

Unleashing In-context Learning of Autoregressive Models for Few-shot Image Manipulation

Supplementary Material

This is the supplementary material for the publication titled “Unleashing In-context Learning of Autoregressive Models for Few-shot Image Manipulation”. We organize the content as follows:

A – Defects of CLIP-I as a Metric

B – Additional Experiment Results

B.1 – Analysis on the Number of Manipulation Tokens

B.2 – Manipulation with the Same Textual Instruction and Different Exemplar Images

B.3 – Comparison with the Generic Autoregressive Model

B.4 – Reverse Image Transformation

B.5 – Object Addition and Removal

B.6 – Additional Visualization

B.7 – Failure Cases

C – Implementation Details

C.1 – Establishment of Test Set

C.2 – Training Details of Our Model

C.3 – Implementation of Previous Methods

C.4 – Details of User Study

D – Limitation and Future Work

E – Code and Data Release

F – Video Demonstration

A. Defects of CLIP-I as a Metric

In Sec. 4.2 of the main paper, we argue that CLIP-I (similarity between the query image and the manipulated image) has inherent defects when used as a metric for image manipulation. In order to further explain the reason, we calculate the four CLIP-based metrics used in our experiments (CLIP-Dir, CLIP-Vis, CLIP-T, CLIP-I) on the outputs of three models and the ground truth, which is shown in Fig. 1.

Compared with InstructPix2Pix [1] and PromptDiffusion [6], our model follows the textual and visual guidance more faithfully in this instance. Nevertheless, this advantage is not correctly reflected by the CLIP-I metric. InstructPix2Pix conducts a trivial modification to the query image, thus resulting in a high similarity between the query image and the output. It’s worth noting that the CLIP-I score of InstructPix2Pix is even higher than the score of the ground truth. In contrast, PromptDiffusion overly edits the query image, leading to a CLIP-I score lower than InstaManip and ground truth. Our model (which has the best performance) and ground truth have medium CLIP-I scores between InstructPix2Pix and PromptDiffusion. This example suggests that a higher or lower CLIP-I score does not necessarily correspond to a better performance in the image manipulation task. Hence, it’s hard to accurately compare the

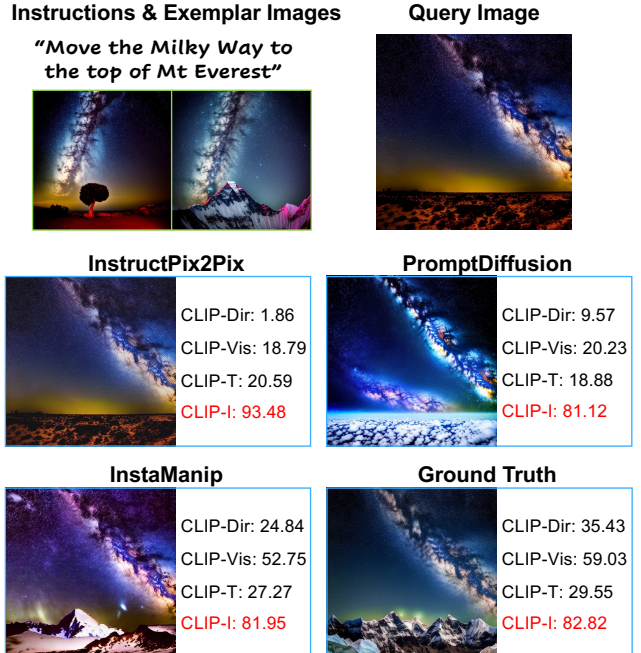


Figure 1. Comparison of the four CLIP-based metrics on the outputs of three models and the ground truth. CLIP-I is highlighted in red. Please refer to Sec. A for the explanation.

# Manipulation Tokens	CLIP-Dir	CLIP-Vis
10	18.24	29.87
20	19.07	31.10
30	19.81	32.39
40	19.74	32.21
50	19.66	32.20

Table 1. Analysis on the impact of the number of manipulation tokens. The orange row indicates our final model. Please refer to Sec. B.1 for the explanation.

performance of two methods based on CLIP-I alone. Fortunately, the other three metrics correctly discriminate the performance of the three models, so we use them as the primary metrics in our experiments.

B. Additional Experiment Results

B.1. Analysis on the Number of Manipulation Tokens

We implement experiments to validate the impact of different numbers of manipulation tokens. Tab. 1 shows that the

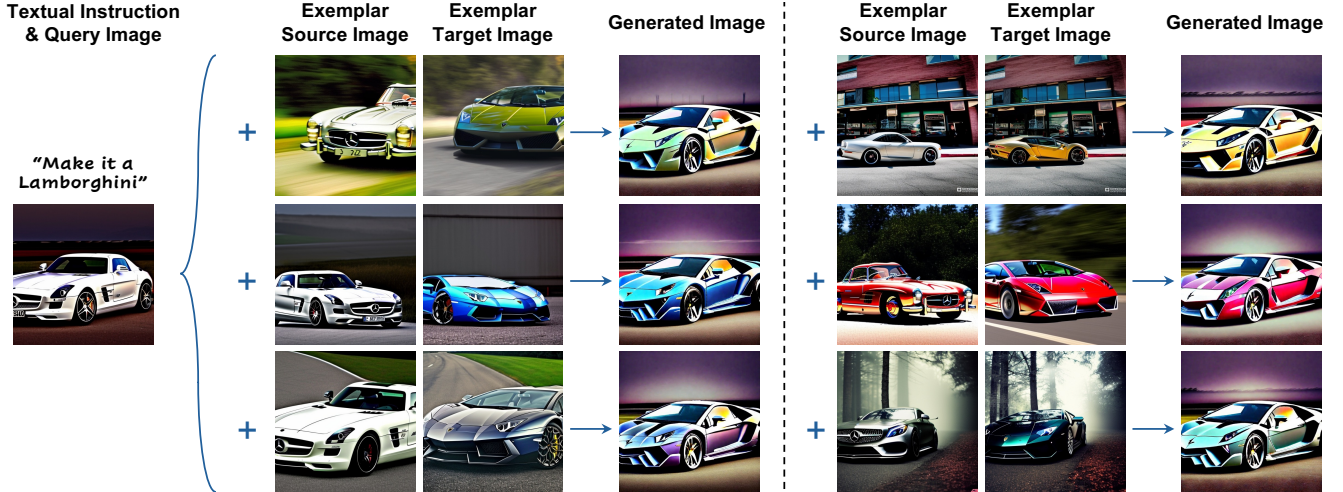


Figure 2. The visualization of manipulating the query image using the same textual instruction, yet different visual examples. When we use exemplar target images of Lamborghini with different colors, our model successfully captures this local feature from the visual guidance, and changes the colors in the generated images accordingly. Please refer to Sec. B.2 for the detailed analysis.

performance is boosted by increasing the number of manipulation tokens from 10 to 30. If more than 30 tokens are used in our model, the performance remains comparable to that observed with 30 tokens, suggesting that the model has reached a saturation point. Consequently, we set the number of manipulation tokens as 30 in the final InstaManip model.

B.2. Manipulation with the Same Textual Instruction and Different Exemplar Images

One benefit of using exemplar images in image manipulation is that the images effectively convey the desired local details to the model, which may be missing in textual instructions. To validate if the proposed model can effectively learn the visual features, we apply our model to a given image using the same textual instruction yet different visual examples. The results are illustrated in Fig. 2. In this experiment, we use different exemplar pairs following the same textual instruction. The major difference of these examples is the color of the Lamborghini in the exemplar target images. Our model learns this visual feature and successfully edits the query image using similar colors, which exactly reflects the advantage of few-shot image manipulation.

B.3. Comparison with the Generic Autoregressive Model

In this paper, we propose an autoregressive model with enhanced in-context learning capability for few-shot image manipulation. Prior to our work, there is some work about using the autoregressive architecture as a generic in-context learner for various tasks. Emu2 [4] is one of the recent studies in this field, showing awesome performance in visual understanding and image generation problems. We compare our model with Emu2 on few-shot image manipula-

Methods	Guidance	CLIP-Dir	CLIP-Vis	CLIP-T	CLIP-I
<i>In Distribution</i>					
Emu2 [4]	Text + Image	15.26	24.64	27.02	76.89
InstaManip	Text + Image	19.81	32.39	27.72	80.11
<i>Out of Distribution</i>					
Emu2 [4]	Text + Image	14.09	21.65	20.17	65.80
InstaManip	Text + Image	18.27	28.23	26.81	79.71

Table 2. Comparison with Emu2. InstaManip outperforms the generic autoregressive model by a great margin. Additional discussions are shown in Sec. B.3.

tion. The results are reported in Tab. 2. InstaManip greatly surpasses Emu2 across all metrics in both evaluation settings. Despite the existence of generic in-context learners, the result suggests that few-shot image manipulation is still a challenging problem that requires specific novel model design. It also validates the necessity of investigating how to improve in-context learning performance for specific tasks like our work.

B.4. Reverse Image Transformation

We further validate the few-shot learning capability of InstaManip by reverse image transformation. Specifically, we swap the exemplar source and exemplar target images in the prompt, to test if the model can learn a reverse transformation embedding to transfer an edited image back to the source image. The textual instructions are also rephrased accordingly. The results are illustrated in Fig. 3. Our model successfully reverses the transformation of Lamborghini and Van Gogh style. This experiment provides more evidence of the robustness and adaptability of our model.

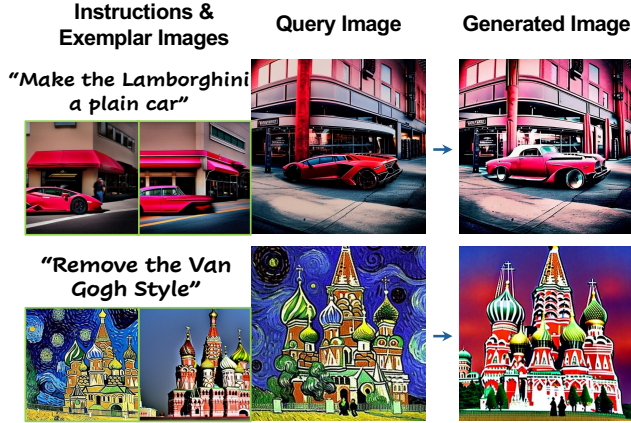


Figure 3. Examples of reverse image transformation. We swap the positions of exemplar source and exemplar target images, and use the target image as query in the prompts. InstaManip is able to learn the reverse image transformation and transfer the target images (*i.e.*, Lamborghini, Van Gogh painting) back to the source images (*i.e.*, plain car, regular painting). Please refer to Sec. B.4 for more details.

B.5. Object Addition and Removal

We find operations of object addition and removal are rare in the dataset of InstructPix2Pix [1]. To further assess the capability of InstaManip, we show two examples of adding and removing a specific object in Fig. 4. In this experiment, our model shows decent performance in the two object-level editing operations. Meanwhile, we also observe some undesired distortion (in object removal) and context change (in object addition) in the output images. We think involving more training data of object-level editing can further improve the performance of our model.

B.6. Additional Visualization

To further demonstrate the performance of the proposed InstaManip, we illustrate more outputs from our model in Figs. 5 and 6. By learning an explicit manipulation embedding, InstaManip successfully captures the underlying image transformations from textual and visual guidance, and implements them to the query images faithfully.

B.7. Failure Cases

Though InstaManip shows strong in-context learning capability in image manipulation, we still find it may fail in some cases, as presented in Fig. 8. To begin with, our model still struggles with the big domain gap between the exemplar images and the query image. In the first example of Fig. 8, the exemplar images show a view of mountains with plants, while the query image is a picture of a cook preparing meals. Our model places the fireworks in an incorrect position in the generated image. In addition, our model is very likely to fail if the exemplar images do not show the

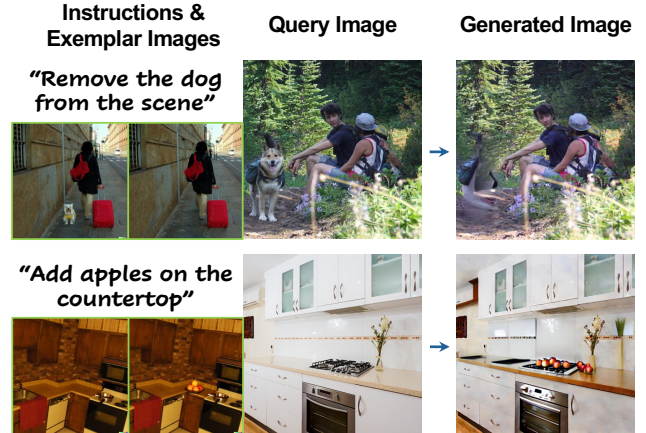


Figure 4. Examples of object addition and removal. InstaManip is able to learn the operation of adding or removing an object through visual examples, and edit the query image precisely. More explanations are shown in Sec. B.5.

desired visual features accurately. In the second example, the exemplar target image does not show the shape, structure and texture of pterodactyl clearly, thus misleading our model into making a random transformation to the query image. In the third example, the saxophone has a complex structure and texture. Our model fails to accurately capture these subtle details in the generated image. These weaknesses can motivate future investigations into novel models with stronger in-context learning capability. Please refer to Sec. D for more discussions.

C. Implementation Details

C.1. Establishment of Test Set

In order to test our model on unseen instructions, we establish the test set based on selected keywords. Specifically, we count the occurrence of each word in the InstructPix2Pix dataset [1], and select 30 keywords with low occurrence. The 30 keywords include “boxing”, “cage”, “carousel”, “catgirl”, “Chihuahua”, “clay”, “devil”, “Everest”, “firefighter”, “firework”, “hoodie”, “joker”, “kayak”, “Lamborghini”, “Lego”, “Monet”, “plaid”, “pterodactyl”, “rainbow”, “saxophone”, “sepia toned”, “solar eclipse”, “toddler”, “toucan”, “tower of pisa”, “tropical”, “tundra”, “turtleneck”, “Van Gogh” and “wildflower”. We check out each instance of these keywords manually to filter out low-quality data and incorrect ground truth. The remaining data is used as the test set. We also exclude all instructions that contain any of these selected keywords from the training data, to make sure none of the models is optimized on these keywords in the experiments. Finally, we end up with 325 instructions and 1296 data samples in the test set.

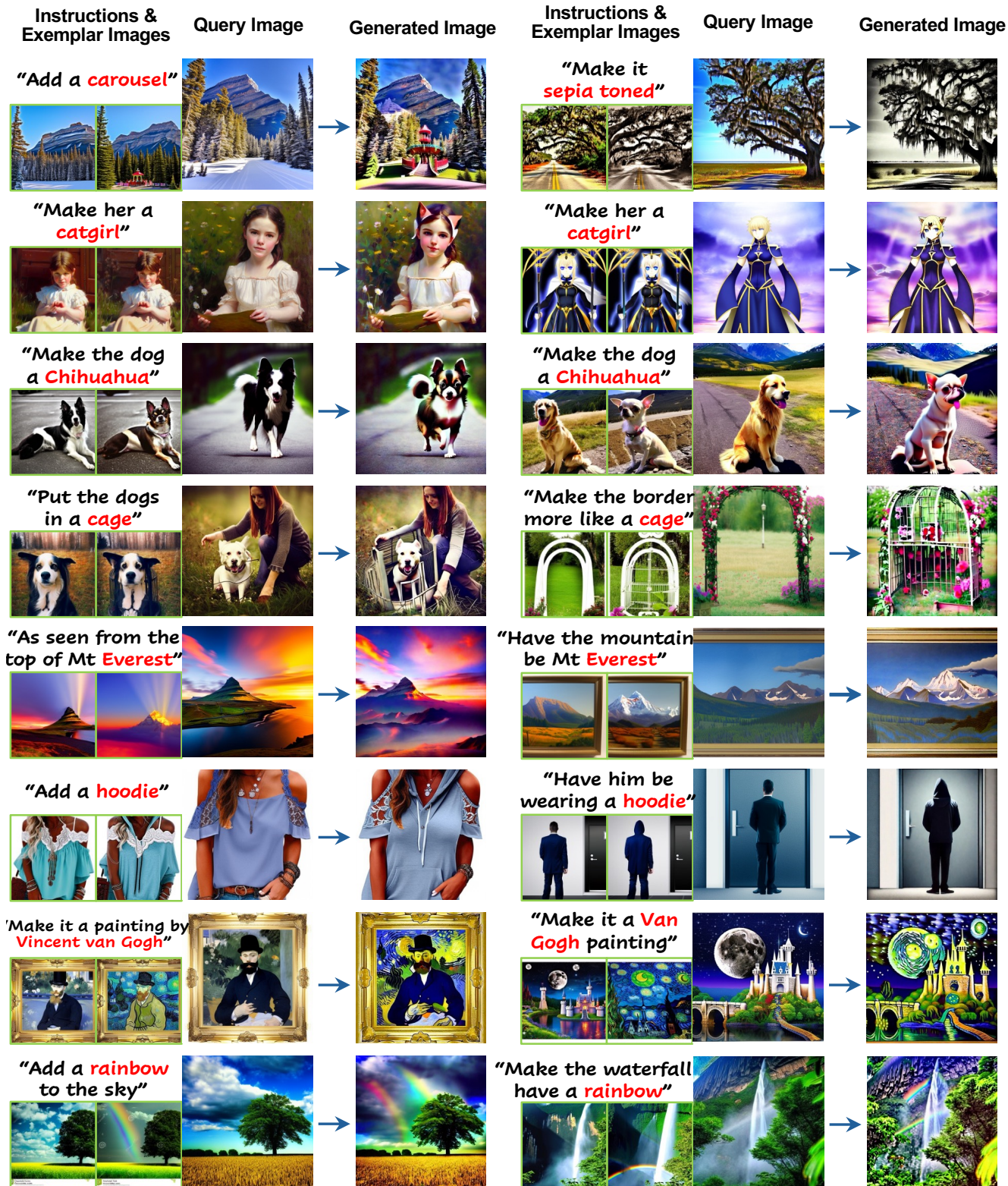


Figure 5. Additional visualization of the output from InstaManip. All instructions containing selected keywords (highlighted in red) are excluded from the training set. Our model learns unseen image manipulation operations from both textual and visual guidance, and applies the learned transformations to the new query images. More examples are presented in Fig. 6. See Sec. B.6 for the discussions.

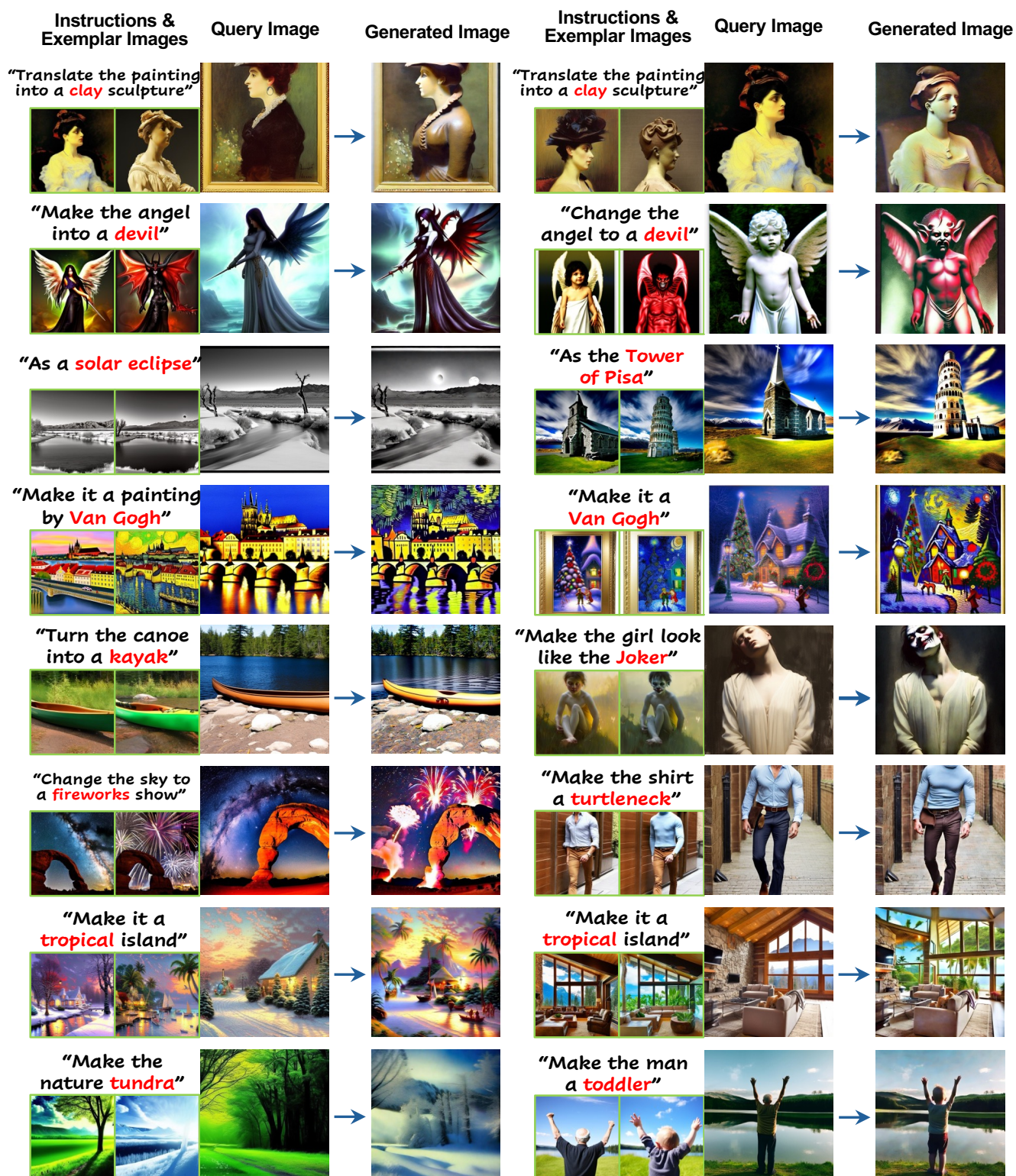


Figure 6. More demonstration of the output from InstaManip (continuation of Fig. 5). All instructions containing selected keywords (highlighted in red) are removed from the training set. Our model edits the query image aligned with both textual instructions and exemplar images. See Sec. B.6 for the discussions.

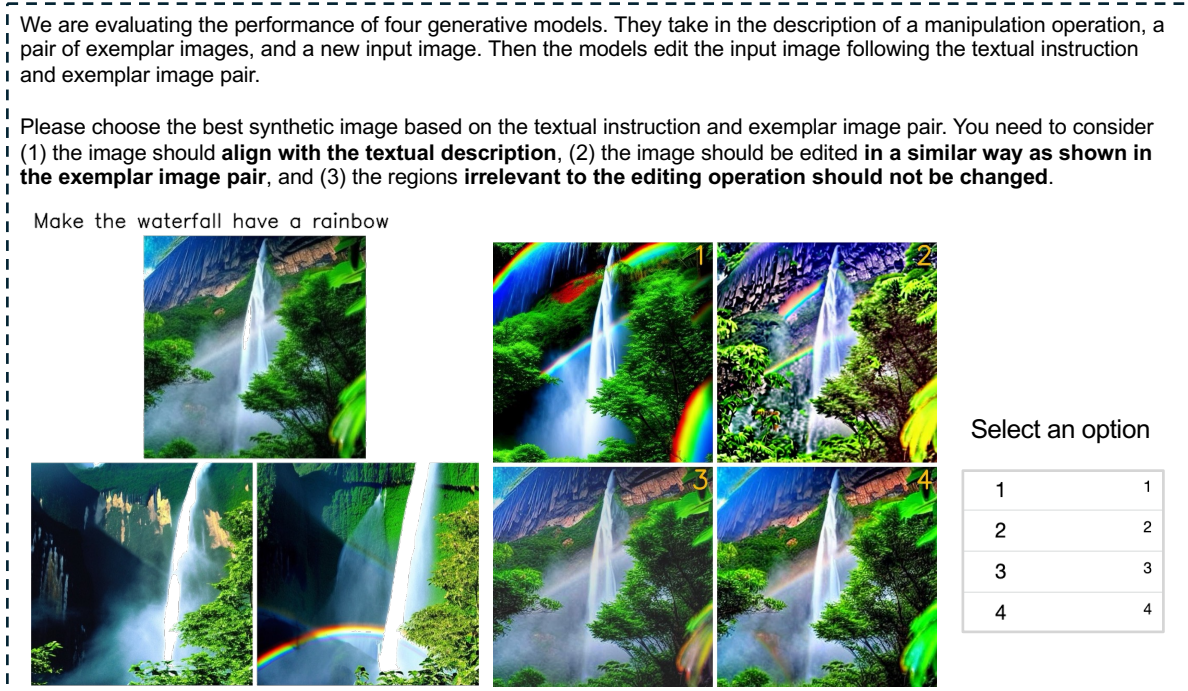


Figure 7. The interface used for human evaluation. The four manipulated images are randomly shuffled to avoid potential bias. Please refer to Sec. C.4 for the detailed elaboration.

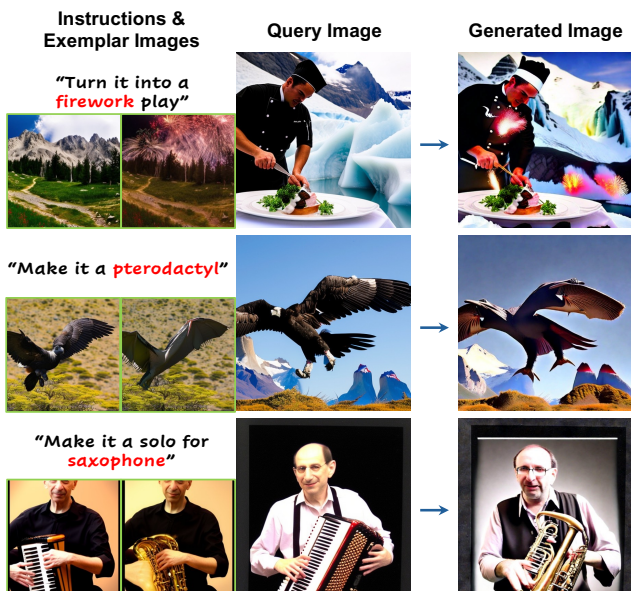


Figure 8. Failure cases of InstaManip. Please refer to Sec. B.7 for the discussions.

C.2. Training Details of Our Model

We interpolate the images to a resolution of 448×448 before forwarding them to the image encoder. We train our model using the AdamW optimizer [2] for 20000 iterations on 8 GPUs of NVIDIA A100-SXM4-80GB for 6 days. The

batch size is set as 480. We warm up the model to a learning rate of 10^{-4} in the first 500 iterations, and reduce the learning rate by cosine annealing in the remaining steps. The weight decay, β_1 and β_2 of AdamW are set as 0.05, 0.9 and 0.98 respectively.

C.3. Implementation of Previous Methods

InstaManip is compared with four models in the main paper Sec. 4.2: InstructPix2Pix [1], ImageBrush [5], VISII [3] and PromptDiffusion [6]. As a baseline of text-guided image editing model, InstructPix2Pix is trained only with textual instructions. The model weights are also used for VISII, which relies on a pre-trained InstructPix2Pix model for test-time finetuning. We freeze the weights of InstructPix2Pix and finetune a learnable instruction embedding for each test instance as described in the VISII paper. In contrast, ImageBrush and PromptDiffusion can be trained in an end-to-end way. We train the two models on our training set following the default hyperparameters specified in their work. For a fair comparison, we use both textual instructions and visual examples for VISII, ImageBrush and PromptDiffusion.

C.4. Details of User Study

We implement human evaluation across our model and the three prior few-shot image manipulation models in the main paper Sec. 4.2. We sample 100 examples from the test set

for evaluation. For each sample, we show the textual instruction, exemplar images, query image and the outputs from the four models to human raters. The raters are asked to select the best output image based on three criteria: (1) alignment with the textual instruction, (2) alignment with the exemplar image pair and (3) preservation of irrelevant regions. Each instance is evaluated by six raters. The human evaluation is conducted on Amazon Mechanical Turk. The interface is illustrated in Fig. 7.

D. Limitation and Future Work

In this paper, we propose a novel autoregressive architecture to model the learning stage and applying stage separately in in-context learning. Despite the superiority over existing approaches, we still find there are some problems that are not solved by our model. Our model suffers from an obvious performance drop when there is a big gap between the query image and exemplar images. Learning a new object with complex textures is also challenging. Our model may fail to fully capture the subtle details in the visual examples. The failure cases and analysis are elaborated in Sec. B.7.

In addition to the limitation, our work also points out several valuable research directions.

- Addressing cases with significant gap between the query image and visual examples is crucial for real-world applications. Innovative approach for this problem and large datasets containing such out-of-distribution examples are required in future studies.
- The dataset used in our work provides four instances at most for each instruction, which prevents us from exploring the saturation point of our model capability by using more than three exemplar pairs in the experiments. More efforts are demanded to build a dataset specifically for few-shot image manipulation.
- While our model has shown strong in-context learning capability on image manipulation problem, how to exploit our method for other problems remains to be explored. We expect more future investigations of our findings for stronger generic in-context learning across various tasks.

E. Code and Data Release

We have released our training and evaluation code, model weights and train/test split to the research community to facilitate future studies. Please check out our project page: <https://bolinlai.github.io/projects/InstaManip/>.

F. Video Demonstration

We also provide a video in the supplementary materials to present our work. Please watch the video for the demonstration and narration of our method.

References

- [1] 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, 6
- [2] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017. 6
- [3] Thao Nguyen, Yuheng Li, Utkarsh Ojha, and Yong Jae Lee. Visual instruction inversion: image editing via visual prompting. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 9598–9613, 2023. 6
- [4] Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14398–14409, 2024. 2
- [5] Yasheng Sun, Yifan Yang, Houwen Peng, Yifei Shen, Yuqing Yang, Han Hu, Lili Qiu, and Hideki Koike. Imagebrush: learning visual in-context instructions for exemplar-based image manipulation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 48723–48743, 2023. 6
- [6] Zhendong Wang, Yifan Jiang, Yadong Lu, Yelong Shen, Pengcheng He, Weizhu Chen, Zhangyang Wang, and Mingyuan Zhou. In-context learning unlocked for diffusion models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 8542–8562, 2023. 1, 6