Supplementary Material
InstructPix2Pix: Learning to Follow Image Editing Instructions

A. Additional results

See Figure 14 for an example of the biases present in our model. See Figures 15, 16, 17, 18, and 19 for more results.

B. Additional comparisons

In this section, we offer a number of additional comparisons. First, we compare our method qualitatively with Prompt-to-Prompt [2] on generated images (Figure 20). These comparisons show that our method performs comparably on synthetic images (and does better on real images, as shown in Figure 9 of the main paper). It should also be noted that P2P requires input and output captions, while our method uses edit instructions.

The fact that our method outperforms Prompt-to-Prompt (both in quantitative metrics and qualitative results on real images) may seem counter-intuitive, since Prompt-to-Prompt is used to generate training data, but this may be for a number of reasons: (1) we train on CLIP-filtered examples, (2) our method does not need DDIM inversion, (3) different classifier-free guidance formulation, or (4) the benefit of training on a large dataset.

We additionally include a comparison to a variant of SDEdit where the edit instructions provided to our method are used as the conditioning text. Qualitative results can be found in Figure 21.

We also include an additional qualitative comparison to Text2Live [1]. In Figure 22, we show results of our model on images from the Text2Live paper. We prepend "make it" to the provided prompts to make them instructions.

One potential source of bias in our evaluation protocol is the use of the same CLIP model both in evaluation and in our method (as a conditioning signal, and as a metric for training dataset filtering). To help assuage concerns of bias, we additionally include another comparison in Figure 23, where we perform the same quantitative study as in Figure 8 in the main paper, but with a different CLIP model (ViT-B/32 instead of ViT-L/14). We find that results are consistent across different CLIP models.

Finally, we comment on the runtime differences between our method and the baselines presented in the paper. Editing an image with our model takes roughly 9 seconds on an A100 GPU. This is the same speed as SDEdit (although varying with number of diffusion steps) and twice as fast as Prompt-to-Prompt, since it requires DDIM inversion for real images. Text2Live takes ∼5min since it involves optimizing for a single image.

C. Implementation Details

C.1. Instruction and Caption Generation

We finetune GPT3 to generate edit instructions and edited captions. The text prompt used during fine-tuning is the input caption concatenated with `"\n##\n"` as a separator token. The text completion is a concatenation of the instruction and edited caption with `"\n%%\n"` as a separator token in between the two and `"\nEND"` appended to the end as the stop token. During inference, we sample text completions given new input captions using `temperature=0.7` and `frequency_penalty=0.1`. We exclude generations where the input and output captions are the same.

C.2. Paired Image Generation

We generate paired before/after training images from paired before/after captions using Stable Diffusion [6] in combination with Prompt-to-Prompt [2]. We use exponential moving average (EMA) weights of the Stable Diffusion v1.5 checkpoint and the improved ft-MSE autoencoder weights. We generate images with 100 denoising steps using an Euler ancestral sampler with denoising variance schedule proposed by Kerras *et al.* [4]. We ensure the same latent noise is used for both images in each generated pair (for initial noise as well as noise introduced during stochastic sampling).

Prompt-to-Prompt replaces cross-attention weights in the second generated image differently based on the specific edit type: word swap, adding a phrase, increasing or decreasing weight of a word. We instead replaced *self*-attention weights of the second image for the first p fraction of steps, and use the same attention weight replacement strategy for all edits.

We generate 100 pairs of images for each pair of captions. We filter training data for an image-image CLIP

Input *"Make them look like flight attendants"* *"Make them look like doctors"*

Figure 14. Our method reflects biases from the data and models it is based upon, such as correlations between profession and gender.

Input *"Make it a grocery store"*

Figure 15. Leighton's *Lady in a Garden* moved to a new setting.

Input *"Convert to a realistic photo"* *"Turn into a 3D model"*

Figure 16. Van Gogh's *Self-Portrait with a Straw Hat* in different mediums.

Input | *"Add boats on the water"* | *"Replace the mountains with a city skyline"*

Figure 17. A landscape photograph shown with different contextual edits. Note that isolated changes also bring along accompanying contextual effects: the addition of boats also adds wind ripples in the water, and the added city skyline is reflected on the lake.

Input | *"It is now midnight"* | *"Add a beautiful sunset"*

Figure 18. A photograph of a cityscape edited to show different times of day.

Input | *"Apply face paint"* | *"What would she look like as a bearded man?"* | *"Put on a pair of sunglasses"* | *"She should look 100 years old"*

"What if she were in an anime?" | *"Make her terrifying"* | *"Make her more sad"* | *"Make her James Bond"* | *"Turn her into Dwayne The Rock Johnson"*

Figure 19. Vermeer's *Girl with a Pearl Earring* with a variety of edits.

Figure 20. Editing generated images. P2P generates a pair of images from a pair of captions (left, middle). Our model can perform comparable edits, given the first image and an instruction (right).

Figure 21. An alternative version of our qualitative experiments with SDEdit [5], where the guiding text used is instead the edit instruction provided to our method. These results correspond with examples in Fig. 9.

Figure 22. A comparison on the highlighted results from the Text2Live [1] paper. Here we show our results on the bottom and the edited images from Text2Live on the top.

threshold of 0.75 to ensure images are not too different, an image-caption CLIP threshold of 0.2 to ensure images correspond with their captions, and a directional CLIP similarity of 0.2 to ensure the change in before/after captions correspond with the change in before/after images. For each each pair of captions, we sort any image pairs that pass all filters by the directional CLIP similarity and keep up to 4 examples.

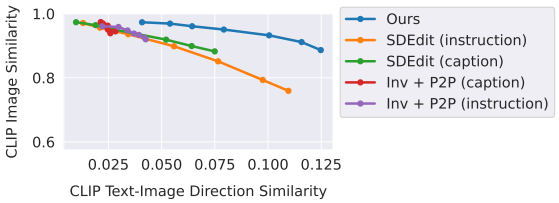

Figure 23. The same quantitative study from Figure 8 in the main paper, but using a different CLIP model (ViT-B/32 instead of ViT-L/14).

C.3. Training InstructPix2Pix

We train our image editing model for 10,000 steps on $8\times$ 40GB NVIDIA A100 GPUs over 25.5 hours. We train at 256×256 resolution with a total batch size of 1024. We apply random horizontal flip augmentation and crop augmentation where images are first resized randomly between 256 and 288 pixels and then cropped to 256. We use a learning rate of 10^{-4} (without any learning rate warm up). We initialize our model from EMA weights of the Stable Diffusion v1.5 checkpoint, and adopt other training settings from the public Stable Diffusion code base.

While our model is trained at 256×256 resolution, we find it generalized well to 512×512 resolution at inference time, and generate results in this paper at 512 resolution with 100 denoising steps using an Euler ancestral sampler with denoising variance schedule proposed by Kerras *et al.* [4]. Editing an image with our model takes roughly 9 seconds on an A100 GPU.

D. Classifier-free Guidance Details

As discussed in Section 3.2.1 of the main paper, we apply classifier-free guidance with respect to two conditionings: the input image c_I and the text instruction c_T. We introduce separate guidance scales s_I and s_T that enable separately trading off the strength of each conditioning. Below is the modified score estimate for our model with classifier-free guidance (copied from Equation 3 in the main paper):

$$\begin{aligned}\tilde{e_\theta}(z_t, c_I, c_T) = e_\theta(z_t, \varnothing, \varnothing) \\ + s_I \cdot (e_\theta(z_t, c_I, \varnothing) - e_\theta(z_t, \varnothing, \varnothing)) \\ + s_T \cdot (e_\theta(z_t, c_I, c_T) - e_\theta(z_t, c_I, \varnothing))\end{aligned}$$

Our generative model learns $P(z|c_I, c_T)$, the probability distribution of image latents $z = \mathcal{E}(x)$ conditioned on an input image c_I and a text instruction c_T. We arrive at our

particular classifier-free guidance formulation by expressing the conditional probability as follows:

$$P(z|c_T, c_I) = \frac{P(z, c_T, c_I)}{P(c_T, c_I)} = \frac{P(c_T|c_I, z)P(c_I|z)P(z)}{P(c_T, c_I)}$$

Diffusion models estimate the score [3] of the data distribution, i.e., the derivative of the log probability. Taking the logarithm gives us the following expression:

$$\begin{aligned}\log(P(z|c_T, c_I)) = \; & \log(P(c_T|c_I, z)) + \log(P(c_I|z)) \\ & + \log(P(z)) - \log(P(c_T, c_I))\end{aligned}$$

Taking the derivative and rearranging we attain:

$$\begin{aligned}\nabla_z \log(P(z|c_T, c_I)) = \; & \nabla_z \log(P(z)) \\ & + \nabla_z \log(P(c_I|z)) \\ & + \nabla_z \log(P(c_T|c_I, z))\end{aligned}$$

This corresponds with the terms in our classifier-free guidance formulation in Equation 3 in the main paper. Our guidance scale s_I effectively shifts probability mass toward data where an implicit classifier $p_\theta(c_I|z_t)$ assigns high likelihood to the image conditioning c_I, and our guidance scale s_T effectively shifts probability mass toward data where an implicit classifier $p_\theta(c_T|c_I, z_t)$ assigns high likelihood to the text instruction conditioning c_T. Our model is capable of learning these implicit classifiers by taking the differences between estimates with and without the respective conditional input. Note there are multiple possible formulations such as switching the positions of c_T and c_I variables. We found that our particular decomposition works better for our use case in practice.

References

[1] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *European Conference on Computer Vision*, pages 707–723. Springer, 2022. 1, 4

[2] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 1

[3] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005. 5

[4] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022. 1, 4

[5] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2021. 4

[6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1