InstructAny2Pix: Image Editing with Multi-Modal prompts.

Anonymous ACL submission

Abstract

Image editing has made incredible progress in recent years. Early works only supported caption-guided editing, but recently, free-form text instructions and reference images have been incorporated to allow for more flexibility. However, existing methods still struggle with complex editing instructions involving multiple objects or reference images. We present InstructAny2Pix, a novel image editing model that leverages a multi-modal LLM to execute intricate edit instructions. Compared with previous works, InstructAny2Pix extends the flexibility of edit instructions in three key ways: First, it can perform complex instructions involving multiple object edits; second, it supports the interleaving of text instructions with multiple reference images; and third, it supports audio and music inputs as part of the edit prompts, unlocking creative applications such as album cover generation and music-inspired merchandise design. To evaluate the effectiveness of InstructAny2Pix, we propose two new benchmark datasets, MM-Inst and Dreambooth++, consisting of human-written, multi-modal prompts. InstructAny2Pix outperforms baselines on these two proposed multi-modal benchmarks, as well as on conventional image editing benchmarks such as InstructPix2Pix.

1 Introduction

The ability to edit an existing image using free-form text instructions vastly expands the usability of image editing models. Compared with early works, such as Prompt2Prompt(Hertz et al., 2022), which require caption pairs, instruction-based image editing methods, such as InstructPix2Pix(Brooks et al., 2023), offer users unparalleled flexibility to describe edit instructions in natural language, such as "add a dog." More recent models, such as Kosmos-G(Pan et al., 2023), additionally accept reference images, allowing users to add a specific dog from the reference image to the scene. Despite these progresses, existing methods still have limited instruction-following capabilities. For text-guided edits, they are limited to simple instructions on which they were trained and cannot generalize to complex instructions involving multiple objects, such as "add a wolf howling under the moon." For image-guided edits, they often struggle to complete complex instructions, such as "replace the cat with [reference image]," while faithfully respecting both the image to edit and the reference image.

To address these limitations, we propose InstructAny2Pix, the first instruction-following image editing system capable of following a wide range of complex, multi-modal, multi-object instructions. Specifically, InstructAny2Pix not only supports text instructions involving multiple objects, such as "add a wolf howling under the moon" or "add a cat and remove the dog," but it can also optionally accept multiple reference images of the objects (e.g., the wolf and the moon). Furthermore, it works with arbitrary free-form, multi-modal instructions interleaving text, image, and audio, such as "change [image A] to the style of [image B]" or "fit [image] to [music]," while previous multi-modal models only support limited modalities (i.e., image) and very basic instructions (e.g., add, remove).

InstructAny2Pix greatly enhances the flexibility and usability of image editing models. When creating a scene with multiple objects, instead of writing lengthy descriptions for each object, uploading reference images can be far more efficient. Music or audio inputs, though less obvious, also unlock creative possibilities, such as designing T-shirts based on music or dynamically adapting a background image during live performances. While these tasks could be done through text instructions, they would require designers to first develop specific ideas like "add a circle to the T-shirt" or "change the background to sunset," which demands artistic intuition and experience. Using music as a prompt, however,

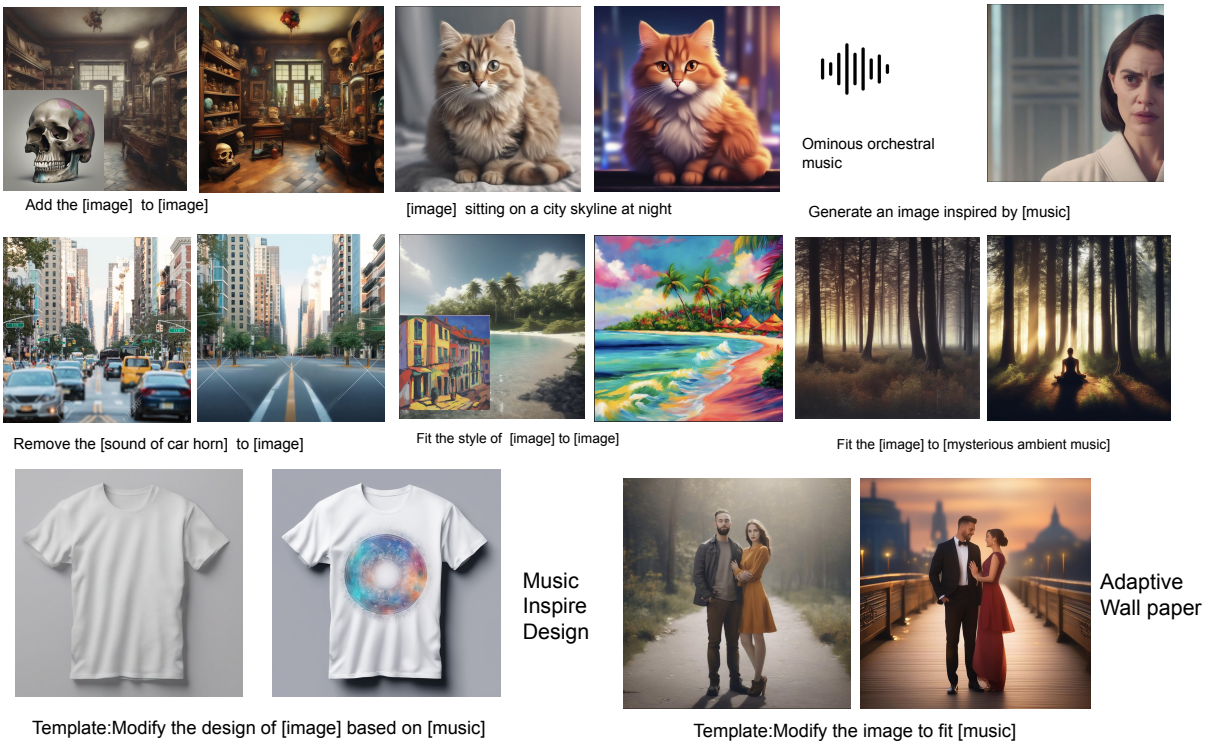

Figure 1: Illustration of InstructAny2Pix's ability to flexibly edit an image based on a variety of multi-modal instructions. More examples of audio-guided editing are provided in the supplementary demo video.

simplifies this process by reducing the creative burden. Even if the designer isn't explicitly seeking a music-inspired design, experimenting with various tracks and selecting from generated options is quicker than manually drafting multiple design ideas or refining text prompts. Examples of these innovative applications are shown in Fig. 1.

Concretely, we build InstructAny2Pix by combining a multi-modal encoder that "perceives" audiovisual inputs, a large language model that "reasons" about the edit instructions, and a diffusion model that "draws" the edited results. To achieve flexible image editing with multi-modal prompts, we curated a large training dataset of diverse multi-modal editing instructions in three steps. In the first step, we prompt a large language model (LLM) to generate a diverse set of complex, multi-modal edit instructions and captions of intended edit results. Since the LLM cannot generate reference images and audio, we ask it to generate captions of these multi-modal prompts instead. In the second step, we use off-the-shelf text-to-image and text-to-audio models to create reference images and audio from the captions generated in the previous step. In the final step, we employ a pool of caption-based edit methods alongside segmentation and in-painting models to generate edit results using the input images and the captions of intended edit results.

To evaluate InstructAny2Pix on the proposed tasks, we created two benchmark datasets: MM-Inst and Dreambooth++. Both datasets consist of high-quality, human-written, multi-modal edit instructions. MM-Inst comprises complex multi-modal edit instructions interleaving text, image, and audio. Dreambooth++ specifically focuses on image editing with reference images. Through extensive experiments, InstructAny2Pix outperforms existing baselines on these two benchmarks. InstructAny2Pix also achieves competitive performance on simpler instruction datasets, such as InstructPix2Pix, in a zero-shot manner, highlighting InstructAny2Pix's ability to adapt to unseen prompts. After fine-tuning on the InstructPix2Pix dataset, InstructAny2Pix was able to outperform existing baselines.

2 Related Works

2.1 Instruction-Guided Image Editing

There are numerous image-editing methods based on text-to-image diffusion models (et al, 2022; Rombach et al., 2021; Podell et al., 2023; Kawar et al., 2023). The earliest works required a pair of source and target prompts to perform an edit.

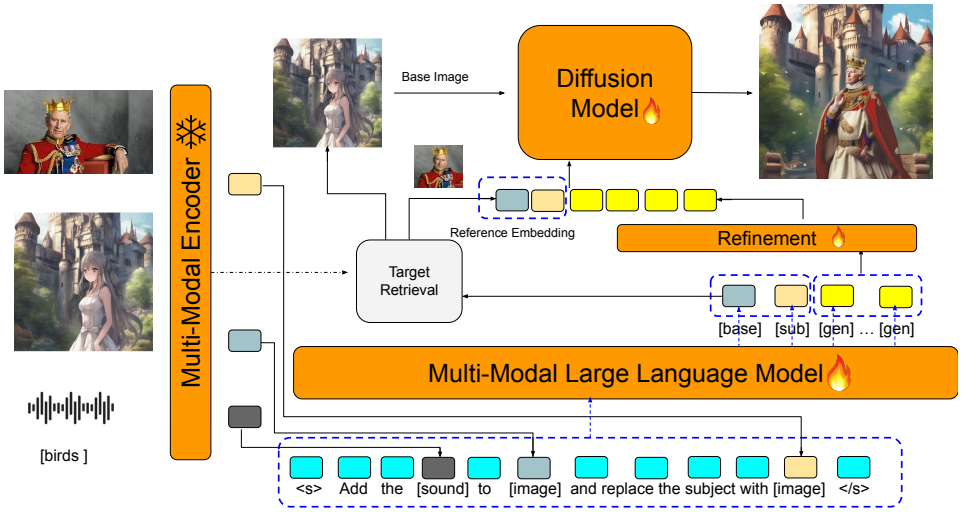

Figure 2: The InstructAny2Pix pipeline consists of three building blocks: a multi-modal encoder that "perceives" audiovisual inputs, a large language model that "reasons" about the edit instructions, and a diffusion model that "draws" the edited results. For improved training and generation, we include an additional refinement module to refine the LLM outputs.

Common approaches include DDIM (Song et al., 2020), Prompt2Prompt (P2P) (Hertz et al., 2022), Plug-and-Play (Tumanyan et al., 2023), and Null-text Inversion (Mokady et al., 2022). These models have very limited flexibility. To achieve good editing results, users must provide long, detailed captions paired in a specific way.

By contrast, instruction-guided image editing methods only require vague text instructions, such as "add fireworks." InstructPix2Pix (Brooks et al., 2023) first achieved this by curating a large machine-generated image-editing dataset using P2P and then directly fine-tuning a diffusion model end-to-end on this dataset. MagicBrush (Zhang et al., 2023) curated a higher-quality human-annotated dataset by requesting humans to perform editing operations using tools such as Photoshop. MGIE (Fu et al., 2023) utilizes a multi-modal large language model to process editing instructions and input images. While it achieves better results than pure diffusion-based methods like InstructPix2Pix and MagicBrush, it still operates only with a single source image and text-only instructions.

Unlike previous works in this area, our work extends the edit instructions to multi-modal, multi-object instructions, greatly enhancing the flexibility of image editing models.

2.2 Multi-Modal Conditioned Generation

Parallel to these image editing methods, there have been previous attempts to achieve image generation with multi-modal conditioning using multi-modal language models. BLIP-Diffusion (Li et al., 2023a) incorporates BLIP (Li et al., 2023b) as a multi-modal encoder that generates subject embeddings for the diffusion model. Using this approach, it can generate images following text prompts and reference images. Kosmos-G (Pan et al., 2023) directly aligns the representation space of multi-modal language models with that of a diffusion model. Kosmos-G allows image generation based on multiple reference images. However, since these works focus on generation rather than instruction-based image editing, they support neither the removal and replacement of objects nor other free-form instructions. They also cannot faithfully respect the spatial structure of input images.

Audio-guided image generation is a relatively uncharted area. AAI (Yang et al., 2023) achieves sound-guided generation by aligning audio representations to reference images. This method is very limited in that it requires retrieving 3-5 reference images and performing gradient descent optimization steps for each audio input at inference time.

Unlike previous works in this area, our work is the first to support interleaved audiovisual inputs and free-form image editing instructions.

3 Methods

3.1 Model Architecture

The architecture of InstructAny2Pix is illustrated in Figure 2. It consists of a multi-modal encoder that maps multi-modal inputs to a unified latent

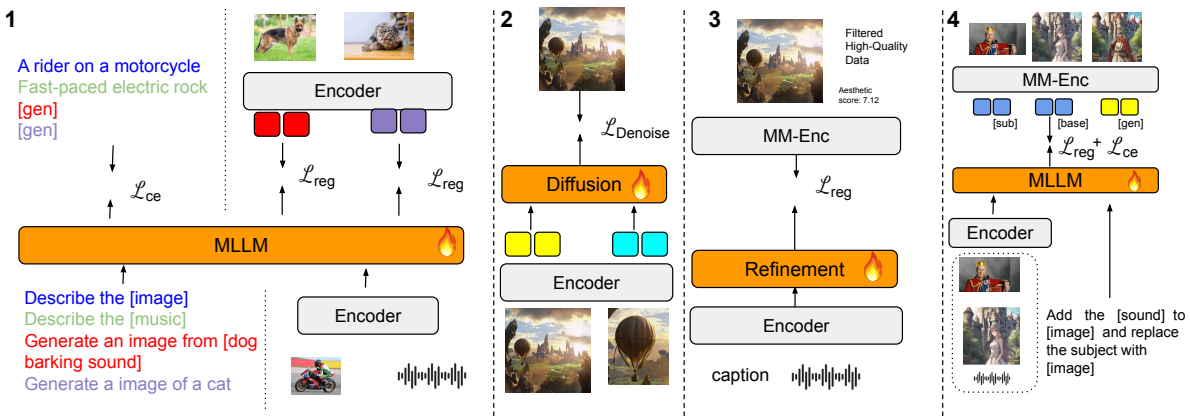

Figure 3: Training pipeline of InstructAny2Pix consists of four steps. 1. Pretraining of Multi-Modal LLM with text-to-x and x-to-image tasks. 2. Pretraining of Diffusion Decoder 3. Pretraining of Refinement Module. 4.Instruction Finetuning

space, a multi-modal LLM that generates a set of edit tokens autoregressively, and a diffusion image decoder that generates the edit result conditioned on the input image and edit tokens. We initialize the multi-modal encoder with Imagebind(Girdhar et al., 2023), the LLM with Vicuna-7B(Team, 2023) and the diffusion image decoder with SDXL(Podell et al., 2023).

The input of InstructAny2Pix consists of a multi-modal instruction (T, I, A). It includes a text instruction T, a set of images I and a set of audio pieces A. I contains the input image and optional additional reference images. It is always non-empty. A contains optionally referece audio pieces. It can be an empty set.

We first leverage the multi-modal encoder Enc to encode reference images and audios into embeddings $E_I = Enc(I), E_A = Enc(A)$. We use the token embedding layer Emb_{LM} of the LLM to obtain the text embedding $E_T = Emb_{LM}(T)$. We then interleave the text, image and audio embeddings into a sequence of instruction embeddings E_{inst}. These embeddings are then passed to the large language model, which generates a sequence of *discrete* control tokens C autoregressively. For each token, we also extract the corresponding *continuous* hidden state from the last transformer block to form a sequence of control embeddings $E_c = (E_{base}, E_{sub}, E_{gen})$.

Each control token belongs to one of the following type: [base], [gen], [sub]. The embedding of the [base] token is used to retrieve the input image. The embedding of [sub] tokens is used to retrieve reference images of relevant subjects in the prompt. Typically there is only one [base] token, but there can be more than one [sub] tokens to account for multiple reference images. [sub] token will only be generated if the referenced object should appear in the final image. For example, in instruction "replace [image of object A] with [image of object B]", only the [sub] token corresponding to [image of object B] will be generated.

The [gen] embedding E_{gen} are further processed by a refinement module. The output of the refinement module $E_{refined}$ is used to condition the diffusion image decoder Dec alongside retrieved images to create the final edit result X_{out}. The whole process can be summarized in the following formula.

$$E_A = Enc(A), E_I = Enc(I) \quad (1)$$
$$E_T = Emb_{LM}(T) \quad (2)$$
$$E_{inst} = Interleave(E_T, E_A, E_I) \quad (3)$$
$$(E_{base}, E_{sub}, E_{gen}) = LLM(E_{inst}) \quad (4)$$
$$E_{refined} = Refinement(E_{gen}) \quad (5)$$
$$X_{out} = Dec(E_{refined}, \quad (6)$$
$$Retrieve(E_{base}, E_{sub})) \quad (7)$$

To account for mismatches in the dimension of embeddings, we add MLP projectors where needed. We provide additional details in Appendix F.1.5.

3.2 (Continued) Pretraining

3.2.1 Diffusion Image Decoder

The diffusion decoder was initialized with SDXL, which generates images based on CLIP text embeddings. At inference time, we want the decoder to generate images based on E_{gen}, which encodes

some global semantics, and reference images retrieved by E_{sub}, which contain some objects that should appear in the edit result. To achieve this end, we repurposed SDXL to generate images based on the CLIP image embedding of the whole image E_{global}, and the CLIP image embeddings of objects in the image E_{obj}. When performing an image edit, we use E_{gen} generated by LLM as E_{global}, and the CLIP image embeddings of retrieved reference images as E_{local}.

To obtain object-level encodings E_{local}, we employ an object detector to find the bounding boxes of objects in the images. We crop the image using the bounding boxes, and use the CLIP embedding of the cropped image as E_{local}. To prevent biases and limitations of the object detector (e.g. the detector cannot detect certain classes), we also incorporate additional bounding boxes sampled randomly from a uniform distribution. We randomly drop E_{global} and E_{obj} independently during the training to prevent the model from over-relaying on one of the embeddings while ignoring the other.

We use a subset of LAION-Aesthetics-V2 (Schuhmann et al., 2022) dataset with 4M images for this task. We only use the images in the dataset, ignoring the text captions.

3.2.2 Multi-Modal LLM

We initialize the LLM with Vicuna-7B (Team, 2023). Since Vicuna was only trained for language modeling, we continue to pretrain it on multi-modal data consisting of images, texts, and audio. We use text-image pairs from LAION-Aesthetics-V2, text-audio pairs from Audioset(Gemmeke et al., 2017), LP-MusicCaps (Doh et al., 2023), Audiocaps(Kim et al., 2019), and audio-image pairs from SoundNet(Aytar et al., 2016), VGGSound(Chen et al., 2020). We compose multi-modal prompts for 4 tasks as our pretraining objective: image captioning, audio captioning, text-to-image generation, and audio-to-image generation. Examples of these prompts are "describe the [image]", "generate an image of a cute dog running in a garden", and "generate an image based on [audio]". For captioning task, we apply autoregressive next-token prediction loss. For generation tasks, we set the target token to "[gen]" and apply next-token prediction loss. Additionally, we also extract the embedding E_{gen} and apply L2 regression loss in the embedding space. For example, if the prompt is "generate an image based on [audio]", we apply the next-token classification loss between the output logits and the target sequence "[gen]", and apply the L2 regression loss between the output embedding and the visual embedding of the target image.

3.2.3 Refinement Model

After pretraining the Diffusion Image Decoder and Multi-Modal LLM independently, we observe that directly use the output embedding E_{gen} of LLM as the conditioning embedding E_{global} of the image decoder leads to very low-fidelity image outputs. This occurs because most image-audio pairs in the training data come from low-fidelity YouTube videos (LAION Aesthetic Score < 4.5), while typical text-to-image training schemes use datasets with high-fidelity images (LAION Aesthetic Score > 5.0). To mitigate the effect of low quality data, we incorporate a refinement module, which is a transformer that learns to improve the image quality in the embedding space.

Concretely, the refinement module takes an image embedding E and a target aesthetic score s, and generates $E_{refined}$, which is the embedding of an image with the same semantics but achieves the target fidelity. Since pairs of two images that are identical in semantics but different in fidelity are hard to find, we make use of existing image-text and image-audio pairs to generate $(E, E_{refined}, s)$ triples for training. Given an image-text or image-audio pair, we use the visual embedding of the image as $E_{refined}$, and the aesthetic score of the image as s. We use the corresponding text or audio embedding as E. While E is not an actual image embedding, it is still an embedding representing the same semantics, since the latent space of different modalities are aligned through the multi-modal encoder.

At inference, we use the LLM output embedding E_{gen} as E and set the target score s to a high number (e.g. 6.0). This process is shown in Fig. 3.

3.3 Instruction Fine-tuning

3.3.1 Data

To train InstructAny2Pix for image editing tasks, we curated a diverse dataset of 500k instructions and corresponding image pairs, called MM-Inst. The dataset generation pipeline consists of three steps: text instruction generation, reference multi-modal inputs generation, and input-output image pair generation.

In the first step, we prompt a Large Language Model (LLAMA2 (Touvron et al., 2023)) to generate creative instructions using 36 manually written

examples. Since the LLM cannot generate reference images, audio and music, we ask the LLM to generate the captions of multi-modal references instead. To further increase the diversity of instructions, we prompt the language model to generate instructions based on ground-truth music captions from LP-MusicCaps and AudioCaps, as well as ground-truth image captions from LAION. We provide further details in Appendix F.2

In the second step, we curate the corresponding reference images and audios using the captions created in the first step. If a caption is a ground-truth image, audio or music caption, we directly fetch the corresponding media. If the caption is generated, we use SDXL(Podell et al., 2023) and AudioLDM2(Liu et al., 2023a) to generate images and audio respectively.

In the last step, we curate pairs of input images and edit results using a combination of six methods:

1) Edit image using captions and Prompt2Prompt (Hertz et al., 2022) 2) Edit image using captions and Plug-and-Play (Tumanyan et al., 2023) 3) Use DDIM(Song et al., 2020) inversion on the source image and generate a new image using target prompt with inversed latent. 4) For object removal, we use an open-vocabulary object detector (GroundingDINO (Liu et al., 2023c)) to locate the object and perform inpainting in the area of the removed object. 5) For object addition, we first generate an image using target prompt as the target image. Then we use the detector to localize the added object and remove it through inpainting. The resulting image is used as the source image. 6) For object replacement, we first follow the removal procedure. Then we perform inpainting in the area of removed objects using the replacement object as prompts.

When the detector fails to localize the object or yields low confidence scores, we fall back to caption-based methods. Following InstructPix2Pix (Brooks et al., 2023) we filter the results using CLIP scores. We additionally filter the results using the LAION aesthetic predictor and remove low quality images. We also provide additional details in Appendix F.2.

3.3.2 Training

3.4 Instruction Guided Finetuning

We fine-tune the LLM using the same objective as the continued pretraining phase, which consists of a next-token prediction loss on the output logits and a regression loss on the output embeddings. Unlike the continued pretraining phase which only includes simple generation or captioning tasks, we use interleaved multi-modal instructions from MM-Inst dataset as the input. We formulate the target output sequence as "[base] [sub] .. [sub] [gen] .. [gen]" where each [sub] corresponds to a reference image. We only add a "[sub]" token if the referenced object should appear in the desired output. For example, in the following instructions "add [a]", "remove [b]", "replace [c] with [d]", only [a] and [d] will have corresponding [sub] tokens in the target sequence. L2 regression loss is applied between the [base] embedding and the visual embedding of the source image, between [sub] embeddings and the visual embeddings of reference images, and between the [gen] embedding and visual embeddings of the edit results. The diffusion model is not directly used in this process.

4 Evaluation

4.1 Evaluation Dataset

Image editing following multi-modal instruction is a novel task with no existing benchmarks. To fairly evaluate InstructAny2Pix's performance in real-world settings, we curated 1500 manually written multi-modal instructions. Unlike previous works which perform evaluations on samples from the same distribution as their training data, our evaluation benchmark caters for diverse real-world use-cases. We call this dataset MM-Inst-Test. We provide further details in Appendix F.3.2.

For image guided generation, Dreambooth(Ruiz et al., 2023) is a commonly-used benchmark. However we find it inadequate to provide a holistic evaluation of multi-modal generative models. Firstly, it only contains two classes of live objects (cats and dogs), which accounts for 9 out of 30 subjects in the dataset. Its diversity is limited. Secondly, its task only involves changing the background of a single subject. This setup cannot evaluate a model's capability of making use of multiple image inputs. To address this gap, we propose Dreambooth++, a Dreambooth-like dataset with more diverse prompts. It consists of 30 subject images which are evenly distributed across humans, animals, small objects, and large structures. We also include 30 diverse background images with corresponding prompts. In total, there are 900 generation tasks. We evaluate models on this dataset using two protocols: single-image and multi-image. The single-image setup is similar to Dreambooth,

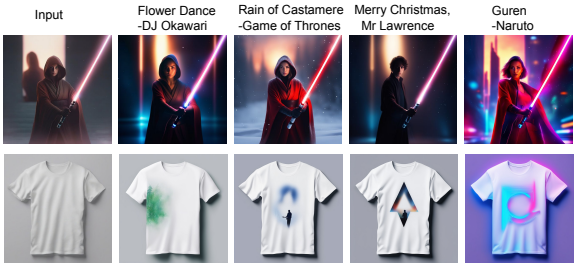

Figure 4: **Music-Gudied Image Variation**: Music uniquely conveys emotions that are hard to describe using other modalities such as language. We show qualitative results of music guided image variation and music inspired design. InstructAny2Pix is able to understand a diverse set of emotions embedded in music and generate creative designs and edits. We include these examples with audio in our supplementary video.

Figure 5: **Editing with Multi-Object Instructions.** Compared with previous text-based method (MGIE) and image-based method (Kosmos-G), InstructAny2Pix uniquely accomplish complex edit tasks.

which requires generating a given subject under different context (background) prompts. The multi-image task requires generating a new image by combining a subject image and a background image. We provide more details in Appendix F.3.2

Additionally, for completeness and fair comparison with existing models, we evaluate our model on 1000 samples from InstructPix2Pix dataset. We report both the zero-shot results and fine-tuned results.

4.2 Instruction guided Image Editing

We evaluate the capability of InstructAny2Pix on MM-Inst-Test dataset described in Sec. 4.1. Because no previous methods can perform such a task, we selected text-only instruction models as our baseline. For fair comparison, we convert all multi-modal instructions to text-only instructions using the captions of referenced audio and image. For InstructAny2Pix, we report the performance of using multi-modal prompts and the performance of using text-only prompts. We also report results on 1000 randomly selected images from InstructPix2Pix dataset. We report quantitative metrics in Table 1. CLIP_{dir} measures the agreement between changes to captions and changes to images, CLIP_{im} measures the similarity between the source and targeted images. CLIP_{out} measures the similarity between edited images and targeted captions.

We also conducted human evaluations on both dataset and report the win rate. Human evaluators are asked to pick a preferred edit output in a one-to-one comparison between InstructAny2Pix and each baseline method. For a fair comparison, we use the text-only version of our method.

The results are shown in Table 1. InstructAny2Pix show decisive advantages in human preference and strong performances in quantitative metrics. We compare with baseline methods InstructPix2Pix(Brooks et al., 2023), Magicbrush (Zhang et al., 2023) and InstructDiffusion (Geng et al., 2023). Notably, we achieve competitive performance on InstructPix2Pix dataset without ever training on such dataset. This result showcases the superiority of our data generation pipeline. It incorporates a diverse range of instructions and enables our model to generalize to unseen instruction patterns. We observe that InstructAny2Pix has slightly higher CLIP_{im} and CLIP_{out} when using only text instructions. This may reflect the fact that multi-modal image editing process are affected by multiple reference images, rather than just an input image and a text instruction.

4.3 Image Conditioned Generation

We evaluate InstructAny2Pix on DreamBench++ dataset described in Sec. 4.1. We conduct both single-image and multi-image evaluation and compare results with BLIP-Diffusion (Li et al., 2023a) and Kosmos-G (Pan et al., 2023). In addition to metrics reported in the previous section, we additionally report DINO scores, which measures the image similarity. For multi-image benchmark, evaluating the DINO similarity of the entire image does not make sense, as the subject is added to the scene may not necessarily occupy the entire image. To address this, we use a segmentation model (Liu et al., 2023c) to segment the subject in generated images. We crop the image according to the bounding box of the object. We report the DINO similarity between the cropped image and the reference subject image as DINO_{sub}, and the DINO similarity between the whole result and the background input image as DINO_{ref}. We provide quantitative metrics in Table 2 and more qualitative comparison in Appendix D. InstructAny2Pix shows a clear advantage in generation quality and image consistency.

Table 1: Multi-modal image editing on MM-Inst-Test dataset and Text-based Image editing on InstructP2P dataset. (I+T+A) and (T) refers to using multi-modal instruction and text-only instruction respectively. The best number is **bolded**. We report both zero-shot and fine-tuned performance on InstructP2P. All baseline methods are trained on InstructP2P. Win rate represents the percentage of human responses that prefer InstructAny2Pix over baseline methods. We use the zero-shot results (Row 2) for human eval on InstructP2P dataset.

	MM-Inst				InstructP2P			
	CLIP_{dir}	CLIP_{im}	CLIP_{out}	Win.	CLIP_{dir}	CLIP_{im}	CLIP_{out}	Win.
Ours(I+T+A)	.099	.816	.260	-				
Ours(T,Zero-Shot)	**.095**	.856	**.270**	-	.147	.808	.312	-
Ours(T,Finetuned)	-	-	-	-	**.182**	**.873**	**.323**	-
InstructP2P	.091	.824	.243	.712	.145	.742	.241	.646
MagicBrush	.084	.807	.199	.707	.165	.760	.250	.698
InstructDiff.	.066	**.940**	.193	.746	.126	.857	.301	.631

Table 2: Image conditioned generation on DreamBench++ dataset. C_{dir}, C_{im}, C_{out} is abbreviated form of CLIP_{dir}, CLIP_{im} and CLIP_{out}. For multi-image setup, numbers are reported in DINO_{ref}/DINO_{sub} format. The best number is **bolded**.

	Single-Image				Multi-Image				
	C_{dir}	C_{im}	C_{out}	DINO	C_{dir}	C_{im}	C_{out}	DINO	Recall
Ours(T+I)	**.147**	.810	**.260**	**.688**	.154	**.789**	**.309**	**.625**/.471	**.841**
BLIP-Diffusion	.089	.779	.231	.660	.091	.701	.292	.526/.422	.693
Kosmos-G	.126	**.843**	.251	.683	**.166**	.740	.286	.485/**.476**	.812

4.4 Discussions

Does InstructAny2Pix outperforms baselines on complex, multi-object instructions? Unlike previous works, InstructAny2Pix can perform complex editing operations involving multiple multi-modal inputs. In figure Fig. 5, we provide visual examples of InstructAny2Pix performing complex instructions where existing models fail. We also visualize the performance gap on single-object and multi-object prompts in Fig. 6. InstructAny2Pix exhibits a larger lead in multi-object prompts. While the numerical performance on instructions with only one object is similar, we still observe qualitative differences. Additional analysis is provided in Appendix D.

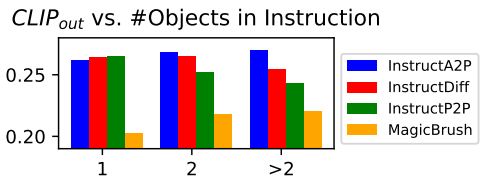

Figure 6: **CLIP_{out} with respect to number of objects on MM-Inst-Test dataset.**

Does the audio/music capabilities of InstructAny2Pix enable useful real-world applications? InstructAny2Pix uniquely enables music guided image editing, which can be quite useful in cases like music-inspired designs and image variations. We show some examples in, Fig. 4. We recognize that connecting music to visual elements (e.g. smooth music to peaceful scenes) can be a subjective process. We provide further discussion in the appendix Appendix B. We also include an audible demonstration video in the supplementary.

5 Conclusion

In summary, we propose InstructAny2Pix, a flexible system for editing images based on multi-modal, multi-object instructions. Compared with previous works, it uniquely supports complex multi-object, multi-modal instructions. It also unlocks creative new use cases such as multi-image synthesis and music inspired designs. We proposed two novel benchmarks: MM-Inst-Test and Dreambooth++ for image editing with multi-modal prompts and multi-image synthesis respectively. InstructAny2Pix outperforms existing baselines on these benchmarks, while also achieving competitive performance on conventional image-editing benchmarks with only text instructions.

6 Limitations

6.1 Biases

Our model makes use of a pretrained diffusion model (Podell et al., 2023) and a pretrained LLM (Team, 2023). Hence, it may inherit biases from the training process of these models. For example, the SDXL is known to have some biases towards certain skin color (Esposito et al., 2023). Our system will inherit these biases.

6.2 Style of Output Images

Our model tends to bias towards artistic/painting outputs instead of photorealistic ones. This is caused by multiple factors: First, the LAION-Aesthetic-3M (Schuhmann et al., 2022) dataset used to pretrain the diffusion model contains a lot of art and paintings. Additionally, the LAION Aesthetic score used to condition the refinement 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.

6.3 Types of Supported Edits

In this work, we explore mostly object-level edits and global edits that change the semantics of images, such as adding and removing objects, changing backgrounds, changing the image style, and changing the overall atmosphere of the scene. InstructAny2Pix does not currently support Photoshop-style edits such as increase the image brightness, zoom in on objects. Users may choose to fine-tune InstructAny2Pixon relevant datasets. We left that for future exploration.

References

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

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

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

Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. 2020. 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.

SeungHeon Doh, Keunwoo Choi, Jongpil Lee, and Juhan Nam. 2023. Lp-musiccaps: Llm-based pseudo music captioning. *arXiv preprint arXiv:2307.16372*.

Piero Esposito, Parmida Atighehchian, Anastasis Germanidis, and Deepti Ghadiyaram. 2023. Mitigating stereotypical biases in text to image generative systems. *arXiv preprint arXiv:2310.06904*.

Aditya Ramesh et al. 2022. Hierarchical text-conditional image generation with clip latents.

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

Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. 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.

Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Han Hu, Dong Chen, and Baining Guo. 2023. Instructdiffusion: A generalist modeling interface for vision tasks. *CoRR*, abs/2309.03895.

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

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

Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. 2023. 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.

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

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. 2023. Pick-a-pic: An open dataset of user preferences for text-to-image generation.

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

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

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

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

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. 2023c. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*.

Karttikeya Mangalam, Haoqi Fan, Yanghao Li, Chao-Yuan Wu, Bo Xiong, Christoph Feichtenhofer, and Jitendra Malik. 2022. Reversible Vision Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10830–10840.

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

Xichen Pan, Li Dong, Shaohan Huang, Zhiliang Peng, Wenhu Chen, and Furu Wei. 2023. Kosmos-G: Generating images in context with multimodal large language models. *ArXiv*, abs/2310.02992.

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

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

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2021. High-resolution image synthesis with latent diffusion models. *Preprint*, arXiv:2112.10752.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510.

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

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

The Vicuna Team. 2023.

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

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. 2023. 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.

Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. 2022. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers`.

Yusong Wu*, Ke Chen*, Tianyu Zhang*, Yuchen Hui*, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. 2023. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*.

Yue Yang, Kaipeng Zhang, Yuying Ge, Wenqi Shao, Zeyue Xue, Yu Qiao, and Ping Luo. 2023. Align, adapt and inject: Sound-guided unified image generation. *arXiv preprint arXiv:2306.11504*.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. 2023. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*.

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

InstructAny2Pix: Image Editing with Multi-Modal prompts.

Appendix

A Table of Contents of Supplementary Material

In the supplementary software submission, we provide the code of InstructAny2Pix. The prompt used to generate the training data is also provided with the code. In the supplementary data submission, we provide the full evaluation dataset, including MM-Inst-Test and Dreambooth++. We also include a demo video with various examples shown in the paper, accompanied by audible music and sound.

B Additional Discussion of Music-guided image editing.

In this section, we discuss two novel use cases of InstructAny2Pix in music-guided image editing. These examples are also included in the supplementary video with accompanying audio. It is challenging to evaluate such capabilities because even humans can associate music and visual elements in various ways. Hence, we focus on qualitative results and show that InstructAny2Pix can reasonably associate music with visual elements. Specifically, we aim to demonstrate that: 1) InstructAny2Pix associates music inputs with visual edits in a consistent manner, rather than performing random edits on the input image; and 2) such associations make sense to humans.

B.1 Music Inspired Design and Music-guided Image Variation.

We provide additional qualitative results of music-inspired design and music-guided image variations. in Fig. A.1. For these results, we prompted InstructAny2Pix with structured templates such as "Please modify the design of [image] based on [music]" and "modify [image] to convey the feeling of [music]". We use the same music prompt as in Fig. 4 from the main paper. InstructAny2Pix was able to grasp the mood conveyed by different pieces of music and generate appropriate images. For example, it consistently associates *Guren*, a Japanese Rock Song used in the anime *Naruto* as its opening, with vibrant, highly-saturated, neon-light-like colors. This suits the fast-paced, highly energetic music well. Similarly, it consistently associates *Rain of Castamere*, a piece from the TV series *Game of Thrones*, with a cold, lifeless atmosphere. In Fig. 4 from the main paper, it turns the background of a Jedi warrior to into a snowfield. In Fig. A.1, it changes lush trees into lifeless branches. These edits align with the slow-moving, somber theme of the piece.

C Live Performance Visuals

As a proof of concept, we used InstructAny2Pix to generate a set of visuals corresponding to an improvised performance involving four pieces. In this setup, we generated an initial image and performed image editing via structured templates. The model was prompted with 10 seconds of each piece. To further understand the reasoning behind each edit, we also prompted the model to generate captions using the template "please describe [music]." This was possible because the music captioning task is included in the pretraining dataset. Using captions, we observed that InstructAny2Pix was able to identify the tempo and associated emotions of the music. We used an iPhone to record the performance, which led to some artifacts in the recording, and this was reflected in the generated captions. Despite this, our model was still able to perform reasonable edits. For example, it associated keywords such as "soft, relaxing" with bright colors, and keywords like "suspenseful, slow" with dark tones. Notably, when the caption included "nostalgic," the model converted modern buildings into antique ones. This particular part is a segment of *Departure* from the series *Rurouni Kenshin*, which describes a farewell of samurais in ancient Japan. InstructAny2Pix was able to capture the essence of this piece and make appropriate edits. An audible version is included in the supplementary video.

The demo is only a proof of concept. The edits were not performed in real time. Instead, we recorded a video and retrospectively applied the edits. However, considering that we only took 10 seconds of music to prompt the model, and that the edits can be performed with structured templates without human text input, it is feasible to build a real-time system for this application using our model.

C.1 Additional Discussions

How did the model learn music-visual correspondence? There are two sources of music-visual correspondence in the training data. First, during pretraining, we incorporated an audio-guided image generation task with the prompt "Generate an image from [sound]," where the sound-image pairs came from audiovisual datasets such as AudioSet. Empirically, these pairs were curated by extracting the audio and corresponding frames from YouTube videos. When the audio piece is music, the image would be what human creators deemed suitable for that music. We show examples of images corresponding to music pieces in Fig. A.3 from the AudioSet dataset. They include a diverse set of images such as album cover art, documentary stills, music videos, video game footage, and nature scenery shots. These reflect the diverse ways humans associate music with images and allow InstructAny2Pix to learn a general music-visual correspondence. Second, during instruction fine-tuning, the MM-Inst dataset includes a music-guided image editing task generated by LLM. An example would be "fit the [image] to [a peaceful, slow-moving, piano music]."

Figure A.1: Additional qualitative results of music inspired designs and image variations. InstructAny2Pix was able to make diverse image edits given a music prompt. We include these examples with audio in our supplementary video.

We observed that removing music-image pairs from the pretraining data led to catastrophic failures, even after fine-tuning on MM-Inst. This suggests that large-scale pretraining with natural music-visual correspondences created by humans is necessary. Interestingly, removing audio-image pairs from the pretraining data did not affect the performance as much. For example, after removing audio-image pairs from the pretraining data, the model failed to perform tasks such as "fit the [image] to [a piece of slow-moving, piano music]." However, the model could still perform tasks like "add [sound of dog barking] to the image." We hypothesize that learning simple, explicit audio-image correspondence is easy (e.g., dog sound to dog images) and can be implicitly achieved if the audio and image embeddings are well aligned with the text embedding. However, learning abstract, implicit music-image correspondence is nontrivial and requires direct training on music-image pairs.

How useful is this capability of music-guided editing? While music-guided editing may seem like a niche application at first glance, we have provided various examples above, including three practical use cases: music-inspired designs, music-guided image variation, and live performance visuals. One of the remaining concerns is usability. For example, would writing a text prompt and uploading music be more inconvenient than just writing a more detailed text instruction? We argue this is not necessarily the case. For example, if a user wants to have 10 good T-shirt designs but lacks expertise in fashion, it would be hard for them to write 10 detailed prompts describing all the visual elements of the designs. However, they could simply write one template, "please modify the design [an image of a blank T-shirt] to [music]," and apply it to tens or hundreds of music tracks, then pick the 10 best results. Moreover, these selected tracks can serve as a medium to apply similar designs to other objects, such as dresses. In Fig. 4 of the main paper and Fig. A.1, we see that the same track leads to consistent designs across different objects. This makes music-guided designs more appealing than alternatives (e.g., randomly generating 100 images of T-shirts and picking one).

In the case of live performance visuals, we can also apply a predefined template and change the background of the stage about 10 seconds after a new piece is played. This makes it particularly suitable for impromptu performances. Since it is impossible to know what will be played or when transitions between pieces will happen, it is impractical to create stage visuals in advance. InstructAny2Pix offers unparalleled flexibility, as it allows the staff to create stage visuals that fit the piece being played with just one click.

In a similar spirit, InstructAny2Pix can also be used in bars and restaurants with a real-time "social media wall," where customers can post photos that are displayed in real time. It would be exciting if the posted photos were automatically adjusted to suit the piece being played by an impromptu artist or just the background music of the venue. Likewise, InstructAny2Pix can be used to create video filters for short videos on YouTube or Instagram. Beyond the use cases the authors have imagined, the possibilities are limitless.

D Additional Comparisons with Baseline Methods.

D.1 Text-Guided Image Editing

We provide a qualitative comparison with methods using text instructions. We compare our results against InstructPix2Pix (Brooks et al., 2023), MagicBrush (Zhang et al., 2023), Instruct Diffusion (Geng et al.,

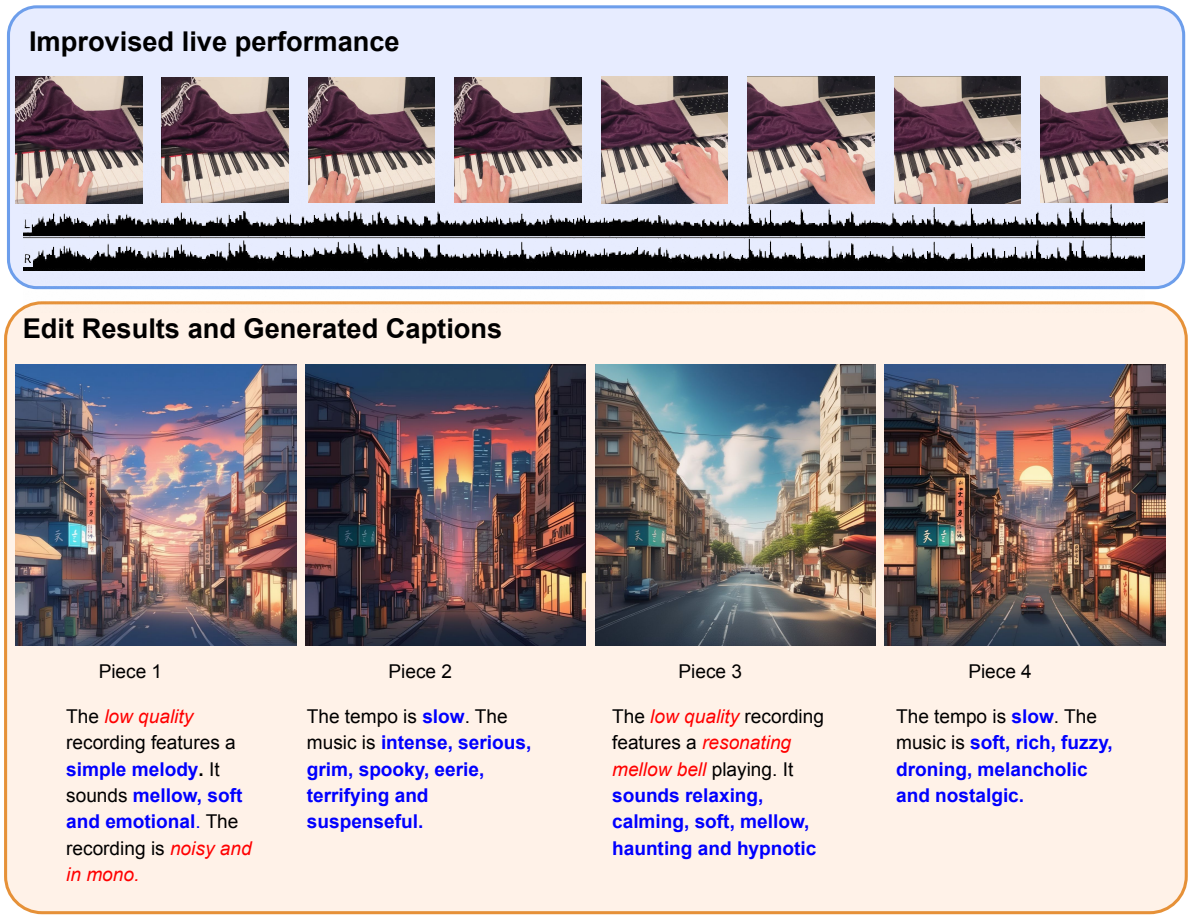

Figure A.2: Qualitative results of live performance visuals. We use InstructAny2Pix to generate a set of visuals corresponding to an improvised performance involving four pieces. We also prompt the model to generate the captions of each music to understand its reasoning process. We mark attributes that are reasonable interpretations of the music with blue, and unreasonable interpretations with red. To maximize creativity, we use a low CFG for this task for better image diversity.

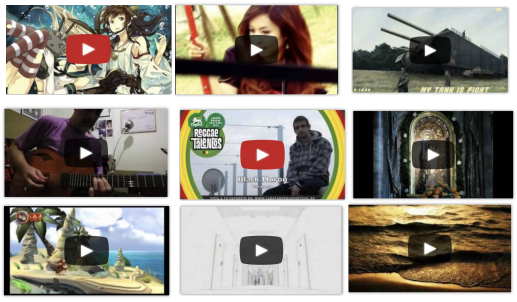

Figure A.3: **Example Thumbnails of Videos with Music.** We show samples of thumbnails from the music category of the AudioSet dataset, which incorporates diverse music-image correspondences.

2023) and MGIE (Fu et al., 2023) in Fig. A.4. For fairness, we incorporate instructions from both the InstructPix2Pix dataset and the MM-Inst-Test dataset. For MM-Inst-Test, we convert the multimodal instructions to text by replacing the multimodal token with captions of the referred image and audio. For InstructAny2Pix, we use the checkpoint that is not trained on InstructPix2Pix dataset for these results.

InstructAny2Pix shows better editing results on both datasets. Particularly, on some tricky examples, such as changing a sea turtle into an elephant and adding water to the glass, InstructAny2Pixis the only model that can successful perform the edits. These results are exceptionally impressive, considering that InstructAny2Pix is not trained on the InstructPix2Pix dataset, unlike other methods.

D.2 Image-Guided Synthesis

We provide a qualitative comparison with multimodal generation methods that use reference images as prompts. We compare our results against BLIP-Diffusion (Mangalam et al., 2022) and Kosmos-G (Pan et al., 2023) in Fig. A.5. Visual results show that InstructAny2Pix outperforms these two baselines both in terms of generation quality and image consistency. We also conducted human evaluation using

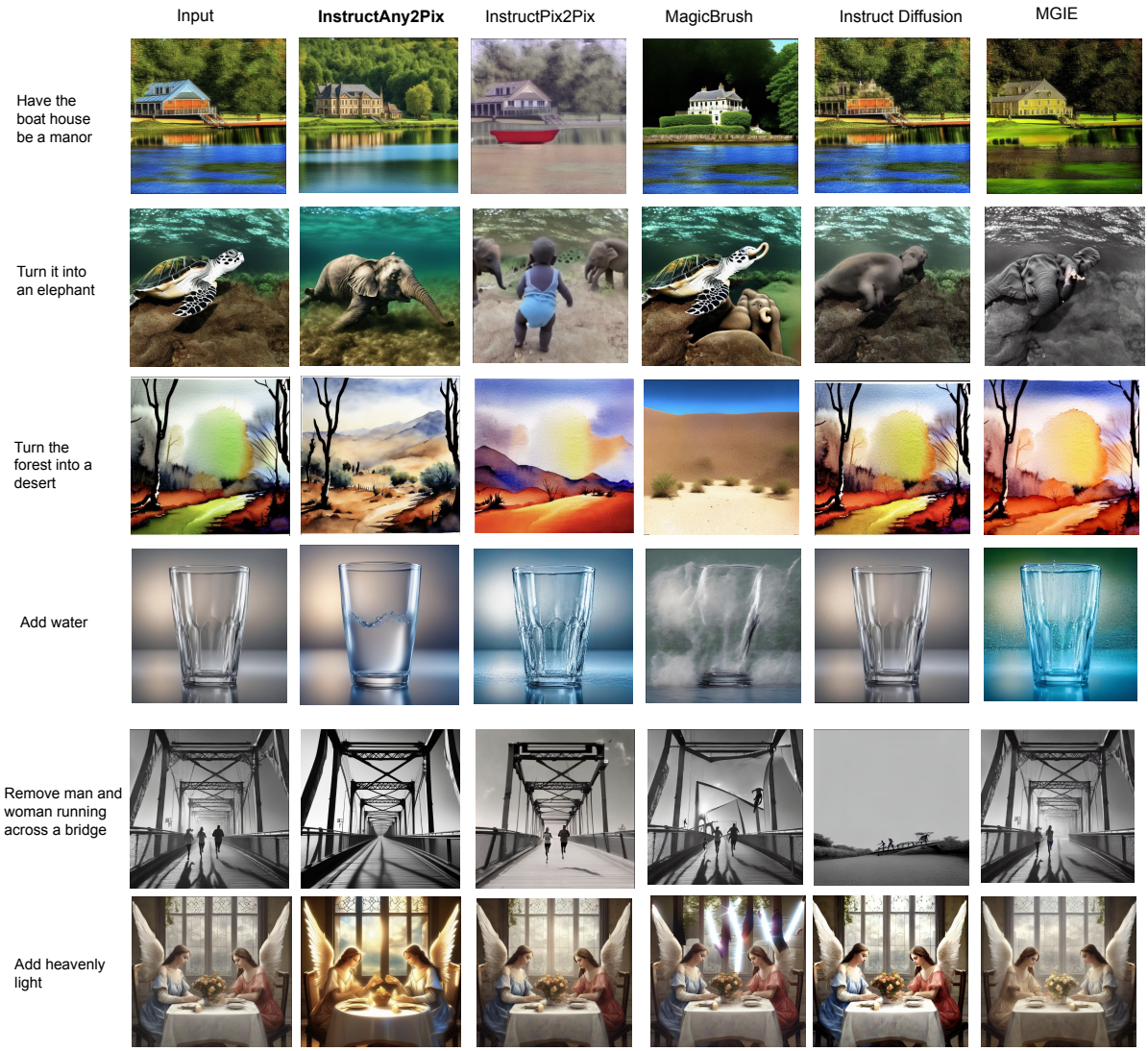

Figure A.4: **Qualitative Comparison Against Text-based Editing Methods.** We show editing results of different models using diverse editing instructions. The top three rows are sampled from the InstructPix2Pix dataset, and the bottom three rows are sampled from the MM-Inst-Test dataset.

Amazon Mechanical Turk. We asked users to pick the best result in a side-by-side comparison of InstructAny2Pix and baseline methods using generations of Dreambooth++ dataset. We achieved a win rate of 79.0% against BLIP-Diffusion and 86.2% against Kosmos-G.

D.3 Comparison with MGIE

We attempted to compare against MGIE (Fu et al., 2023), another image editing method based on Multi-Modal Language Model, on text-based editing. However, we are unable to reproduce the results using the official checkpoint in the Github repo. As our best efforts, we provide qualitative comparisons in Fig. 5 and Fig. A.4 using the online inference demo hosted by Apple. We use the default parameters provided by the website for these results. InstructAny2Pix outperforms MGIE on both complex instructions involving multiple objects, such as "Remove a man and woman running across a bridge", as well as simple ones such as "turn the forest into a desert".

We further analyze the failure cases of MGIE in Fig. A.6 by making use of the "expressive instruction" generated by the online demo. These outputs reveal the underlying reasoning process of the MGIE. We observe two failure modes. For simple instructions, the model can generate the correct reasoning based on instructions, but failed to apply these edits to the image. For example, for the instruction "turn the forest into a desert", the text output of MGIE identifies relevant concepts: "little to no vegetation", "dry, lifeless branches". However, the generated image fails to respect these concepts. For more complex instructions such as "remove man and woman running across a bridge", the model fails to understand the intent, and believes the output image should "depict a man and a woman running together, likely as a couple, on a bridge", which is contrary to the instruction.

This limitation can be caused by a variety of reasons: MGIE only uses InstructPix2Pix as its pretraining dataset, whose edit instructions are less diverse than those in MM-Inst. MGIE uses a MLLM to model the *expressive edit instructions*, which contains *what should be done* to achieve desired edits. By contrast, InstructAny2Pix directly models the semantics of the intended output image, which contains *what the output should look like*. The objective of InstructAny2Pix is more straightforward. Lastly, MLLM freezes the language model itself and only trains its adaptors and edit-heads, which may limit its capabilities.

Additionally, we note that while MGIE also make use of a Multi-Modal Language Model, they do not support "multi-modal editing" in that they only accept text-only instructions. The vision encoder of MGIE is used to only process the input image, instead of multi-modal prompts like InstructAny2Pix. In summary, InstructAny2Pix outperforms MGIE on text-based edits, particularly in the presence of challenging edit prompts. InstructAny2Pix also supports more flexible instructions and multi-modal prompts, making it more preferable in most practical use cases.

D.4 Image Editing with Multi-Modal Instruction

We present additional qualitative results of multi-modal editing in Fig. A.7. The results demonstrate that InstructAny2Pix can effectively handle a diverse range of instructions involving multiple modal inputs.

E Ablation Studies

E.1 Pretraining

We experiment with three pretraining setups: no-pretraining, captioning tasks only (x-to-text), and full pretraining (x-to-text and x-to-image). We report quantitative results of image-editing task on MM-Inst-Test dataset in Table A.1. Full pretraining is required to achieve optimal performance.

Table A.1: **Ablation Study on Pretraining Strategies.** We experiment with three pretraining setups: no-pretraining, captioning tasks only (x-to-text), and full pretraining (x-to-text and x-to-image). We report results on MM-Inst-Test dataset.

	C_{dir}	C_{im}	C_{out}
No Pretraining	.071	.795	.207
Caption Only	.090	.802	.251
Full	**.099**	**.816**	**.260**

E.2 Factors affecting Image Consistency

An important goal of image editing is to ensure the edited images can reflect the intended changes while respecting the source image. There is usually a trade-off between these two goals. The most relevant hyper-parameter of InstructAny2Pix is classifier free guidance (CFG). CFG determines the degree at which the text instruction affects the generation output. We visualize edit results under different CFG in Fig. A.9. We find that CFG=5 is a sweet spot for achieving high quality edit results that follows the instructions while respecting the original image.

In addition to CFG, we can control how well the model respects the input image by adding Gaussian to the input image in the latent space. The variance of added noise is proportional to $(1-\alpha)$ where α is a hyperparameter between 0 and 1. Intuitively, when α is 1, there is no corruption to the image latent. When α is zero, the diffusion model mostly ignores the input image. We visualize this effect in Fig. A.8. For a typical use case, there is no need to corrupt the input image. We suggest setting α to 1.0.

We qualitatively evaluate the effect of these two hyperparameters on a subset of MM-Inst-Test dataset by

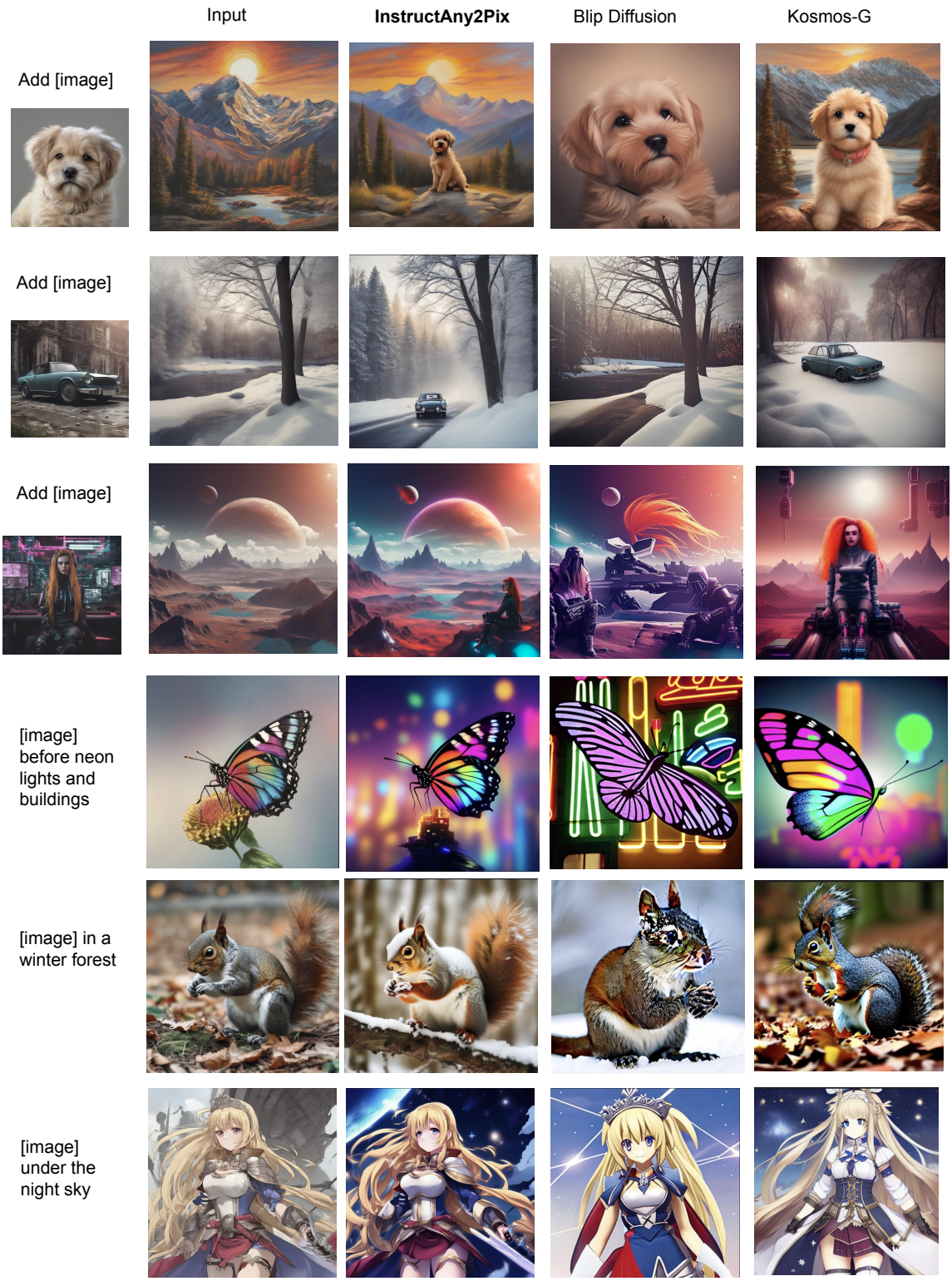

Figure A.5: **Qualitative Comparison Against Image-based Generation Methods.** We present generation results of different models under both single-image and multi-image setups. We employ multi-modal prompts from the Dreambooth++ dataset.

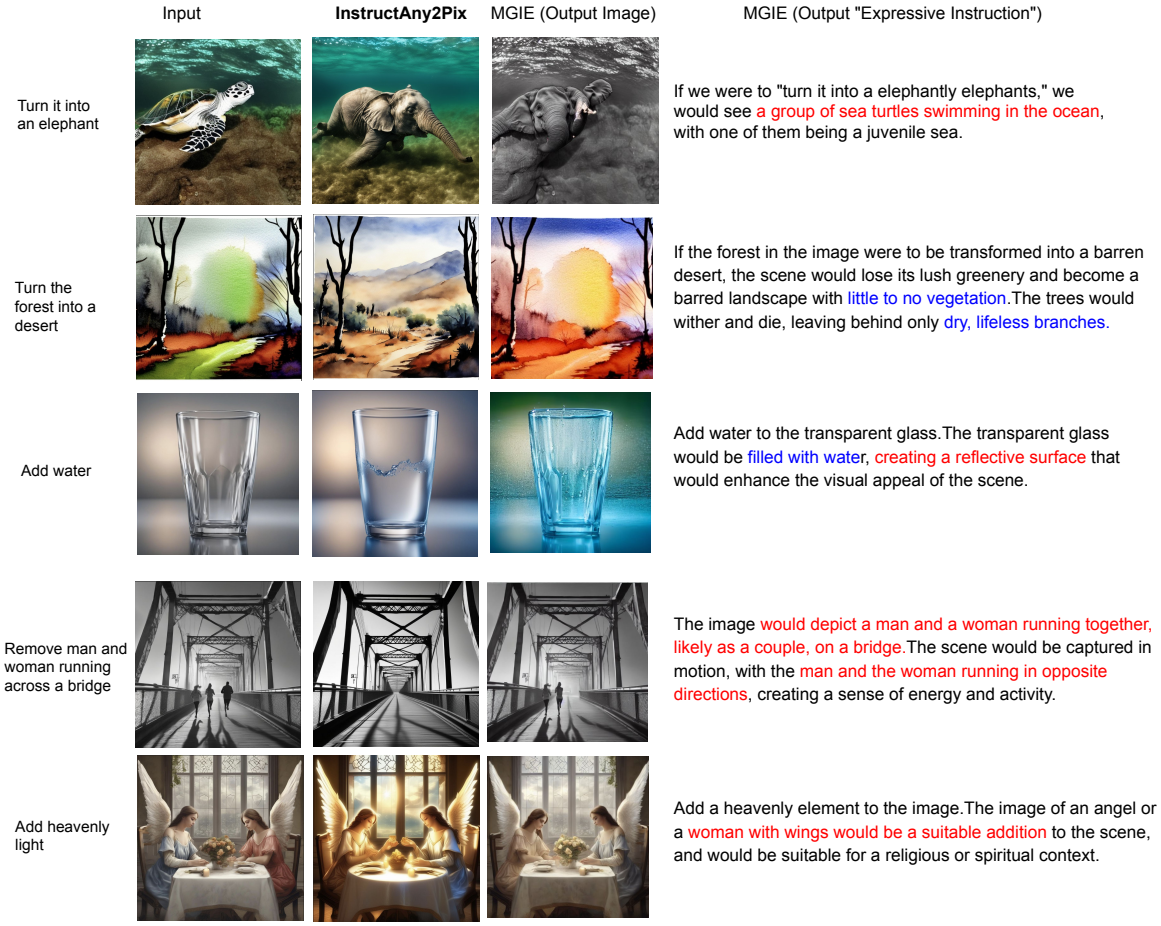

Figure A.6: **Analysis of failures of MGIE.** We show "expressive instructions" generated by MGIE alongside image outputs. These samples reveal the underlying reasoning process of the MGIE. We observe two failure modes. For simple instructions, MGIE can generate the correct reasoning based on instructions, but failed to apply these edits to the image. We mark these examples as blue. For more complex or abstract edits, MGIE fails to understand the instructions and generates wrong reasoning. We mark these examples as red. By contruct, InstructAny2Pix consistent perform the intended edits successfully.

Figure A.7: **Additional Results of Multi-Modal Editing.** We showcase qualitative results of multi-modal editing. InstructAny2Pix can handle a diverse set of instructions involving multiple modalities.

sweeping over different values of CFG and α. We report CLIP$_{out}$, which measures alignment with edit instructions and CLIP$_{im}$, which measures consistency with input images. We show these results in Fig. A.10. In general, increasing the CFG and decreasing the alpha will increase CLIP$_{out}$ and decrease CLIP$_{im}$, giving the user the flexibility to balance the instruction alignment and image consistency. We also found that removing the refinement module leads to a small drop in both metrics, highlighting its effectiveness.

Figure A.8: **Results of Varying α in the generation process.** We visualize the image generation conditioned on the CLIP embedding of the source image. As α decreases, the generation become less consistent with the source image.

Figure A.9: **Visual examples of InstructAny2Pixand baseline method under different classifier free guidance (CFG).** We compare InstructAny2Pix with InstructPix2Pix. InstructPix2Pix has two independent CFG for text and image. We abbreviate this as "t" and "i". Notably, our method generates artifact-free results in all setup, while other methods have visible artifact at all CFGs.

E.3 Factors affecting Image Quality

The refinement module is introduced as a regularization to mitigate the effect of low quality images in the data. We sample 500 generated images and evaluate the LAION Aesthetic score and PickScore (Kirstain et al., 2023), which measures image quality. Aesthetic score only considers the image quality, while PickScore additionally takes the prompt into account. We compare generations with refinement module to those without refinement module and report the results in Fig. A.11. On both metrics, adding the refinement module leads to considerable improvements.

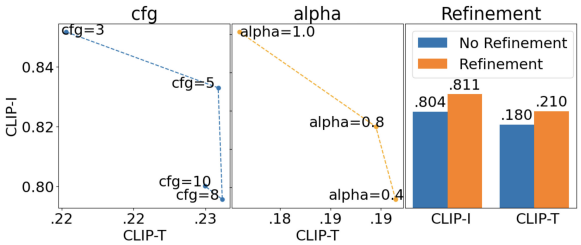

Figure A.10: **How CFG, α and the refinement module affects instruction alignment and image consistency.** Increasing the CFG and decreasing the alpha will increase CLIP$_{out}$ (CLIP-T) and decrease CLIP$_{im}$ (CLIP-I). Adding refinement module improves both metircs.

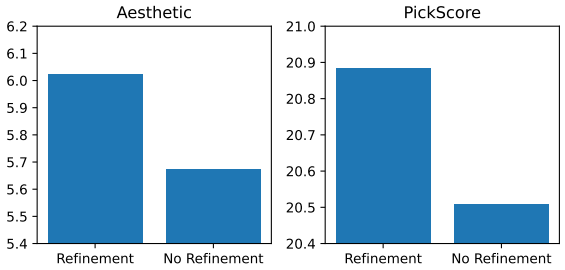

Figure A.11: **How refinement module affects image quality.** We report average Aesthetic score and PickScore (Kirstain et al., 2023) on 500 randomly sampled captions. Aesthetic scores only consider the image quality, while PickScore additionally takes the prompt into account.

F Technical Details

F.1 Model Architecture

F.1.1 Multi-Modal Encoder

We use ImageBind (Girdhar et al., 2023) as our multi-modal encoder. Particularly, ImageBind uses CLIP-ViT-L as its text and image encoder. It includes an additional audio encoder that is aligned to the representation space of CLIP-ViT-L. We use the pooled token as our multi-modal embedding. It is kept frozen throughout all training stages.

F.1.2 Diffusion Model

We use the SDXL (Podell et al., 2023) as our diffusion model. It was originally conditioned on CLIP-ViT-G and CLIP-ViT-L text features. We incorporate an MLP projection layer following (Ye et al., 2023) that maps the ImageBind embedding to the dimension of cross-attention layers. During the pretraining process, the loss is

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_t \left[\|\epsilon_t - p_\theta(z_t, C_t, C_g, C_l)\|_2^2 \right] \quad (8)$$

where p_θ is the U-Net, ϵ_t is the noise at timestamp t, z_t is the noised image latent sampled in the forward

diffusion process at time t, C_t is the CLIP embedding of captions. C_g is the embedding of the whole image, C_l is the embedding of a cropped region. The cropped region is sampled from an object detector or a uniform distribution of bounding boxes at a 1:1 ratio. At each training step, C_t, C_g, C_l are randomly dropped independently with a probability of 0.2.

Diffusion Model is not used in the instruction fine-tuning stage.

F.1.3 Refinement

We adopt a decoder-only transformer with 24 layers and a hidden size of 1024. We also incorporate an MLP projector that maps ImageBind features to the hidden dimension of the transformer. Another MLP projector is used to map the output of the transformer back to the dimension of ImageBind features.

F.1.4 Multi-Modal Large Language Model

We use Vicuna-7B (Team, 2023) as our base model. We made no additional changes to the LLM architecture other than adding input and output projectors, which are two two-layer MLPs. The input projector maps the embedding of multi-modal encoder to the dimension of MLLM's hidden states. The output project maps the extracted hidden states from the MLLM to the dimension of encoder embeddings.

F.1.5 Parameter Count

We report the total number of parameters in each module in Table A.2. In total, our model has around 10B parameters.

Table A.2: **Number of Parameters in InstructAny2Pix**. We report the total number of parameters in each module.

	Params
LLM	7B
SDXL	2.5B
Refinement	71.1M
Projectors	62.9M

F.2 Training Dataset

F.2.1 Paired Training Data

We use SoundNet (Aytar et al., 2016), VGG-Sound (Chen et al., 2020), and AudioSet (Gemmeke et al., 2017) for image-audio pairs. 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 2 million audio-image pairs, and VGG-Sound consists of 197,958 pairs. Particularly, out of 2 million videos in AudioSet, 1 million are under the music category. These videos can be music videos, concerts, documentaries, and other kinds of videos that use music as background, such as news programs.

We also make use of audio captions from MusicCaps (Agostinelli et al., 2023) and AudioCaps (Kim et al., 2019) 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 (Schuhmann et al., 2022) 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 the alignment process.

F.2.2 Instruction Tuning Dataset (MM-Inst)

As described in Sec. 3.3.1 of the main paper, MM-Inst was generated in three steps. We show this in Fig. A.12.

In the first step, we prompt a large language model (Llama 2) to generate creative instructions. Each instruction contains the caption of the input image, the caption of the output image, the text instruction and optionally captions of reference images and audio. We also ask the model to explicitly mark objects that need to be added to the scene. This information is used later in stage 3 to generate input-output image pairs.

Example instructions include adding, dropping, removing, or replacing objects as well as other free-form instructions. To generate a diverse set of image editing instructions, we prompt the language model to create editing instructions based on captions of sampled LAION images. We use BLIP-2(Li et al., 2023b) to generate these captions instead of using the captions provided in the dataset, because the provided captions are noisy alt text that is not natural English. To further increase the diversity of audio-related instructions, we further prompt the language model to generate instructions involving ground-truth music captions from LP-MusicCaps and AudioCaps. We observe that without this step, the language model tends to only generate simple audio captions such as "sound of water" or "sound of rain" and fails to incorporate complex descriptions of music. In total, we generate 500k instructions.

In the second step, we collect or generate reference images and audios using captions created in the first step. There are two types of captions. The first type appears in instructions generated by explicitly prompting the LLM with ground truth captions of music, sound, and images. For these captions, the corresponding media can be directly used. The second type is generated solely by the LLM. We use AudioLDM2 (Liu et al., 2023b) to generate audio and music, and use SDXL to generate images. We generate 5 samples for each caption and use CLAP (Wu* et al., 2023) and CLIP (Radford et al., 2021) to find the samples that align best with the caption.

In the last step, we employ a diverse set of methods mentioned in Sec. 3.3.1 to create input-output image pairs.

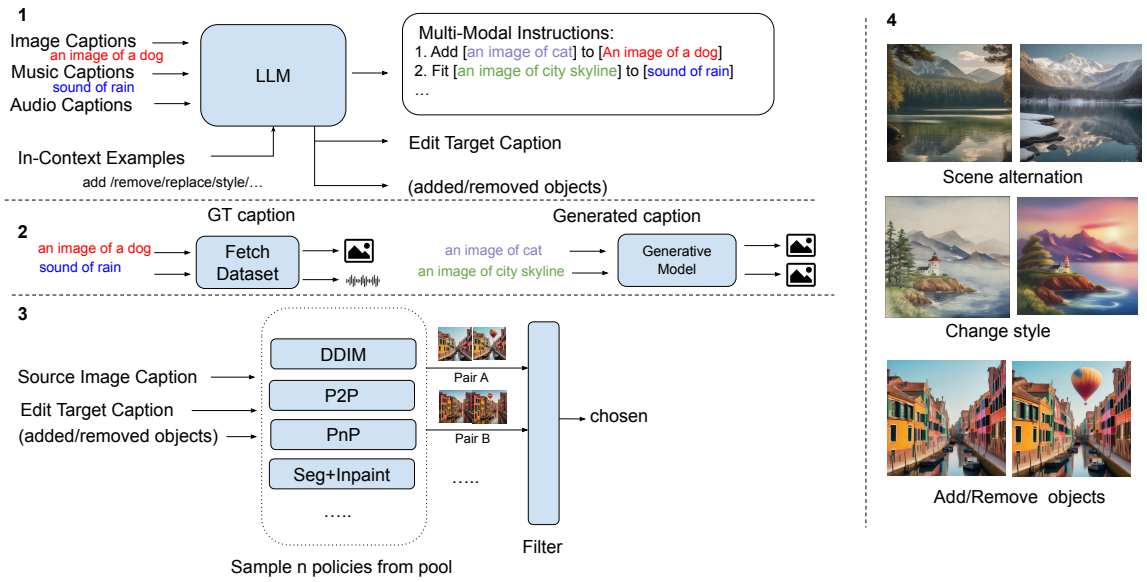

Figure A.12: Data generation Pipeline. 1. We prompt LLM with sampled captions and examples to generate a diverse set of instructions. 2. We obtain reference music, audio, and images by either generating them using SDXL and AudioLDM2. If the caption corresponding to a ground truth caption, we directly fetch the corresponding media. 3. We employ a mixture of methods to generate candidate image pairs for each instruction and filter them using CLIP score. 4. We show some example image pairs in the filtered dataset on the right.

Compared with the InstructPix2Pix dataset, MM-Inst has the following advantages. First, it uses BLIP-generated captions which are grounded with real images instead of the raw caption from LAION, which can be very noisy. Second, it uses a variety of techniques to generate paired data instead of solely relying on Prompt2Prompt. In particular, we observe that the segmentation+inpainting pipeline generates high-quality results for object removal when the segmentation model can correctly localize the target object. Third, we filter the data using both CLIP and Aesthetic score and consider both prompt alignment and generation quality. In contrast, InstructPix2Pix only uses CLIP score as the filtering mechanism. Lastly, MM-Inst incorporates multi-modal inputs, which makes it uniquely suitable for our multi-modal editing tasks. We showcase these differences in Table A.3. We provide examples from both datasets in Table A.4.

F.3 Evaluation Dataset

F.3.1 MM-Inst-Test

To generate a diverse set of prompts, we ask the MTurk workers to generate creative edit instructions using different captions sampled from LAION. We also require the MTurk workers to generate different types of edits for the same caption. We do not generate ground truth target images. For reference images and audio pieces used in the instructions, we use SDXL (Podell et al., 2023) to generate images and AudioLDM2 to generate (Liu et al., 2023a) audio. Results are filtered by CLIP(Radford et al., 2021) score for images and CLAP(Wu* et al., 2023) score for audios.

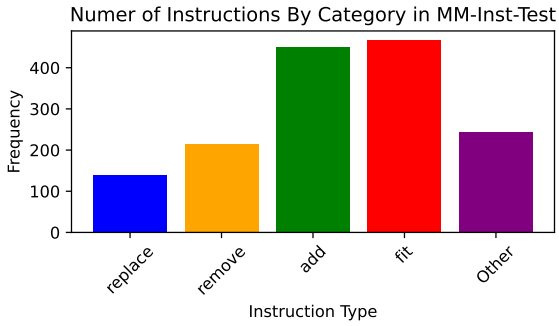

Figure A.13: **Distribution of Instructions in MM-Inst-Test dataset.** We show the number of instructions. "Fit" refers to combining music and image, fitting an image to the style of another image, or other organic ways of combining different modalities together. "Others" include all instructions that cannot be classified as other categories, such as "transform [image] into a night scene with the sound of [sound]".

One of the challenges in generating the test dataset is that there are instances of bad format and low quality. We prompt each MTurk worker to generate five different edit prompts for each caption. After a batch is collected, we manually identify the problematic instructions and redistribute them to a new set of workers. In order to reach 1,500 valid instructions, we collect a total of 1795 instructions. We show the distribution of different types of instructions in Fig. A.13.

F.3.2 Dreambooth++

Dreambooth is commonly used to evaluate generative models that support image prompts. However, its evaluation protocol consists of only a single reference image and a text prompt. It is not suitable to evaluate models that can take multiple reference images, such as Kosmos-G (Pan et al., 2023) and InstructAny2Pix. Moreover, the classes have limited diversity and only have two live subjects (cats and dogs). These two live subjects accounts for 9 out of 30 subjects. All other objects are small, still objects, such as backpacks. There are no medium-to-large objects such as trees or bicycles. To provide a diverse and fair comparison, we propose Dreambooth++, which contains 30 subjects and 30 prompts. In total, there are 900 combinations. We use SDXL to generate all subject images. For each prompt, we also generate a background.

We conduct evaluation on two benchmarks: The single-image setup is similar to Dreambooth, which requires generating a given subject under different context prompts. The multi-image task requires generating a new image by combining a subject image and a background image. For the multi-image task, we use a segmentation model to localize the edited objects in the new image. We report DINO similarity of the cropped subject with the reference subject image as DINO_{sub} and the DINO similarity of the generated image with the reference background image as DINO_{ref}. We also report the recall rate of the segmentation model. These results are listed in the main paper Table 2.

We show the full list of subjects in Table A.5 and compare it with Dreambooth. DreamBooth++ offers a more diverse range of subjects ranging from animals, humans, large structures, and small items.

F.4 Compute

We use AdamW optimizer with a learning rate of 1e-6. We use 8 Nvidia A6000 GPU for our experiments. We train the model for 2 epoch, which takes around two days. The diffusion model is trained on 8 A5000 GPU with AdamW optimizer, a learning rate of 1e-6 for 4 days (40000 steps).

G Error Bars

We report the margin of error of 95% confidence interval of main results in Table A.6 and Table A.7.

H Border Impacts

InstructAny2Pix aims to improve the flexibility of image editing models by incorporating multi-object, multi-modal prompts. In particular, InstructAny2Pix uniquely enables a set of creative music-based applications such as music-inspired design. However, just as any other image-edit models, InstructAny2Pix can be used to for fraud and deception. Particularly, the ability to synthesize multiple images using text instructions can be used to create fake, deceptive images with high quality. Hence, it is important to employ guardrails when deploying InstructAny2Pix to end-user products.

I Safe Guards

InstructAny2Pix is based on the diffuser (von Platen et al., 2022) library. It should be used with the standard safeguards, including NSFW safety checker and hidden watermarks.

J License

We makes use the following models: CLIP (MIT license), PickScore(MIT license), LAION Aesthetics predictor (MIT license), SDXL(CreativeML Open RAIL++-M License), LLAMA 2 (Llama 2 Community License Agreement), Vicuna (Apache2 license). BLIP-2 (BSD-3-Clause license)

We use the following dataset SoundNet (MIT license), VGG-Sound (CC BY 4.0 license), AudioSet (CC BY-SA 4.0 license), MusicCaps (CC BY-SA 4.0), AudioCaps (MIT License) LAION (MIT License).

LAION dataset is currently unlisted publically due to a safety review.

K Human Instructions

In this section, we report the instructions used to generate human evaluation results and image captions.

The following prompt is used to generate the caption

Instruction to Write Prompts for MM-Inst-Test Dataset

Your goal is to generate multimodal image edit instructions including a source image caption, an edit instruction, and a target image caption. The edit instruction can involve other reference image and audios. Some concrete examples are
1.add [image:fireworks] to [image:base:a city skyline at night] == [image:result:an image of a city skyline at night with fireworks]
2.remove [audio: water stream] from [image:base:a painting of a cabin by the lake at sunset] == [image:result:a painting of a cabin by a corn field at sunset]
3.fit [image:base:an image of empty street] to [audio: upbeat electronic music]==[image:result:an image of vibrant city street]
4.fit [image:base:a city skyline] to the style [image:an impressionist painting] == [image:result:an impressionist painting of a city skyline]
5. replace [image:people] with [image:wildlife] in [image:base:a painting of two people standing in a field surrounded by hay bales] == [image:result:a painting of

Table A.3: Comparison of MM-Inst dataset and Instruct Pix2Pix dataset.

	MM-Inst	MM-Inst-Test	InstructPix2Pix
Caption Source	LAION-Aesthetics	LAION-Aesthetics	LAION-Aesthetics
Input Caption Generation	BLIP2	BLIP2	Noisy WebData
Instruction Generation	Llama 2	Human	GPT-3
Paired Data Generation	DDIM	-	Prompt2Prompt
	Prompt2Prompt	-	
	Plug-and-Play	-	
	Segmentation+Inpaint	-	
Filtering	Multiple Metrics	Human	CLIP
Modality	Image,Text,Audio	Image,Text,Audio	Text
Size	500,000	1,500	313,010

Table A.4: **Examples of Instructions from MM-Inst dataset and Instruct Pix2Pix dataset.** Captions are marked by [.]. MM-Inst offers a better set of captions and instructions.

MM-Inst	
Instruction	Output
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 [sound of car accelerating] from [an image of people driving in the countryside road]	An image of a quiet countryside road
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
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
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

InstructPix2Pix	
[misurina XIII... by roblfc1892] have it be a stamp	Stamp... misurina XIII... by roblfc1892
[Manarola during sunrise - Cinque Terre] it is foggy	Manarola during foggy sunrise - Cinque Terre

Table A.5: **Subjects of DreamBooth++ and DreamBooth Dataset.** DreamBooth++ offers a more diverse range of subjects ranging from animals, humans, large structures, and small items.

DreamBooth++ Dataset Subjects	**DreamBooth Subjects**
a cute cat	backpack
a cute dog	backpack_dog
a colorful butterfly	bear_plushie
a colorful bird flying low over a body of water	berry_bowl
spotted horse	can
an image of a squirrel	candle
an african elephant walking through a grassy field	cat
an image of cute anime girl	cat2
an anime princess with long blonde hair and two swords	clock
the woman in a black dress holding a fan	colorful_sneaker
a man standing at a podium	dog
a cyberpunk style	dog2
an image of a scientist	dog3
an image of an astronaut	dog5
an artistic painting of a woman with blonde hair	dog6
a wooden bridge	dog7
car	dog8
traffic lights	duck_toy
train	fancy_boot
tree	grey_sloth_plushie
bicycle	monster_toy
an image of a robot	pink_sunglasses
tablet	poop_emoji
telescope	rc_car
the skull	red_cartoon
vase	robot_toy
wand	shiny_sneaker
an image of chair	teapot
an image of empty glass	vase
a cowboy hat	wolf_plushie

Table A.6: Multi-Modal Image Editing on MM-Inst-Test Dataset and Text-based Image Editing on InstructP2P Dataset.The best number is **bolded** and second-best is underlined.

	MM-Inst			InstructP2P		
	CLIP$_{dir}$	CLIP$_{im}$	CLIP$_{out}$	CLIP$_{dir}$	CLIP$_{im}$	CLIP$_{out}$
Ours(T)	**.095±.003**	.856±.001	**.270±.002**	.147±.003	.808±.003	**.312±.002**
InstructP2P	.091±.003	.824±.002	.243±.002	.145±.003	.742±.004	.241±.002
MagicBrush	.084±.004	.807±.006	.199±.002	**.165±.004**	.760±.006	.250±.001
InstructDiff.	.066±.003	**.940±.004**	.193±.002	.126±.002	**.857±.003**	.301±.002

wildlife in a field surrounded by hay bales] The typical examples can be adding, removing, replacing objects, image style transfer or fitting image to audio. However, you are encouraged to be creative. use [audio:xxx] to mark audio inputs, use [im-

Table A.7: Image Conditioned Generation on DreamBench++ Dataset. C_{dir}, C_{im}, C_{out} is abbreviated form of CLIP$_{dir}$, CLIP$_{im}$ and CLIP$_{out}$. For multi-image setup, numbers are reported in DINO$_{ref}$/DINO$_{sub}$ format. The best number is **bolded** and second-best is underlined.

	Single-Image			Multi-Image		
	C$_{dir}$	C$_{im}$	C$_{out}$	C$_{dir}$	C$_{im}$	C$_{out}$
Ours(T+I)	**.147±.004**	.810±.004	**.260±.002**	.154±.004	**.789±.004**	**.309±.002**
BLIP-Diffusion	.089±.005	.779±.005	.231±.002	.091±.005	.701±.005	.292±.002
Kosmos-G	.126±.005	**.843±.005**	.251±.002	**.166±.005**	.740±.005	.286±.002

age:xxx] to mark reference image inputs. use[image:base:xx] to specify source image,use [image:result:xxx] to specify the results use == to separate instruction and results.
IMPORTANT : The Removed or Replaced Object should exist in the Orignal Source Image. Special Instructions for removal/replace commands.
When removing objects, do try to remove objects not explictly mentioned in the caption
When removing objects, do NOT try to use words like xxx without yyy, simply drop the removed objects Examples: Remove [audio:piano music] from [image:base:an image of a room with piano]
Good Example 1: [image:result:an image of an empty room]
Good Example 2: [image:result:an image of a room]
Bad Example : [image:result:an image of a room without piano]

The following instruction is used to collect human feedbacks:

Instruction for Human Evaluation

Task Description
Your task is to compare the output images from two image edit models based on the following criteria:
Alignment: Does the model follow the instruction accurately?
Quality: How good is the model generation output?
Information Loss: How well does it respect the original inputs?
Note: Some models may crop the source image at the center. Please do not consider cropping as a factor in your judgment.
[images and instructions here]
Given the criteria, which of the edit output among Image A and Image B is better?