requests during interactions.
- Training is more simple and more efficient.

9 More Results on T2I Generation

The more T2I generation visual results are shown in Fig. 5. LLMGA can leverage its vast comprehension, reasoning abilities, and knowledge reservoir to generate visuals with more details. Furthermore, LLMGA refine prompts based on user requirements.

10 More Results on Instruction-based Editing

The more instruction-based editing visual results are depicted in Fig. 6. Our LLMGA can realize more accurate and visually pleasing editing according to the requirements of users.

11 More Results on Inpainting and Outpainting

The more inpainting and outpainting visual results are depicted in Fig. 7. LLMGA can harness its extensive comprehension, reasoning abilities, and knowledge reservoir to infer plausible complete images based on given masked images. Additionally, as demonstrated in paper Fig. 1, LLMGA can edit images in accordance with user specifications and masked images.

12 More Results on Image Restoration

The more image restoration results are shown in Fig. 8. Our DiffRIR can alleviate the texture, brightness, and contrast discrepancies, and generate more realistic details.

13 More Results on Interactive Generation and Editing

The results are shown in Figs. 9, 10, and 11. The results presented here were obtained using LLMGA-7b (SDXL-ft). It can be observed that LLMGA, leveraging the understanding, reasoning abilities, and extensive knowledge repository of LLM, effectively assists users in image design in an interactive manner.

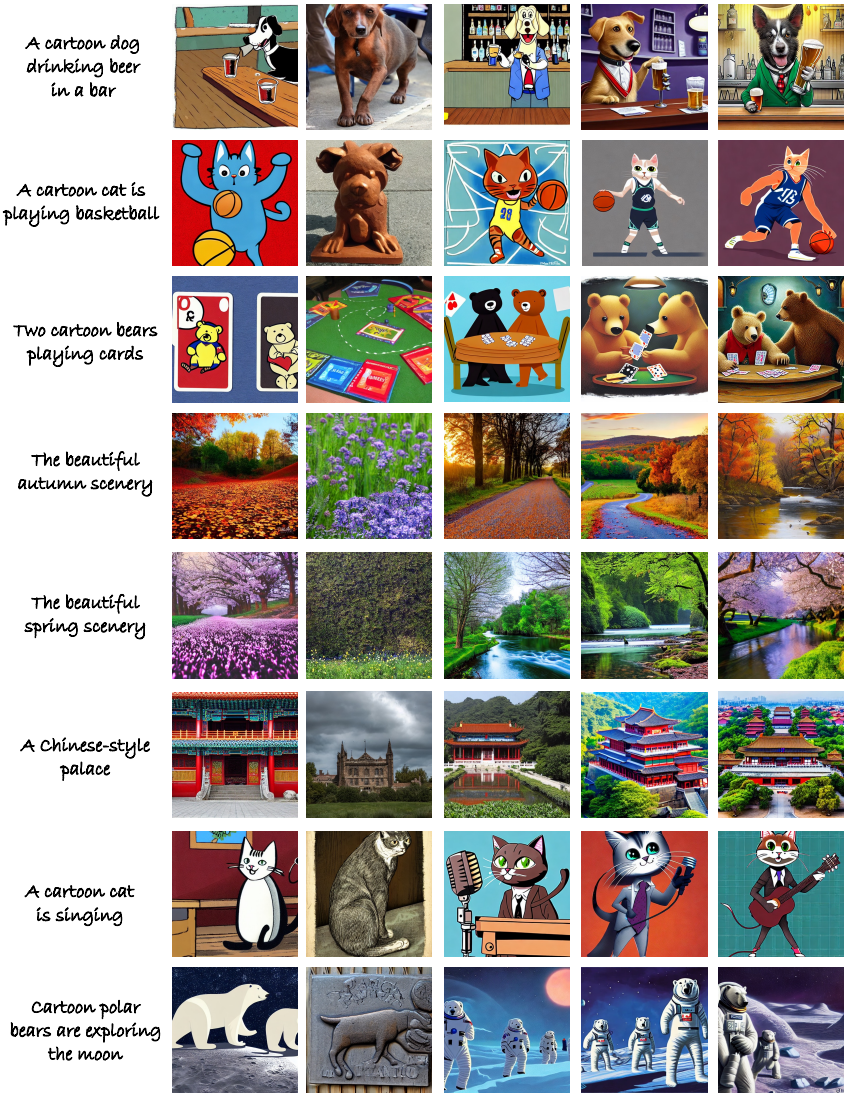


Fig. 5: Visual comparison on **T2I**. LLMGA can refine short prompts by adding details, such as clothing, background, and actions. In addition, for unfamiliar concepts, LLMGA can utilize its extensive knowledge base from LLM to realize accurate generation.

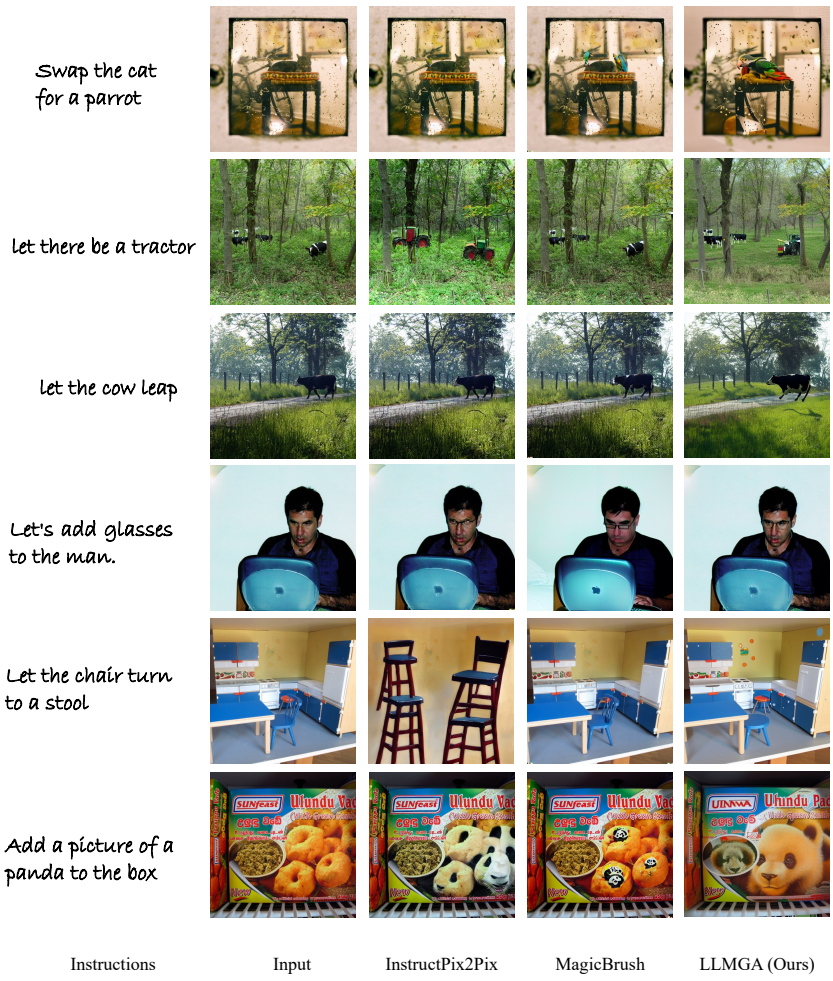


Fig. 6: Visual comparison on **instruction-based editing**.

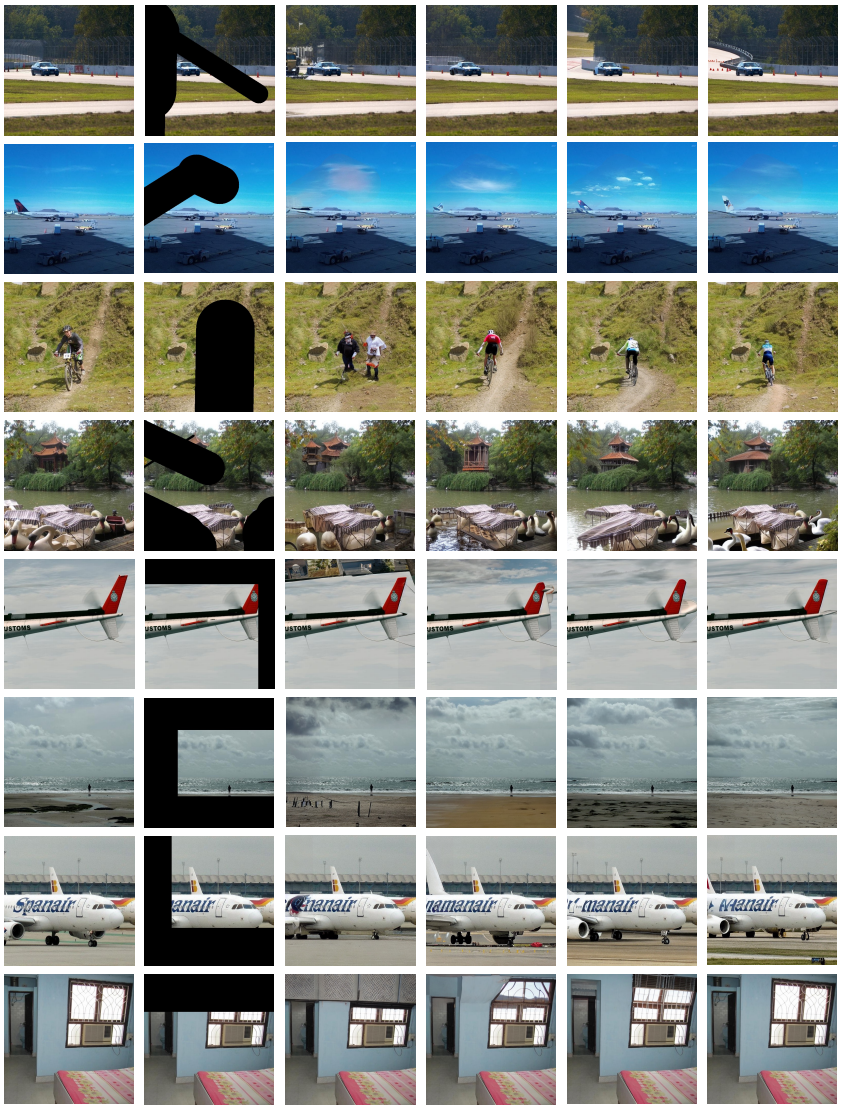


Fig. 7: Visual comparison on **inpainting and outpainting**. LLMGA can infer complete images based on input masked images.

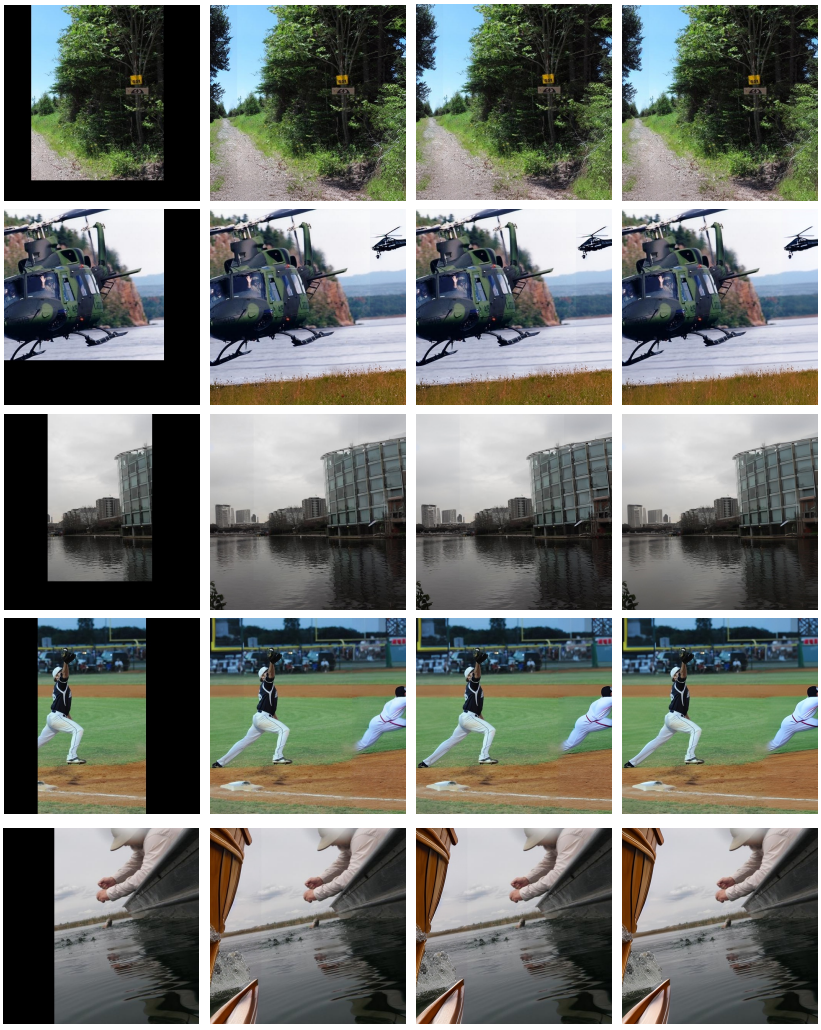
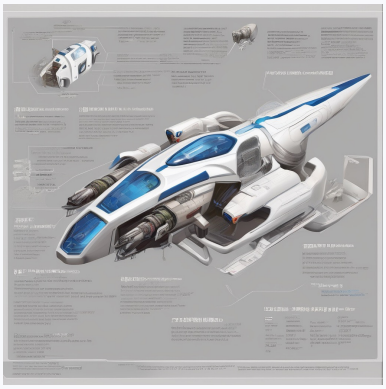


Fig. 8: More visual comparison on image **restoration**.

Please help me design a spaceship.

The image has been successfully generated.

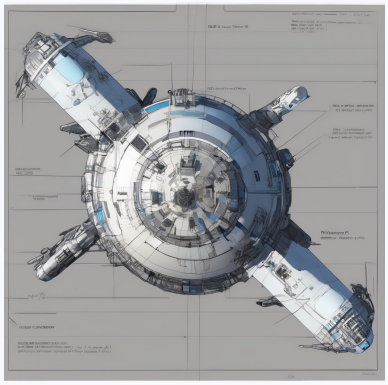


<gen_image> The image is a detailed illustration of a futuristic spacecraft, showcasing its design blueprint. The spacecraft is large and features a prominent white and blue color scheme. It is equipped with multiple engines, including a large one on the back and smaller ones on the sides.

In addition to the engines, the spacecraft is armed with several weapons, including a group of missiles on the top and a laser cannon on the front. The overall design of the spacecraft appears sleek and advanced, reflecting the concept art's purpose of demonstrating a potential future spacecraft design. </gen_image>

I prefer a disc-shaped spaceship. Can you modify the design proposal and generate a new blueprint?

Sure, here is a modified blueprint for a disc-shaped spaceship:



<gen_image> The image features a large, disc-shaped spacecraft with a prominent white and blue color scheme. It is equipped with multiple engines, including a large one on the back and smaller ones on the sides. The spacecraft is armed with several weapons, including a group of missiles on the top and a laser cannon on the front.

The overall design of the spacecraft appears sleek and advanced, reflecting the concept art's purpose of demonstrating a potential future spacecraft design. </gen_image>

Fig. 9: Examples of LLMGA on interactive generation and editing.

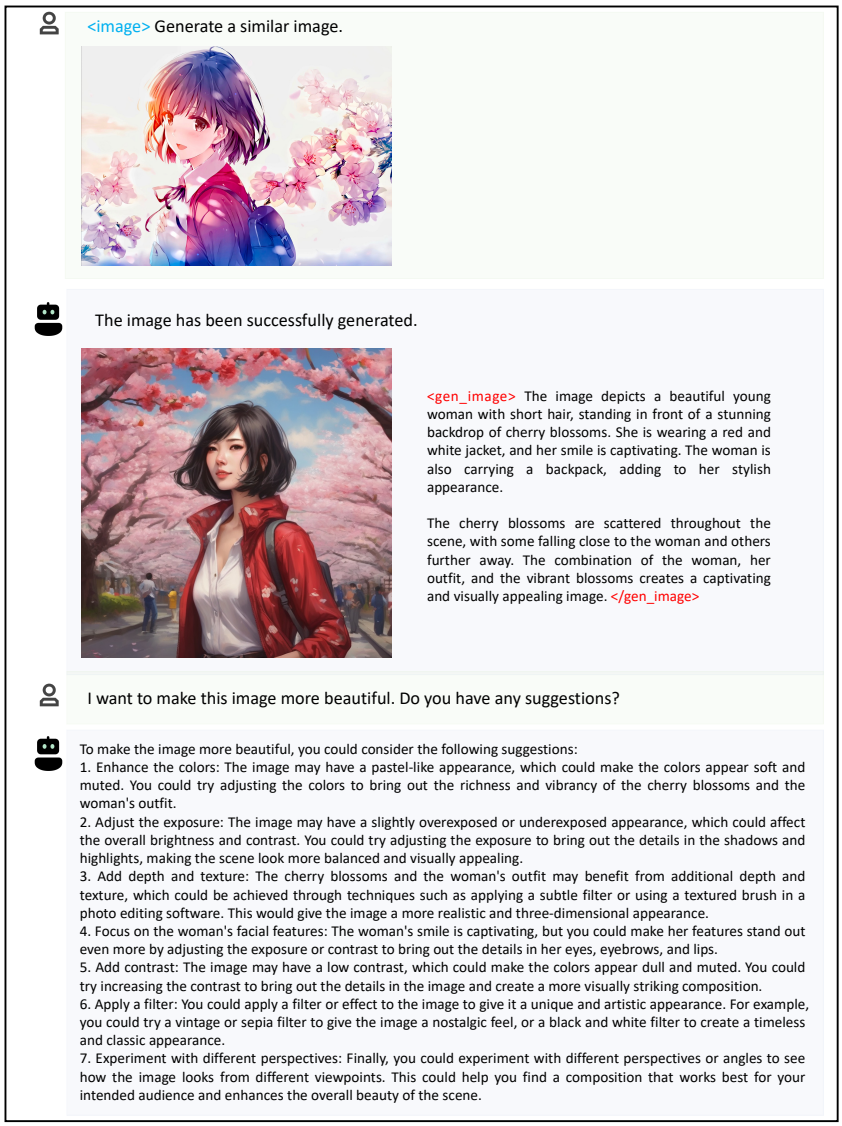


Fig. 10: Examples of LLMGA on interactive generation and editing.

Fig. 11: Examples of LLMGA on interactive generation and editing.

References

1. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: CVPRW (2017)
2. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626 (2022)
3. Ju, X., Zeng, A., Bian, Y., Liu, S., Xu, Q.: Direct inversion: Boosting diffusion-based editing with 3 lines of code. ICLR (2024)
4. Li, D., Li, J., Hoi, S.: Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. NeurIPS (2024)
5. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
6. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. NeurIPS (2024)
7. Timofte, R., Agustsson, E., Van Gool, L., Yang, M.H., Zhang, L.: Ntire 2017 challenge on single image super-resolution: Methods and results. In: CVPRW (2017)