InstructAny2Pix: Flexible Visual Editing via Multimodal Instruction Following

Shufan Li, Harkanwar Singh, Aditya Grover
{jacklishufan,harkanwarsing,adityag}@cs.ucla.edu
University of California, Los Angeles

Abstract

The ability to provide fine-grained control for generating and editing visual imagery has profound implications for computer vision and its applications. Previous works have explored extending controllability in two directions: instruction tuning with text-based prompts and multi-modal conditioning. However, these works make one or more unnatural assumptions on the number and/or type of modality inputs used to express controllability. We propose InstructAny2Pix, a flexible multi-modal instruction-following system that enables users to edit an input image using instructions involving audio, images, and text. InstructAny2Pix consists of three building blocks that facilitate this capability: a multi-modal encoder that encodes different modalities such as images and audio into a unified latent space, a diffusion model that learns to decode representations in this latent space into images, and a multi-modal LLM that can understand instructions involving multiple images and audio pieces and generate a conditional embedding of the desired output, which can be used by the diffusion decoder. Additionally, to facilitate training efficiency and improve generation quality, we include an additional refinement prior module that enhances the visual quality of LLM outputs. These designs are critical to the performance of our system. We demonstrate that our system can perform a series of novel instruction-guided editing tasks.

1. Introduction

With growing fidelity of generative models of images [3, 20], the ability to control their outputs is critical for many real-world applications, ranging from creative generation to synthetic augmentations. However, popular state-of-the-art tools such as ControlNet[32] and T2I[19] can only perform specific edits they are trained on. Many of these models also require inputs such as Canny edge, Depth, or Surface Normal, making them inaccessible to general users without expertise in computer vision. To mitigate this, instruction-based image editing methods such as InstructPix2Pix[4] allow users to describe their instructions in natural languages, such as "add a dog." But, such methods are still limited to simple instruction on which they are trained and cannot generalize to complex instructions involving multiple editing operations or multiple objects. Additionally, they cannot take additional audiovisual inputs, making them struggle with tasks such as style transfer.

We propose InstructAny2Pix, the first instruction-following image editing system that can follow complicated, multi-modal, multi-object instructions. Concrete examples of such instructions can be "add the [sound] to [image]," where the sound can be that of a dog barking or a piece of music. It can also be "add [object A] and remove [object B] from [image]," where objects can be represented by either images, text, or audio. Additionally, it can also include free-form instructions like "change [image A] to the style of [image B]" or "fit [image] to [music]." Our work represents a significant expansion in the scope of image editing instructions. Through both quantitative and qualitative evaluations, we show that our proposed method achieves high performance on diverse editing tasks.

Our framework brings together the multi-modal perception capability of a multi-modal encoder, high quality generation capability of a diffusion model and instruction understanding capability of a LLM. The performance of our system depends on three key components:

First, we train our model on large amounts of multi-modal edit instructions. Our training data consists of diverse instructions, including adding audio to an image, adding an image to an image, adding multiple objects, or replacing an object with another object where objects are represented by either images, text, or audio. We generate this dataset by prompting state-of-the-art Large Language Models with human-written examples for each category of instructions. This diverse dataset is crucial for extending instruction-following capabilities of image-editing systems to long, complex, and multi-modal instructions.

Second, we integrate recent techniques in instruction tuning of language models and multi-modal representation learning to efficiently understand instructions involving inputs of multiple modalities. In particular, we encode the multi-modal inputs using the encoder and insert them into

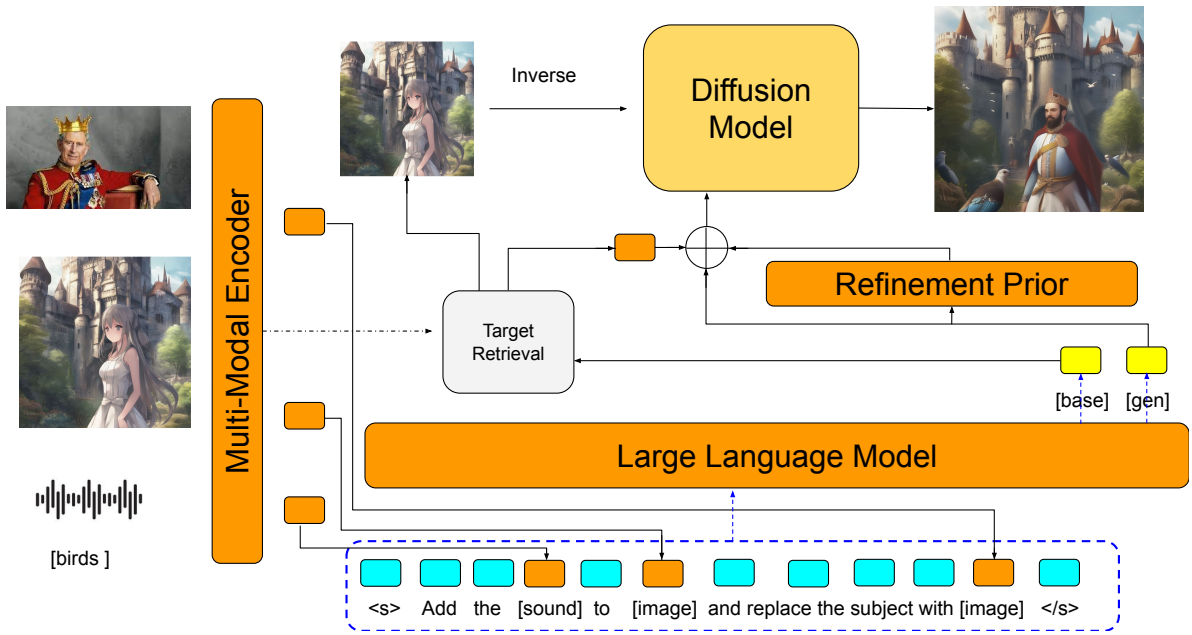

Figure 1. The InstructAny2Pix pipeline consists of three building blocks: a multi-modal encoder that encodes different modalities into a unified latent space, an LLM that can encode instructions (encoding [gen]), and a diffusion model that learns to decode latent representations into edited images. For improved training and generation, we include an additional prior module to refine the LLM outputs.

the instruction sequence that is passed to an LLM. The LLM then learns to understand the instruction and generate an output embedding. This embedding is then used to condition a diffusion model that generates the image.

Last, we propose a refinement prior module to facilitate efficient training and enhance generation quality. The need for such a refinement module arises from two key observations. First, large language models are slow to train, and it may take considerable time for them to learn to generate high-quality representations that are useful for generation tasks. Second, the quality of data typically used for multi-modal alignment is worse than that used to train high-quality generative models. For example, the average aesthetic score for VGG-Sound [5], a dataset commonly used for audiovisual alignment, is only 4.5 because it mostly consists of low-quality YouTube clips. In contrast, LAION-600M [24], which is typically used to train diffusion models, has a minimum score of 5. These observations make it necessary to introduce a method to mitigate unwanted biases and accelerate training. We implement our refinement prior module as a transformer that learns to refine the output embedding of the LLM.

2. Related Works

There are a large number of image-editing methods based on text-to-image diffusion models [6, 12, 20, 23]. They can be generally categorized into three families.

The first family is based on Image2Image translation. Works like ControlNet [32] and T2I [19] added convolution adaptors to the U-Net of Diffusion Models. These models enable image generation conditioned on texts and additional inputs such as canny edge, and depth. They achieve image editing by first translating the image to the desired conditional domain and generating a new image from this condition using desired text prompts. However, such methods lose considerable information from the source image and struggle to preserve details due to the translation mechanism. These methods also have poor accessibility due to the amount of expertise required to generate these conditions.

The second family is based on text prompts. These methods only require prompts describing the desired output, making them more accessible. The naive approach is to perform DDIM[25] inversion that converts an image back into the latent space of a diffusion model and generates a new image from such a latent representation. Prompt2Prompt (P2P) [10] improves this baseline by injecting the cross-attention maps during the diffusion process. Plug-and-Play [29] additionally injects convolution features to provide more refined control. Parallel to these lines of work, Null-text Inversion [18] proposed a better inversion method for input images through a learnable null text prompts. While these methods are easier to use than the previous family, they still require detailed descriptions of each desired output. This limits their usability in applications like batched editing.

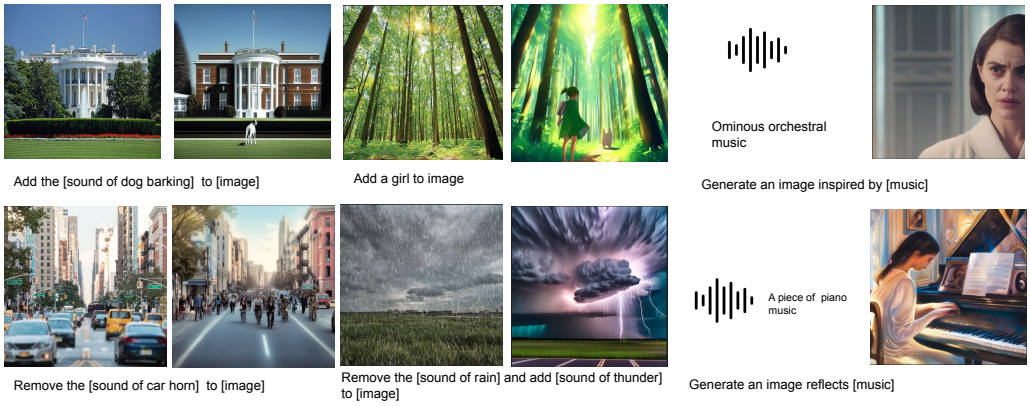

Figure 2. Illustration of InstructAny2Pix's ability to flexibly edit an image based on a variety of different multi-modal instructions.

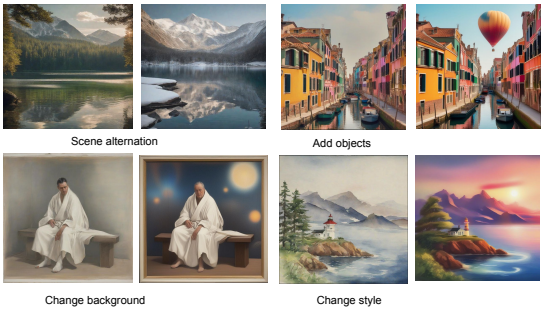

Figure 3. Examples of generated pairs from the MM-Inst dataset showing many image variations.

The last family is instruction-guided methods. Unlike previous methods, these models only requires a vague text instruction such as "add fireworks". InstructPix2Pix [4] achieves this by curating a large machine-generated image editing dataset using P2P, then directly fine-tuning on instruction-following generation. MagicBrush [31] curated a higher quality human-annotated dataset by requesting humans to perform editing operations using tools such as PhotoShop. MGIE [7] utilizes a multi-modal large language model to process editing instructions and input images. But it still operate on a single source image with text-only instructions. All these works are limited to simple instructions and require iteratively applying the model recurrently for complex instructions. Additionally, they cannot take audiovisual input as auxiliary information.

Unlike previous works, our work extends the scope of valid instructions to multi-modal, multi-object inputs and can perform complicated instructions in a single turn.

3. Methods

Our pipeline is illustrated in Figure 1. To generate a sample of an edited image given multi-model instructions, we first leverage a pretrained multi-modal encoder that encodes different modalities into a unified embedding space. We then pass the multimodal instruction through a large language model to generate an embedding sequence. We compute input embeddings of the instruction using a large language model (LLM) and substitute the embeddings of the non-linguistic tokens with these embeddings obtained via multi-modal encoder. The large language model eventually generate an output embedding [gen]. Since there can be multiple input images in the instruction, we let the LLM generate an additional embedding [base] used to retrieve the target image to be edited. A diffusion model then generates the edited image using this embedding and retrieved image.

To maximize training efficiency and provide better control over edited outputs, we train the LLM and diffusion model individually instead of end-to-end. During training, the LLM learns to predict the representation of the edited images. The diffusion model learns to generate an image based on the representation of that image in the same embedding space. We also incorporate a refinement prior module refines the output embedding of the LLM based on human preference.

In particular, given an input text sequence T containing reference tokens such as [image], [audio] referring to multi-modal inputs $\{X_{\text{i}}\}_{\text{i=1:n}}$, we first generate the embedding of $\{H_{\text{i}}\}_{\text{i=1:n}} = F_{\text{enc}}(\{X_{\text{i}}\}_{\text{i=1:n}})$ where F_{enc} is the multi-modal encoder. We then obtain the embedded representation $H_{\text{T}} = E(T)$ where E is the input embedding layer of LLM. The next step is to insert $\{H_{\text{i}}\}_{\text{i=1:n}}$ into the corresponding location of H_{T}. A projection layer P_{enc} is used to project the $\{H_{\text{i}}\}_{\text{i=1:n}}$ to the embedding space of the LLM. This insertion leads to a new sequence of embeddings H'_{T}.

We extract the last hidden state $H_{\text{LLM}} = F_{\text{LLM}}(H_{\text{T'}})$

from the LLM after passing H'_{T} as input. We then locate the position corresponding to two special tokens [base] and [gen]. This gives us two embeddings $H_{\text{base}}, H_{\text{gen}}$. An additional projector P_{out} is employed to map the embedding back to the embedding space of the encoder.

We retrieve the base image id k by finding the embedding with the highest similarity with H_{base}. We then perform DDIM inversion on the source image X_{k} and invert it to a vector z_{k} in the latent space of the diffusion model through DDIM inverse. The diffusion process is then performed on the latent z_{k} of the inverted image using condition H_{gen}, which leads to an output image X_{out} that preserves the spatial structure of the original image while incorporating desired changes. The inference process can be described by Algorithm 1.

Algorithm 1 InstructAny2Pix— Inference

input trained networks $F_{\text{enc}}, F_{\text{LLM}}, F_{\text{diffusion}}$
$H_{\text{i i=1:n}} \leftarrow F_{\text{enc}}(\{X_{\text{i}}\}_{\text{i=1:n}})$ // Encode Multi-Modal:
$H_{\text{T}} \leftarrow E(T)$ // Encode Text
$H_{\text{T'}} \leftarrow \text{insert}(H_{\text{T}}, P_{\text{enc}}(\{H_{\text{i}}\}_{\text{i=1:n}}))$ // Substitute
$H_{\text{LLM}} \leftarrow F_{\text{LLM}}(H_{\text{T'}})$
$H_{\text{base}}, H_{\text{gen}} \leftarrow P_{\text{out}}(\text{extract}(H_{\text{LLM}}))$ // Extract Output
$k \leftarrow \text{argmax}_i(\text{Sim}(H_i, H_{\text{base}}))$ // Retrieval
$z_{\text{k}} \leftarrow \text{inverse}(X_{\text{k}})$ // DDIM Inversion
$z'_{\text{k}} \leftarrow \alpha z_{\text{k}} + (1-\alpha)\epsilon; \epsilon \sim \mathcal{N}(0, \boldsymbol{I})$
$z'_{\text{k}} \leftarrow z'_{\text{k}} \|z_{\text{k}}\| / \|z'_{\text{k}}\|$ // Mixing Latent with Noise
$H'_{\text{gen}} \leftarrow F_{\text{prior}}(H_{\text{gen}}) + H_{\text{gen}} + \beta H_{\text{k}}$ // Mixing Conditions
$X_{\text{out}} \leftarrow F_{\text{diffusion}}(z_{\text{k}}, H'_{\text{gen}} / \|H'_{\text{gen}}\|)$ // Generative Editing
return Edited image X_{out}

Where F_{prior} is the refinement prior, ϵ is i.i.d Gaussian noise introduced to add randomness in generation, α controls such randomness, and β is a hyper-parameter controlling the strength of edits.

3.1. Multi-Modal Encoder

We use ImageBind [9] as our encoder. ImageBind is trained on multi-modal data, including image, audio, depth, and heat maps. We take the image and audio encoder of ImageBind as our multi-modal encoder. We observe that different modalities have different norm distributions, so we perform L_2 normalization on ImageBind features before passing them to the LLM. During training, we keep the encoder frozen. The only trainable part is the projection layer.

3.2. Instruction Following Multi-Modal LLM

We use Vicuna-7b [27] as our base language model. The input sequence consists of a text with multiple references to images and audios, such as "remove [audio] from [image]" and "add [image 1] to [image 2]." The input text is projected to a latent space through the LLM's input embedding layer. This leads to a sequence $H_{\text{T}} \in \mathbb{R}^{L \times D}$ where L is the sequence length and D is the dimension of the latent space. We then identify the locations in this sequence that correspond to references of multi-modal inputs like "[image]," "[audio]" and replace them with corresponding multi-modal features. Since the dimensions of the two embedding spaces are not necessarily identical, we incorporate a MLP projector P_{enc} prior to the substitution. This gives us $H'_{\text{T}} \in \mathbb{R}^{L \times D}$ which is identical to H_{p} except in locations corresponding to reference words such as "image" and "audio".

We add two new output tokens to extend the output of the LLM: namely "[base]" and "[gen]". The [base] token is used to retrieve the desired source image when multiple inputs are present, and the [gen] token is used to produce the conditioning embeddings for the diffusion model. In particular, we apply an MLP projection layer P_{out} to the hidden states from the last LLM layer and obtain $H_{\text{base}}, H_{\text{gen}}$. We compute the cosine similarity of H_{base} and images features from encoder to retrieve the source image, which is then used together with H_{gen} to condition the diffusion models.

During training, the model is trained with a cross-entropy loss for auto-regressively predicting output tokens and a regression loss for the $H_{\text{base}}, H_{\text{gen}}$ embeddings. In particular, the loss can be formulated as:

$$\mathcal{L}_{\text{LLM}} = \mathcal{L}_{\text{ce}} + \|H_{\text{base}} - F_{\text{enc}}(X_{\text{base}})\|_2^2 + \|H_{\text{gen}} - F_{\text{enc}}(Y)\|_2^2 \quad (1)$$

where X_{base} is the target image and Y is the edit outcome.

3.3. Refinement Priors

While it is possible to fully align the output of the LLM to the image embedding space, we discovered that in practice, the LLM fails to converge given a reasonable amount of training time and data because of its large size and slow training. To efficiently train our system, we make use of a Refinement Prior which learns to restore a corrupted image embedding to its original state. We consider two kinds of corruptions: Gaussian noise and domain shift. For Gaussian noise, we simply sample standard Gaussian noise and add it to the ground truth image embedding. For domain shifts, we make use of image-text and image-audio pairs and learn to predict the ImageBind image embedding given corresponding text or audio embedding.

We use a decoder-only transformer as our Refinement Prior. To further enhance the refinement capability, we also use the LAION aesthetic score [24] f of an image as additional input. This facilitates the model to generate samples that are visually pleasing to human eyes. Given an image X, the corruption operator $C(.)$ that randomly adds noise to the embedding or retrieves the paired embedding of a different modality, let F_{ase} be LAION aesthetic predictor, our loss for

the prior module is:

$$\mathcal{L}_{\text{prior}} = \|F_{\text{enc}}(X) - F_{\text{prior}}(C \circ F_{\text{enc}}(X), F_{\text{ase}}(X))\|_2^2. \quad (2)$$

We find that this refinement scheme allows us to improve generation quality without jointly training the LLM and the prior, even though the prior is not explicitly learning to refine the LLM output. We hypothesize that our corruption operator is simply general enough.

While this process involves training on large amounts of paired multi-modal data, including image and audio, the feature extraction can be performed offline, and the final extracted feature dataset only takes up less than 70GB of disk space. The training itself can be performed on eight 24GB Nvidia A5000 GPUs in less than 72 hours.

As a side effect, our corruption scheme gives us the ability to use the prior as paired multi-modal generation and perform tasks such as text-to-image and audio-to-image without instructions. We will provide some brief qualitative evaluation on this feature and leave a more rigorous study for future research since this work focuses on image editing.

3.4. Datasets and Training

3.4.1 Reconditioning Diffusion models

The diffusion model is conditioned on the embedding generated by the multi-modal encoder and trained using the standard diffusion loss [11]:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_t \left[\|\epsilon_t - F_{\text{diffusion}}(z_t, F_{\text{enc}}(X))\|_2^2\right] \quad (3)$$

where z_t is a noised version of the latent in the diffusion process and ϵ_t is the noise at time t. We use 2M samples of high-quality images from LAION-Aesthetic-V2 datasets [24] and SDXL [20] as our base model.

3.4.2 Prior Alignment

To train our refinement prior, we use 2M text-image pairs from LAION-Aesthetic-V2 datasets, which consist of images with high visual quality and corresponding captions. Because the vanilla captions are of low quality, we augment the captions by recaptioning the images using BLIP2 [15]. We also make use of 2M audio-visual pairs sampled from VGG-sound [5], AudioSet [8], and SoundNet [2]. This allows the prior model to learn a broad range of textual-visual and audio-visual correspondences. We also train a version of our prior in a bidirectional way, which means our prior can also be used to predict audio tokens given images. This version is used in our data generation pipeline. The details will be discussed in the following section.

3.4.3 Instruction Tuning

We curated a diverse dataset of 500k instructions, called MM-Inst. Each instruction contains a text description of the instruction and a caption of the desired outputs. We first manually wrote a series of examples for adding, dropping, removing, or replacing objects in the scene. We also included examples of free-form audiovisual instructions, such as style adapation based on reference image or music. We then sampled 500k BLIP2-generated captions from the LAION-Aesthetic-V2 dataset and prompted a Large Language Model (LLAMA2 [28]) to generate editing instructions involving these captions, and output captions. We mark all relevant subjects such as [dog], [cat] in the instruction. During training, they were randomly replaced by the corresponding multi-modal embeddings at a fixed chance. Such embeddings are retrieved in the following way: If the text is a caption of a LAION image, we use the respective image embedding. Otherwise, we obtain the ImageBind text feature and use our prior model to randomly convert it to an audio or image feature. Additionally, we also selected 50k samples and generated pseudo ground-truth using SDXL. When there is a source image, we apply the DDIM inverse and use SDXL to generate the edited image. We visualize some pairs in Fig. 3, and examples of text pairs in Tab. 1. Our generated pairs represent a diverse range of image correspondence, including adding or removing objects, changing the foreground or background, changing the style of the image or the environment of the image. Because of known limitations of DDIM inversion, we only generated scene-level alterations and do not have examples of changing very refined details, such as the color of a person's eyes. This is one major limitation of our data generation pipeline.

During the training, we first pretain our model using converted text embedding from the prior model as targets. We then fine-tune on 50k actual image embeddings generated by the SDXL pipeline.

Table 1. Examples from MM-Inst dataset. [.] indicates multi-modal objects marked by the large language model.

Instruction	Result
Fit the atmosphere of [a piece of music of a futuristic cityscape] to [an old black and white photo of a man, woman, and two young girls]	An image of a futuristic cityscape with the man, woman, and two young girls integrated into the scene.
Remove [the rock formation] from [the night sky is filled with stars and the Milky Way over a rock formation]	An image of the night sky filled with stars and the Milky Way.

4. Experiments

We qualitatively and quantitatively evaluate our results on a wide range of challenging instructions with varying combinations of multi-modal inputs.

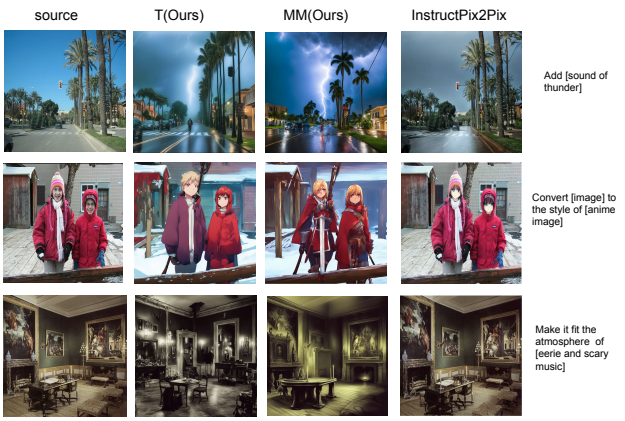

Figure 4. Multi-modal Alignment in InstructAny2Pix. We showcase results of the same instruction expressed in different modalities. We also provide comparisons with InstructPix2Pix using the same instructions. T(Ours): Text-only Instruction. MM(Ours): We replace the corresponding object with multi-modal inputs. InstructPix2Pix: We provide text-only instruction to InstructPix2Pix. Because it was trained on different styles of instructions, we reformatted the relevant instruction to make a fair comparison, e.g. "make it scary".

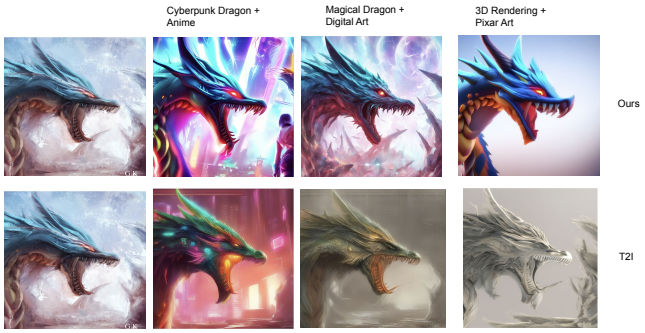

Figure 5. We compare against T2I [19] on image variations. We use an official implementation of T2I based on SDXL. We select the examples from demos on the official homepage. We do not introduce novel styles. In particular, "anime," "digital art," and "3D rendering" are provided style choices on the HuggingFace demo. We demonstrate that taking image inputs can improve the quality of editing results, highlighting the benefits of our setup.

Table 2. Alignment scores of InstructAny2Pix vs Instruct Pix2Pix on MM-Instruct-Edit dataset. Higher is better.

	CLIP_{dir}	CLIP_{im}	CLIP_{out}	Ali.	Qual.
InstructPix2Pix	0.102	**0.766**	0.198	3.31	3.39
InstructAny2Pix(Ours)	**0.132**	0.712	**0.222**	**3.45**	**3.51**

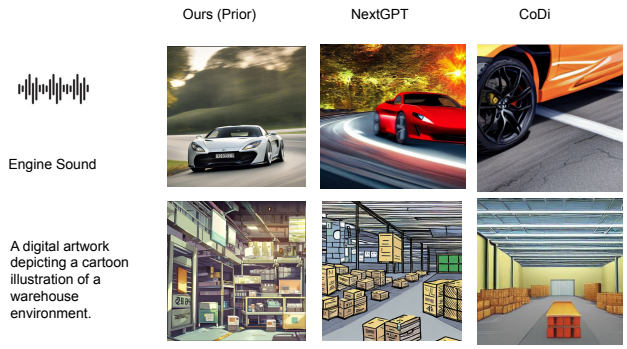

Figure 6. To validate the alignment of our prior, we demonstrate its any-to-image capability. We use our prior model to translate audio embedding to the image space and use our diffusion decoder to generate images. Qualitative results show that it is comparable to state-of-the-art any-to-any generation systems such as NextGPT [30]. As our work focuses on image editing, we leave more rigorous examination and extensions for future works.

Figure 7. We evaluate the effectiveness of our refinement module. Top: conditioned only on target embedding. Bottom: additionally using DDIM inverse of the original image. The leftmost column shows results with no refinement, the right three columns show results of different aesthetic scores f. Generally, refinement improves the result, and a higher aesthetic score leads to higher output quality.

4.1. Multi-Modal Instruction Alignment

Due to our training scheme, our model should be able to understand the same instruction expressed in different modalities. For example, "add [sound of a dog barking]" should not yield significantly different performance than the plain text instruction "add a dog." We validate this by comparing results of text-only instructions and multi-modal instructions. We also compare against an existing text-based instruction-edit method, InstructPix2Pix. Our model is able to follow both text-only instructions as well as multi-modal instructions. InstructAny2Pix achieves comparable or better performance compared with InstructPix2Pix.

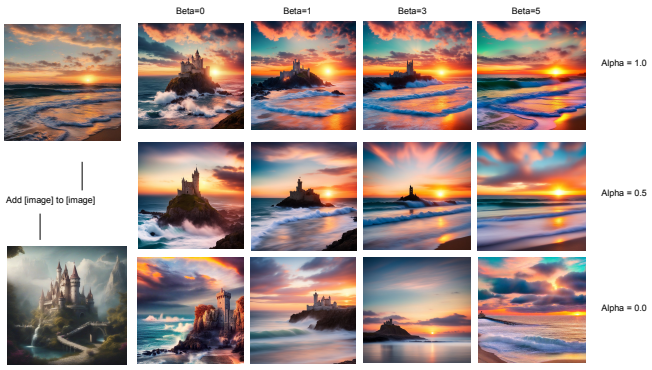

Figure 8. Ablation Study on Control Strength. Hyper-parameters β and α used in Algorithm 1 control the strength of editing. In general, a higher α leads to outputs that respect the spatial composition of the source image, while a higher β leads to outputs that respect the semantics of the original image.

4.2. Multi-Modal Instruction Following Capability

We also provide quantitative evaluation on a set of manually written multi-modal editing instructions. Because no previous methods can perform such a task, we selected a text-only instruction model as our baseline. To make it a fair comparison, we manually formatted the style of instruction to that of InstructPix2Pix dataset, most of which are in the format of "make it X." We report various CLIP-based [22] similarity metrics and human evaluation results in Tab. 2. CLIP_{dir} measures the agreement between changes in captions and the changes in the image, CLIP_{im} measures the similarity between the source and targeted images. CLIP_{out} measures the similarity between edited images and targeted captions. Our model is able to better follow the instruction, showing higher performance in CLIP_{dir} and CLIP_{out}.

We also perform human evaluations. We show decisive advantages in human preference. We asked reviewers to score the alignment (Does the model follow the instructions?) and generation quality (Does the generated image look good?). We outperform InstructPix2Pix on both benchmarks. We recognized that SDXL is generally a stronger diffusion model; hence, we compared against an SDXL-based InstructPix2Pix implementation on HuggingFace. Despite both being based on SDXL, human evaluators consider our model to generate high-quality samples. We provide further discussion in the Discussion section.

4.3. Image Conditioned Style Editing

We compare against T2I [19], a popular method used to create image variations. To make it a fair comparison, we also make use of an official SDXL-based implementations. We select the image from official demos on GitHub and ask T2I to generate variations of the same image. For InstructAny2Pix, we prompt the model with an additional image reference of related styles. We demonstrated that using an image as a reference leads to higher-quality outputs.

4.4. Multi-Object Multi-Modal Instruction Editing

Unlike previous works such as [4], our methods can perform complex editing operations involving many inputs of different modalities. Fig. 9 highlights this capability. Compared with previous methods, our method drastically extends the boundary of instruction-guided image editing.

4.5. Ablations

4.5.1 Refinement Prior

To study the effectiveness of our refinement prior, we provide qualitative results of generated images in Fig. 7. We also show the effect of different aesthetic scores. As shown in the figure, the refinement process significantly improves the image quality. Additionally, we find that the aesthetic score tends to be biased towards high contrast and saturation, so depending on the use case, we can also consider reducing the conditioning score as necessary.

4.5.2 Control Strength

We visually explore the effect of hyper-parameters β and α used in Algorithm 1. In general, a higher α leads to outputs that respect the spatial composition of the source image, while a higher β leads to outputs that respect the semantics of the original image. When both are very high, the model simply gives the original, nearly unedited image as output. These parameters can help the user exert more refined control over desired outputs.

4.5.3 Modality Translation using Prior

Since we train our prior on paired audio-image and text-image data, it can naturally perform paired multi-modal generation. We briefly explore this ability in Eq. (2). We also qualitatively ask human evaluators to compare our work with CoDi and NextGPT. The results are shown in Fig. 10. We achieve better generation quality and are roughly comparable in alignment with these state-of-the-art models. However, this work focuses on image editing, and we use only use such experiments to validate the effectiveness of our prior in the data generation processes.

5. Discussion

5.1. Quality of Generation

We observed that our methods generate higher quality outputs compared with previous works that also make use of SDXL [20]. We hypothesize this may be caused by our data isolation. In particular, we only train on high-quality images in a self-supervised manner, so the model does not

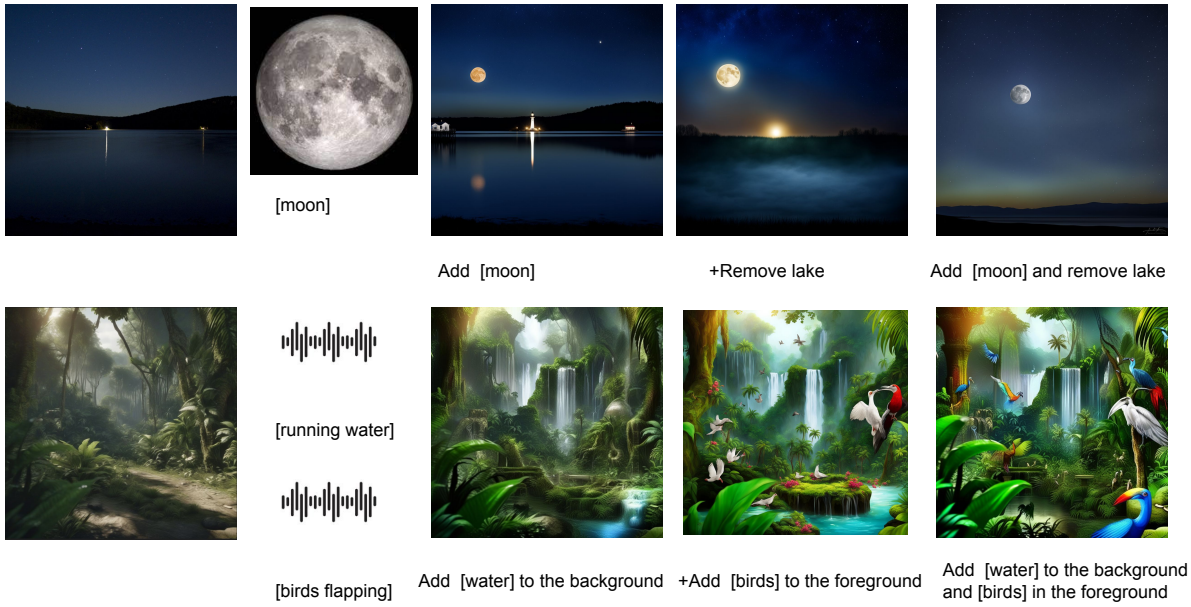

Figure 9. Multi-object Multi-modal Instruction Editing. Unlike previous methods that rely on an iterative chain of editing to perform complex instructions, our model can directly understand complex instructions involving multiple subjects of multiple modalities and perform them in a single turn. In each of the examples, we first provide results of iterative applying our method on two consequent instructions, then we show results of single-turn editing combining multiple instructions.

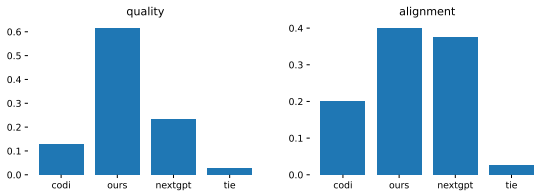

Figure 10. We provide human evaluation image-to-sound and text-to-image capabilities. For image to audio generation, we leveraged AudioLDM2 [16] to decode audio embedding to sound. The human reviewers are asked to decide which model is the most preferable among InstructAny2Pix, CoDi [26], and NextGPT [30]. We use this evaluation only to validate our data generation pipeline as we use the prior to generate audio and image embeddings.

inherit bias from editing datasets that have a lower average quality than the dataset used to pretrain SDXL. While we made this choice primarily for training efficiency, we hope future work can explore the corruption in quality of diffusion models when fine-tuned for editing.

5.2. Limitations

Our model comes with the inherited biases from pretrained diffusion and language models, in particular SDXL and Vincuna. Additionally, our editing instructions mostly consider scene-level editing because DDIM do not provide more refined control. Moreover, since it is very costly to pretrain our model end-to-end (due to the size of LLM), we cannot train on many existing instruction datasets such as [31] that incorporate Photoshop-level image editing or other refined instructions such as changing the contrast of images. Hence, it may not perform well on these benchmarks. Nevertheless, we argue that the ability to incorporate multi-modal inputs and eliminating the need for an iterative chain-of-editing for complex instructions is a major milestone in the development of instruction-based image editing. We hope future works can address these issues.

6. Conclusion

In summary, we propose InstructAny2Pix, a flexible system for editing images based on multi-modal, multi-object instructions. Compared with previous works, we significantly expand the scope of instructions and are capable of performing complex instructions without resorting to a recurrent chain-of-editing. To efficiently train our system which contains an LLM with billions of parameters, we proposed a novel, decoupled training scheme that significantly improved the training efficiency by removing end-to-end finetuning. We hope future works can better address the scalability of the overall system when coupling many modality specific encoders and large language models.

7. Acknowledgement

This research was supported by Cisco.

References

[1] Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023. 1

[2] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. *Advances in neural information processing systems*, 29, 2016. 5, 1

[3] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiao, and Aditya Ramesh. *Improving Image Generation with Better Captions*. 2023. 1, 2, 7

[4] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 1, 3, 7, 2, 5

[5] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 721–725. IEEE, 2020. 2, 5, 1

[6] Aditya Ramesh et al. Hierarchical text-conditional image generation with clip latents, 2022. 2

[7] Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. *arXiv preprint arXiv:2309.17102*, 2023. 3

[8] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 776–780. IEEE, 2017. 5, 1

[9] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190, 2023. 4

[10] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 2

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 5

[12] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023. 2

[13] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. Audiocaps: Generating captions for audios in the wild. In *NAACL-HLT*, 2019. 1

[14] Dongxu Li, Junnan Li, and Steven CH Hoi. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. *arXiv preprint arXiv:2305.14720*, 2023. 4

[15] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 5, 1

[16] Haohe Liu, Qiao Tian, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D Plumbley. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *arXiv preprint arXiv:2308.05734*, 2023. 8

[17] Haohe Liu, Qiao Tian, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D. Plumbley. AudioLDM 2: Learning holistic audio generation with self-supervised pretraining. *arXiv preprint arXiv:2308.05734*, 2023. 6

[18] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. *arXiv preprint arXiv:2211.09794*, 2022. 2

[19] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023. 1, 2, 6, 7, 5

[20] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 1, 2, 5, 7, 4, 6

[21] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2

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

[23] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 2

[24] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 2, 4, 5, 1

[25] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[26] Zineng Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal. Any-to-any generation via composable diffusion. *arXiv preprint arXiv:2305.11846*, 2023. 8

[27] The Vicuna Team. Vicuna: An open-source chatbot impressing gpt-4 with 904, 2

[28] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 5, 1

[29] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1921–1930, 2023. 2

[30] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. Next-gpt: Any-to-any multimodal llm. *arXiv preprint arXiv:2309.05519*, 2023. 6, 8

[31] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. In *Advances in Neural Information Processing Systems*, 2023. 3, 8, 2

[32] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 1, 2

InstructAny2Pix: Flexible Visual Editing via Multimodal Instruction Following

Supplementary Material

8. Details of Datasets

We use SoundNet [2], VGG-Sound [5], and AudioSet [8] for audio-visual alignment. These datasets consist of videos with audio. We extract the audio and the middle frame from the video to create audio-image pairs. SoundNet consists of 802,724 audio-image pairs, AudioSet consists of 888,185 audio-image pairs, and VGG-Sound consists of 197,958 pairs. These numbers represent the number of valid video URLs at the time of data fetching (Oct 2023). They may differ from the original dataset size and the number of valid URLs at the time of writing. We also make use of audio captions from MusicCaps [1] and AudioCaps [13] to create text-audio pairs. These two datasets provide text captions for subsets of AudioSet. They do not introduce new audio files. We use LAION-Aesthetic-3M [24] for text-image alignment, which consists of 2,209,745 valid image URLs at the time of data fetching (Sep 2023). All these datasets are used in our prior training.

9. Implementation Details

9.1. MM-Inst Datasets

9.1.1 Source Captions

We use BLIP2 [15] to generate captions for 500,000 images randomly selected among 2,209,745 images from LAION-Aesthetic-3M. We use an off-the-shelf implementation and do not make any modifications from the default settings. This step is necessary because the original LAION caption contains many non-descriptive texts such as "Wholesale high-quality painting POP art fish free shipping."

9.1.2 Instruction Generation

In the instruction generation phase, we consider the following atomic operations: add, drop, replace, style change, and atmosphere change. In particular, style change refers to changes in visual style, such as changing a realistic photo into a painting, 3D rendering, or anime. Atmosphere change refers to the overall "mood" an image conveys, such as scary, disturbing, exciting, or peaceful. The concept of atmosphere is mostly used when fitting an image to music. We also consider a combination of multiple instructions. For each of the 500,000 image captions, we prompt LLAMA2 [28] with the caption and examples of editing instructions. Since LLAMA2 does not have multi-modal capability, we provide descriptions of multi-modal input and prompt it to generate descriptions of multi-modal input as well. Table 3 lists examples of instructions from each category used for prompting. To ensure the diversity of instructions, we randomly select one or more atomic editing operations for each caption and explicitly prompt LLAMA2 to generate a simple instruction of the specified type or a composite instruction involving the specified types of atomic editing operations. Table 4 illustrates the overall prompting template for a given caption.

Type	Example	Result
Add	Please incorporate [an image of cannon fire] into [an image of a pirate ship sailing on the high sea]	An image of a pirate ship firing at a British Navy warship, fire burning on the ship
Remove	Remove [sound of car accelerating] from [an image of people driving in the countryside road]	An image of a quiet countryside road
Replace	Replace [sound of dog barking] with [sound of a cute cat] for [an image of a dog at the beach]	An image of a cat at the beach
Style	Change [an image of a woman wearing sunglasses in Paris] to the style of [an image of a Renaissance painting of a noble lady]	A Renaissance painting of a woman wearing sunglasses in Paris
Atom.	Make [an image of a cute girl in a school uniform] fit the atmosphere of [a piece of music of stellar constellations]	An image of a cute girl in a school uniform under the night sky

Table 3. Examples of different types of instructions. Atom.: Atmosphere change.

9.1.3 Multi-Modal Feature Generation

For image features corresponding to source captions, we use the features extracted from respective images in the dataset. For additional image and audio features in the instruction, we use our prior model to generate these features from cor-

Template:
I need to generate some multi-modal editing operations. Here are some examples: [examples of instructions] In summary, operations include add, drop, replace subjects. Style transfer and fitting a coherent atmosphere. Please generate following the example above. Given the base caption [base caption]. (You should always use this base). I only need one example on [type(s) of instruction]. Please stop after that. The output should follow the exact format as provided. This is very important! The editing should also be relevant to the original scene. It should not be random. For example, you should not try adding a rainbow to everything. This is very important!

Table 4. Instruction generation prompt used to generate MM-Inst Dataset. Highlighted areas are replaced with corresponding data. Base caption refers to the caption of the edit target.

responding texts. For output image features, we additionally use features extracted from edited images generated by DDIM inversion. These features are used as inputs and outputs of LLM.

9.2. Diffusion Model

We use the typical SDXL [20] implementation, which was originally conditioned on CLIP-ViT-G and CLIP-ViT-L text features. To avoid retraining cross-attention layers from scratch, we adopted an MLP projector that maps ImageBind features to the dimension of the original SDXL conditional inputs.

9.3. Prior

We adopted a decoder-only transformer. In particular, we followed the design of GPT-2 [21]. We substituted the token embedding layer with an MLP projector that maps ImageBind features to the hidden dimension of the transformer. We briefly explored using the instruction-following LLM (Vicuna-7B) [27], but this proved infeasible because of slow training and convergence. We also briefly explored using a diffusion prior similar to that of DALLE-2[6]. In this setup the prior model predicts the the noise added to the target feature given a noised version of the target feature and time embedding. We observe no significant difference in generation quality and did not adopt this approach in our final method because of slower inference (20x).

9.4. LLM

In addition to architecture changes described in Section 3.2, we employed a two-stage training strategy to balance computation cost and convergence speed. During the first stage, we froze the LLM and only updated the projectors, input layers, and output layers. In the second stage, we updated all layers jointly. Using 8 Nvidia A6000 GPUs, the first stage training takes less than two days, while the second stage training takes less than four days.

10. Additional Results

In this section, we provide additional qualitative results to highlight the capability of InstructAny2Pix.

10.1. Additional Editing Results

In Figure 11 and 12, we show more results across a broad spectrum of editing instructions, including adding and removing objects, multi-object scene composition, image style transfer, and fitting an image to the atmosphere of audio. These results complement Figures 9, 2, 4, 5. They demonstrate our model's capability of understanding and performing a diverse set of instructions involving multi-modal inputs.

10.2. Comparison with Text-Only Methods

Since no previous work can perform image editing given multi-modal instructions, we provide a qualitative comparison with methods using text instructions. We compare our results against InstructPix2Pix [4] and MagicBrush [31] in Figure 13. For text-only methods, we convert the multi-modal instructions to equivalent text-only instructions. These results show that our model performs competitively against text-only methods, if not better. It also highlights some of the biases of our model. In particular, our model is biased towards artistic/painting output. This is likely caused by the biases of our dataset and training process. We provide further discussions in Section 12.

10.3. Generalizing to Longer Context

Because InstructAny2Pix leverages the reasoning capability of a LLM, it has the potential to generalize to unseen instructions of a much longer context. To test this capability, we prompt the model to perform image edits based on a piece of story consisting of multiple sentences. This type of instruction is not included in our dataset. Additionally, the context-length is much longer than instructions in our dataset. Fig. 14 show these results. We also provide qualitative comparison against state-of-the are model (GPT4V+DALLE-3) [3]. Since GPT4V cannot understand audio inputs, we provide descriptions of audio in text. While GPT4V+DALLE-3 system cannot perform image edits, it can still generate an image based on textual and visual inputs. Considering the difference in model size and training data, InstructAny2Pix achieves impressive performance.

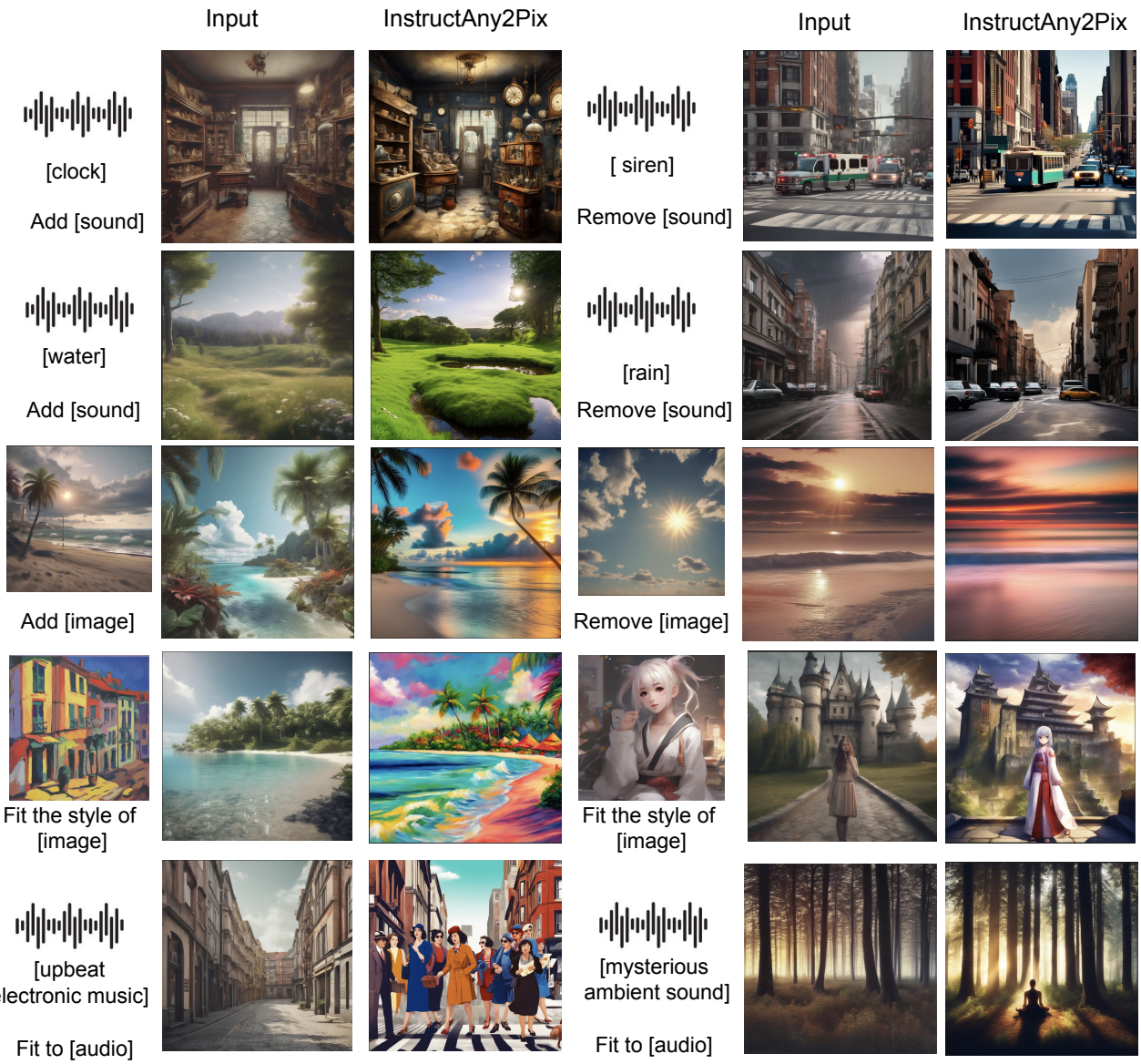

Figure 11. Qualitative Results on MM-Inst Test Dataset. For convenience, we provide text descriptions of images and audios in brackets. At runtime, however, the model is not exposed to text descriptions of multi-modal inputs. We recommend zooming in for a better viewing experience.

11. Failure Cases

We show some failure cases in Figure 15. Our method fails when the sound is rare, ambiguous, or hard to distinguish. For example, it fails to add a volcano explosion to an image of mountain ranges given the sound of a volcano explosion. It also cannot perform counting properly. For example, it fails to add another passing train to an image already with a train in it. This is likely caused by the inherent limitation of DDIM-inversion, which cannot perform fine-grained image editing operations. Additionally, our model may fail to perform style transfer on unconventional pairs. For example, when trying to fit a scenery to the style of minimalist interior design, it adds minimalist furniture to the scene.

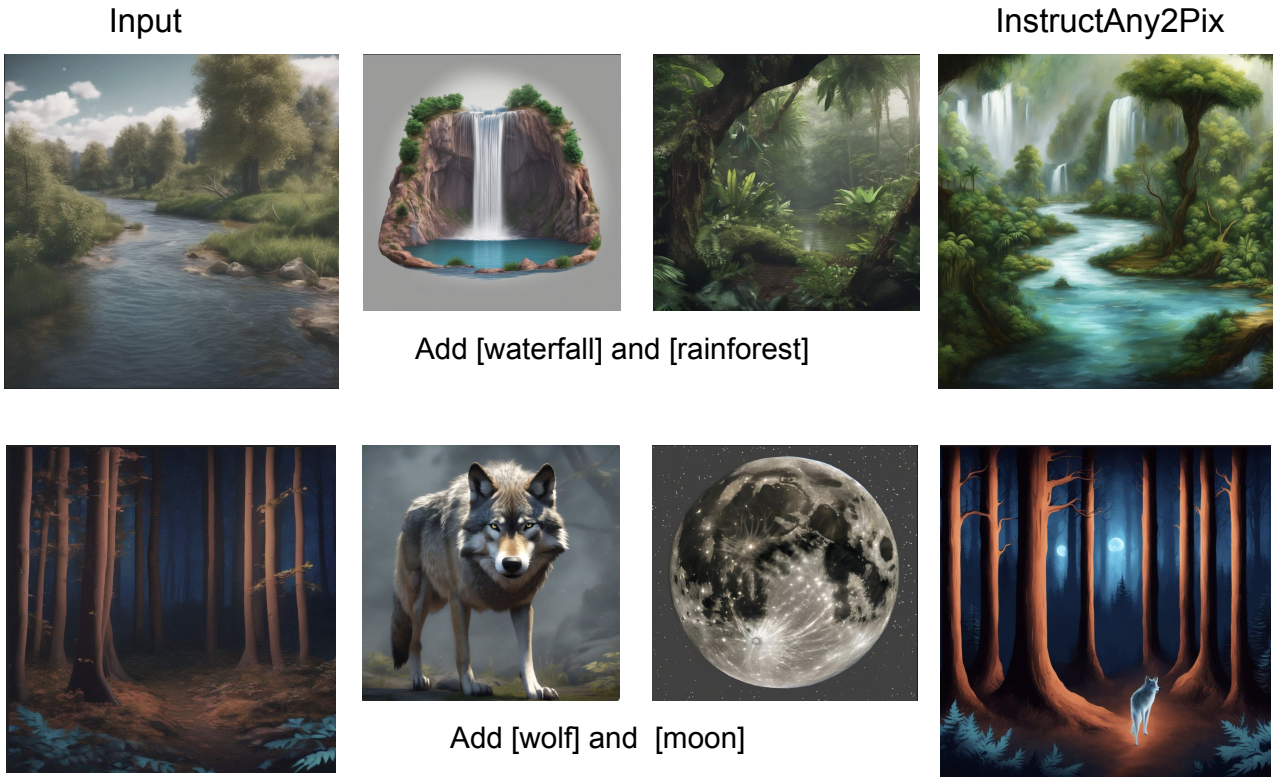

Figure 12. Qualitative Results on MM-Inst Test Dataset. In this figure, we demonstrate multi-image scene composition. For convenience, we provide text descriptions of images and audios in brackets. At runtime, however, the model is not exposed to text descriptions of multi-modal inputs. We recommend zooming in for a better viewing experience.

12. Biases, Limitations and Future works

12.1. Inherited biases

Our model makes use of pretrained a diffusion model [20] and LLM [27]. Hence, it may inherit biases from the training process of these models. For example, it may associate certain careers with particular genders.

12.2. Generation style

Our model tends to bias towards artistic/painting outputs instead of photorealistic ones. This is caused by multiple factors: First, the LAION-Aesthetic-3M [24] dataset used to recondition the diffusion model contains a lot of art and paintings. Additionally, LAION Aesthetic score used to condition the prior model is biased towards high saturation and artistic outputs. Lastly, we use SDXL to generate images for the MM-Inst dataset based on captions. Without explicit style keywords in prompts, we find that SDXL generations are biased towards artistic outputs as well. We will try addressing this limitation by exploring alternative ways of curating a high-quality dataset and explicitly adding diverse style prompts in the generation process.

12.3. Fine-grained editing

Our model fails to perform fine-grained image edits such as counting or editing a small part of an object (e.g., changing the traffic light to green). This is the inherent limitation of DDIM. Additionally, it does not provide "customization." In particular, if one tries to add an image of a dog to another image, there is no guarantee that the dog added to the target image looks identical to the exact same dog. This is caused by the limited expressiveness of a single, pooled, ImageBind feature. Previous methods such as BLIP-Diffusion [14] address this issue by extracting patch-level features and jointly pertaining a multi-modal encoder along with the text encoder and U-Net of the latent diffusion model on synthesized images with the same subject in different backgrounds. It is non-trivial to scale BLIP-Diffusion to multi-object instructions containing multiple modalities and likely to involve prohibitive customization costs.

Since we use a single, multi-modal-aligned embedding to represent each input image, our model currently does

not support instructions involving non-natural images and fine-grained spatial conditioning, e.g., "make the [image] follow this [segmentation map]" or "fit the [depth map] to image". However, in principle, it is possible to integrate existing conditioning adapters such as T2I [19] to our diffusion model and support these extra conditioning factors. We leave such explorations to future work.

12.4. Reconstruction v.s. instructability tradeoff

We observe that there is a trade-off between respecting user instructions and respecting source images. This tradeoff also exists in previous works. Similar to InstructPix2Pix[4], we introduce hyper-parameters that gives user control over how much should the output follow the source image (Fig. 8). In general, our model may introduce minor unintended changes to the image since we do not directly finetune the diffusion model on image-editing. With improvements in base modality-specific models, we expect these issues to be mitigated even in the multimodal setting.

In spite of these limitations, our model successfully extends the scope of image editing instructions to multimodal, multi-object inputs while achieving a favorable balance between performance and computation cost. We hope future works will further scale our approach to diverse editing without incurring substantial computational overhead.

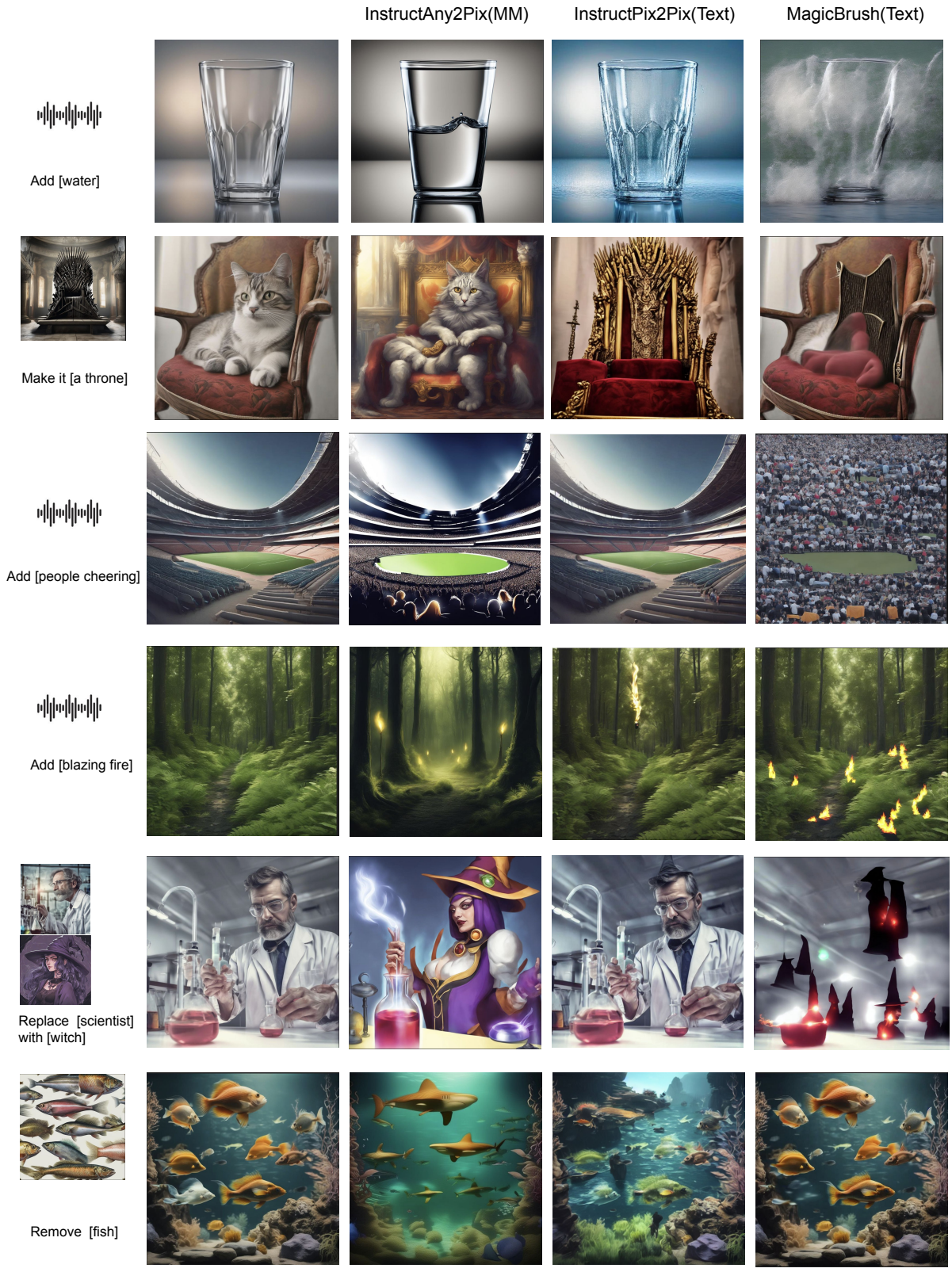

Figure 13. Qualitative Comparison against Magic Brush on MM-Inst dataset. Source and reference images are generated using SDXL [20]. Reference sound is generated using AudioLDM2 [17].

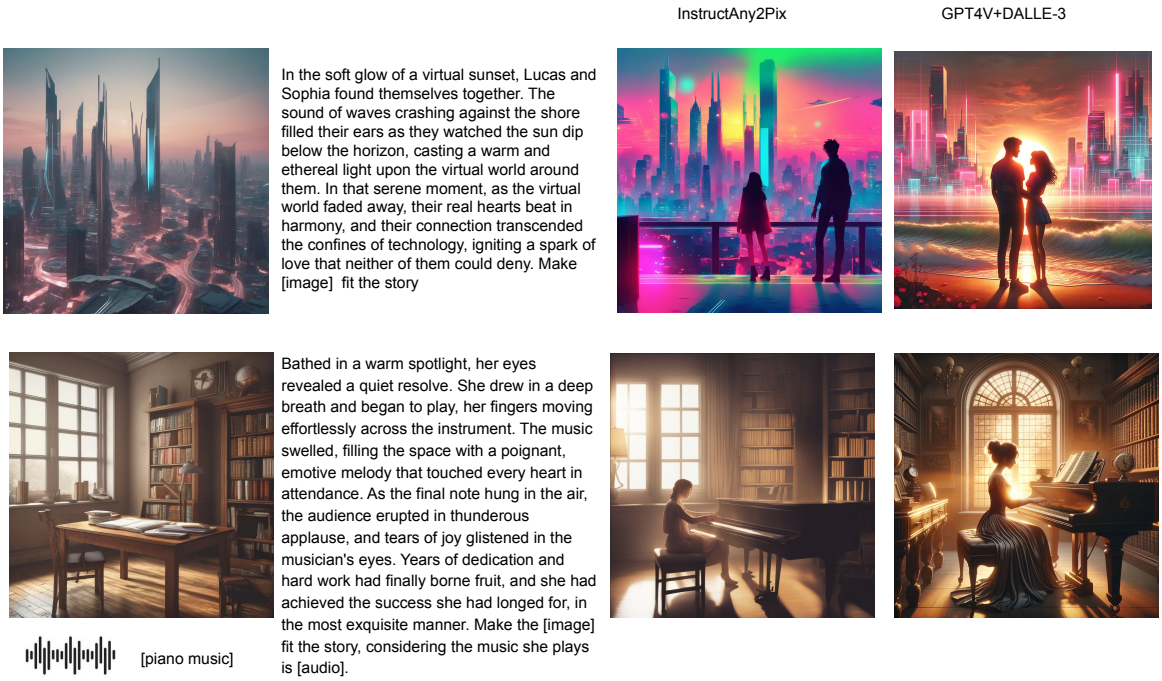

Figure 14. Generalizing to unseen instructions. We provide qualitative results on extending InstructAny2Pix to unseen instructions. In particular, we test it on long-context story telling. We also provide comparisons with state-of-the-art model (GPT4V+DALLE-3)[3]. Considering the difference in model size and training data, InstructAny2Pix achieves impressive performance. In the first example, it was able to reason on the context of the love story and decide to add a couple to the image without explicitly instructed to do so. In the second example, it was able to extract supplementary information from audio that the "instrument" referred in the passage is a piano. Since GPT4V cannot process audio, we reveal the anser in text for it.

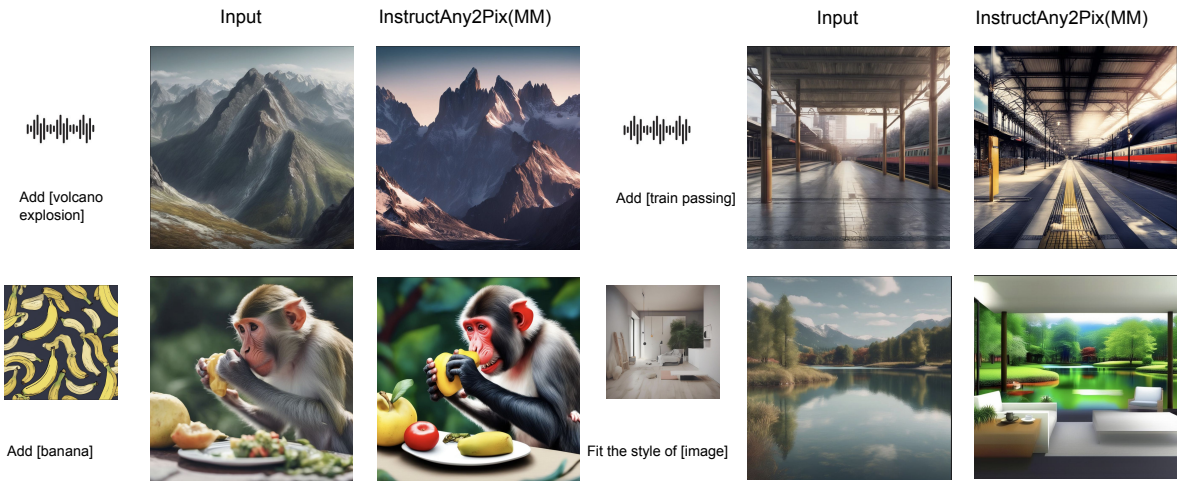

Figure 15. Failure Cases. (1) The sound of a volcano explosion is ambiguous and hard to distinguish. (2) Fails to add another passing train when there is already a train in the scene. (3) Fails to edit small details because of limitations in the training recipe. (4) Fails to perform style transfer when the intended style transfer is not intuitive, e.g., fitting a scenery to an interior design. In this example, the content of the reference image is used in lieu of the style.