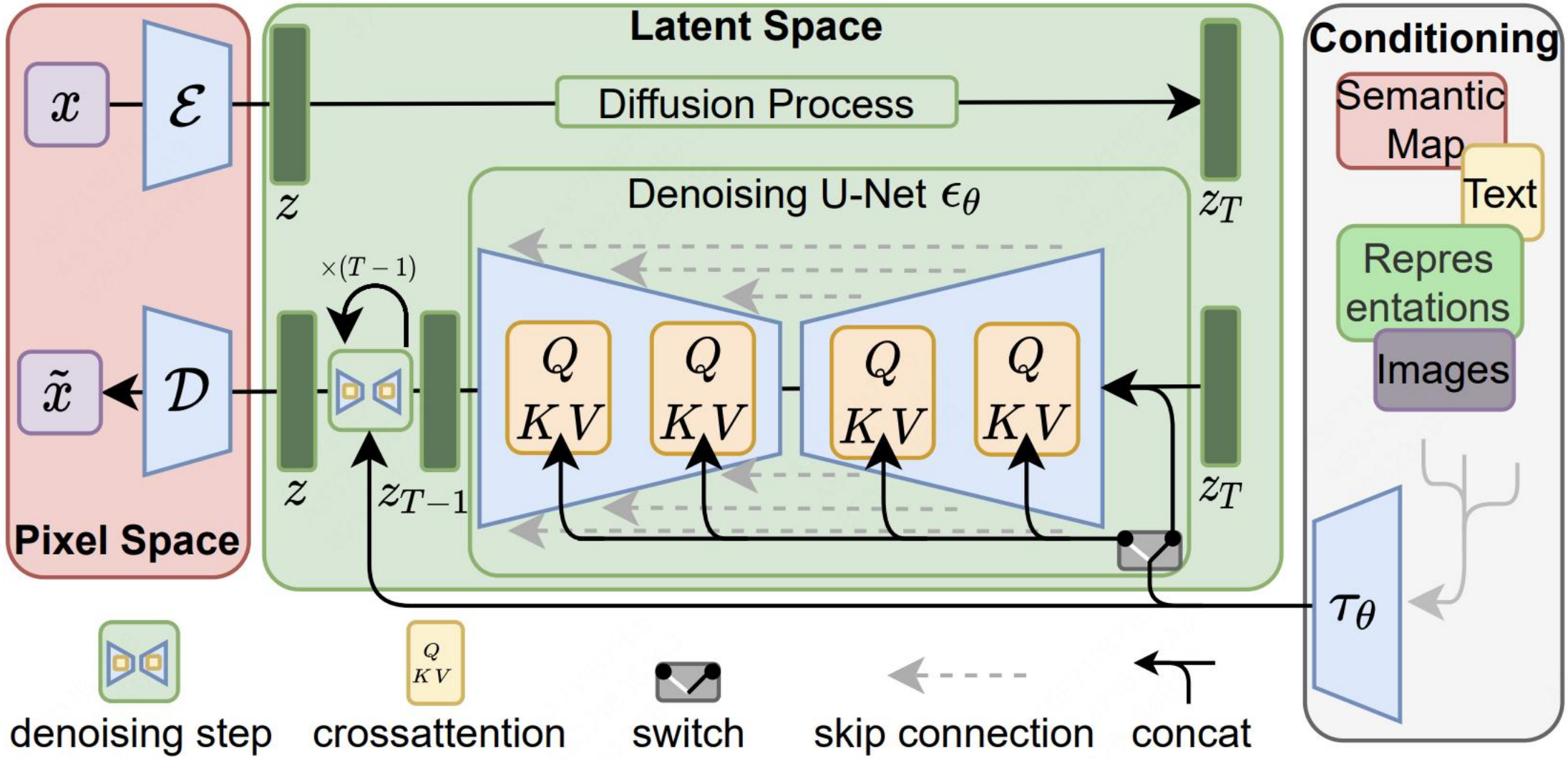
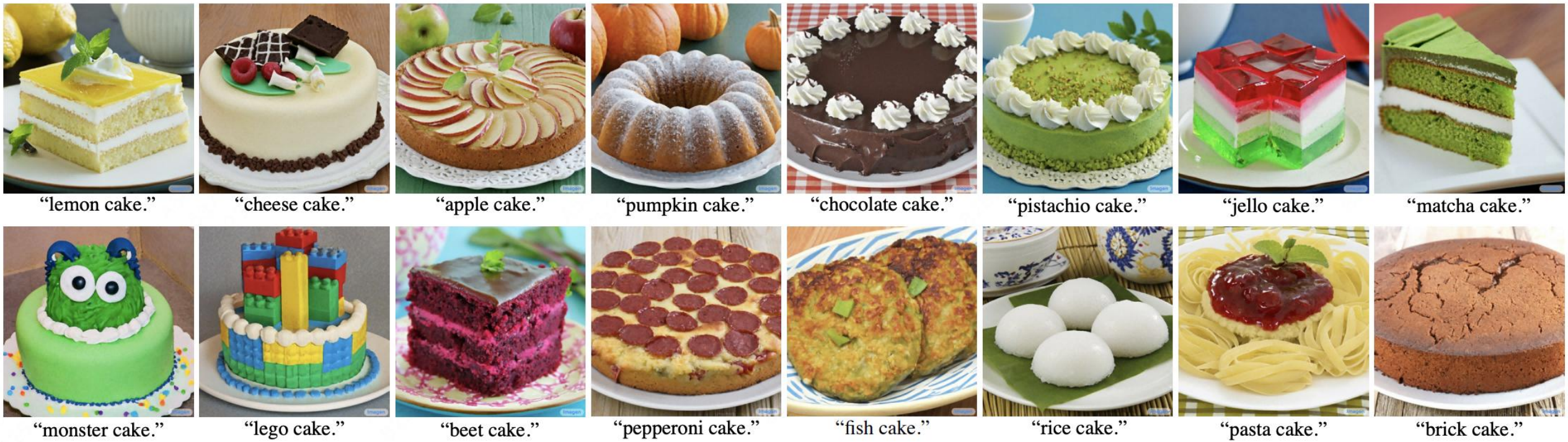

Latent Space
Diffusion Process
Denoising U-Net ϵ_θ
$\times(T-1)$
x
$\mathcal{E}$
z
z_T
$\tilde{x}$
$\mathcal{D}$
z
z_{T-1}
z_T
Q
K V
Pixel Space
Conditioning
Semantic Map
Text
Repres entations
Images
τ_θ
denoising step
crossattention
switch
skip connection
concat

Prompt-to-Prompt Image Editing with Cross Attention Control

Amir Hertz*[1,2], Ron Mokady*[1,2], Jay Tenenbaum[1], Kfir Aberman[1], Yael Pritch[1], and Daniel Cohen-Or*[1,2]

[1] *Google Research*
[2] *The Blavatnik School of Computer Science, Tel Aviv University*

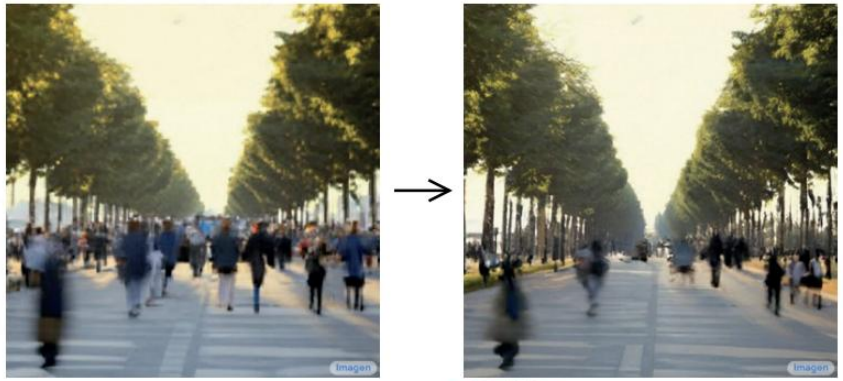
Fixed random seed



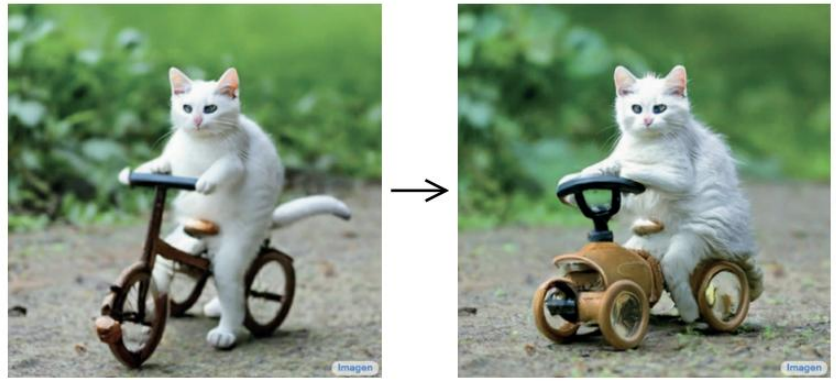
Prompt-to-Prompt Image Editing with Cross Attention Control

Amir Hertz*[1,2], Ron Mokady*[1,2], Jay Tenenbaum[1], Kfir Aberman[1], Yael Pritch[1], and Daniel Cohen-Or*[1,2]

[1] *Google Research*
[2] *The Blavatnik School of Computer Science, Tel Aviv University*

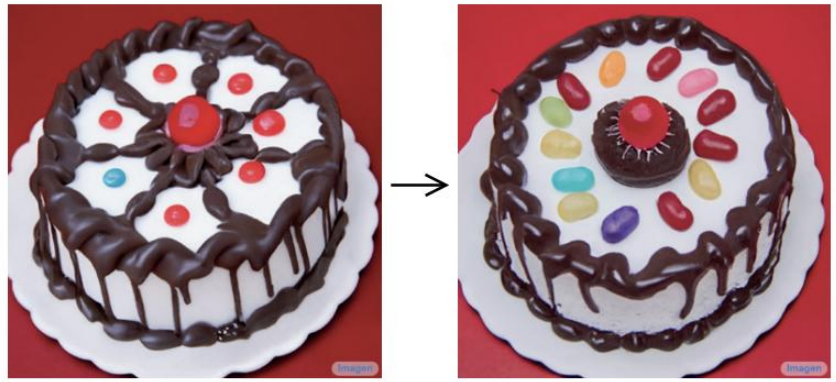


"The boulevards are crowded today."



"Photo of a cat riding on a ~~bicycle~~ car."

"Children drawing of a castle next to a river."



"a cake with decorations."

jelly beans

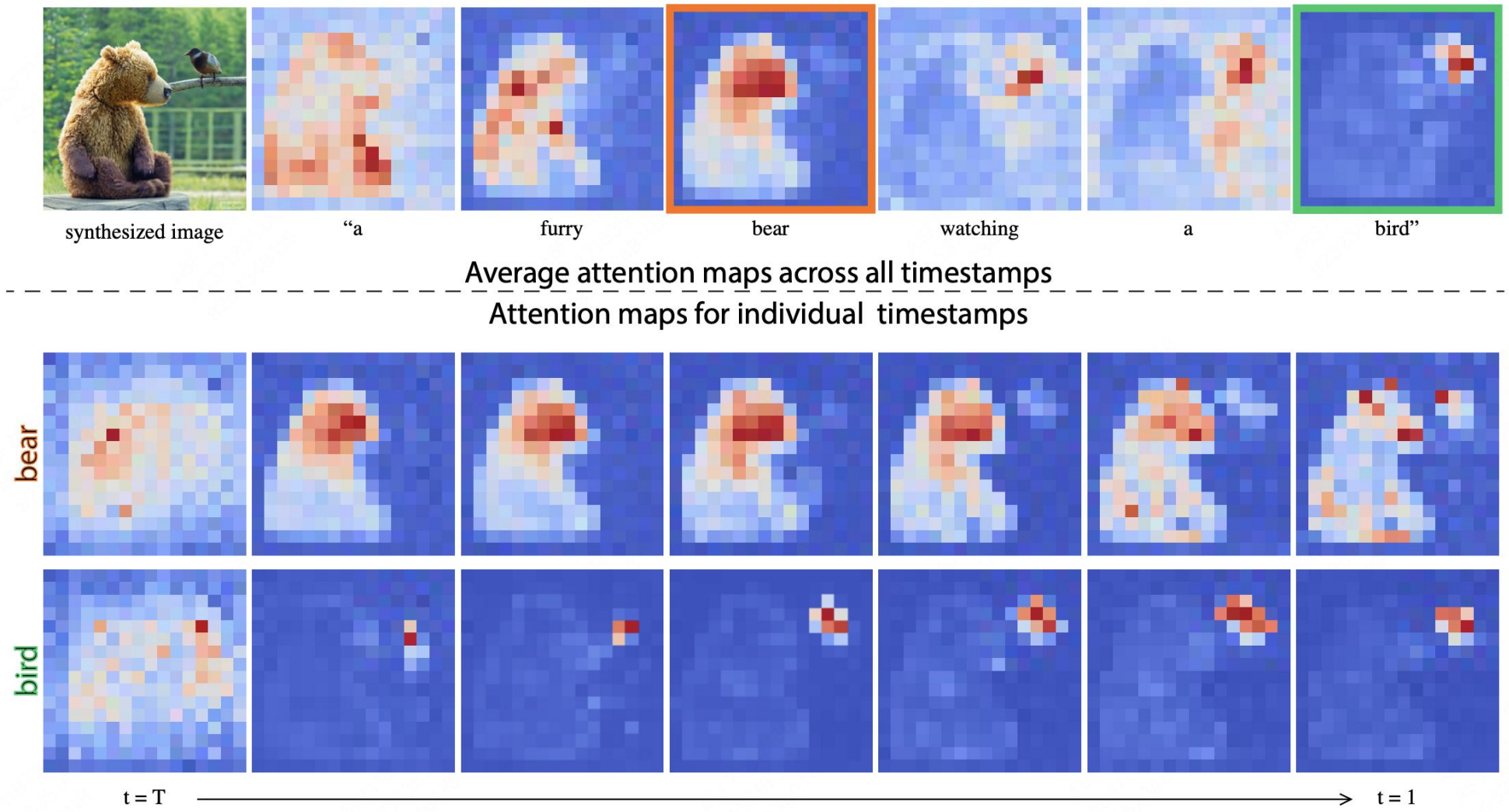


Figure 4: Cross-attention maps of a text-conditioned diffusion image generation. The top row displays the average attention masks for each word in the prompt that synthesized the image on the left. The bottom rows display the attention maps from different diffusion steps with respect to the words “bear” and “bird”.

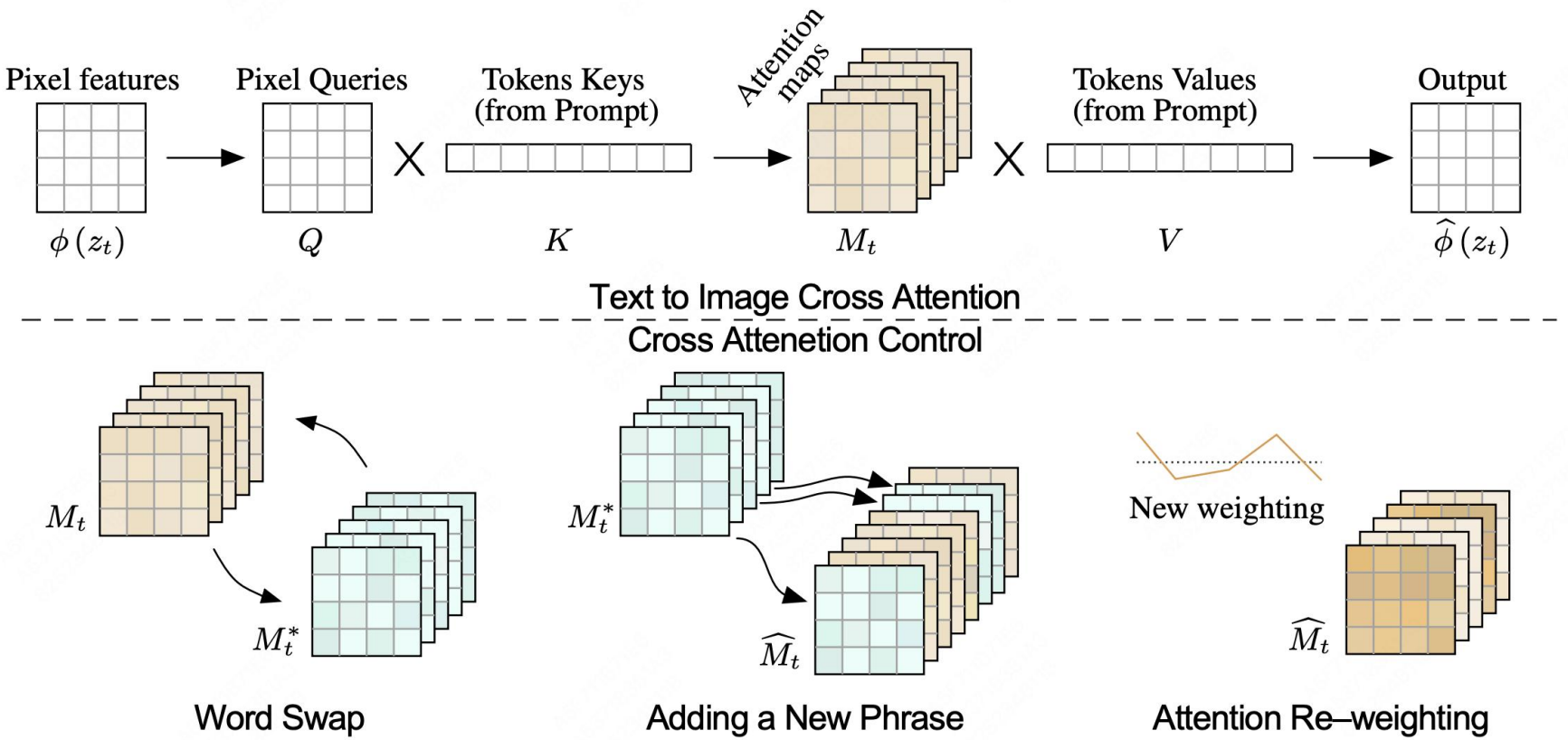


Figure 3: Method overview. Top: visual and textual embedding are fused using cross-attention layers that produce spatial attention maps for each textual token. Bottom: we control the spatial layout and geometry of the generated image using the attention maps of a source image. This enables various editing tasks through editing the textual prompt only. When swapping a word in the prompt, we inject the source image maps M_t, overriding the target image maps M_t^*, to preserve the spatial layout. Where in the case of adding a new phrase, we inject only the maps that correspond to the unchanged part of the prompt. Amplify or attenuate the semantic effect of a word achieved by re-weighting the corresponding attention map.

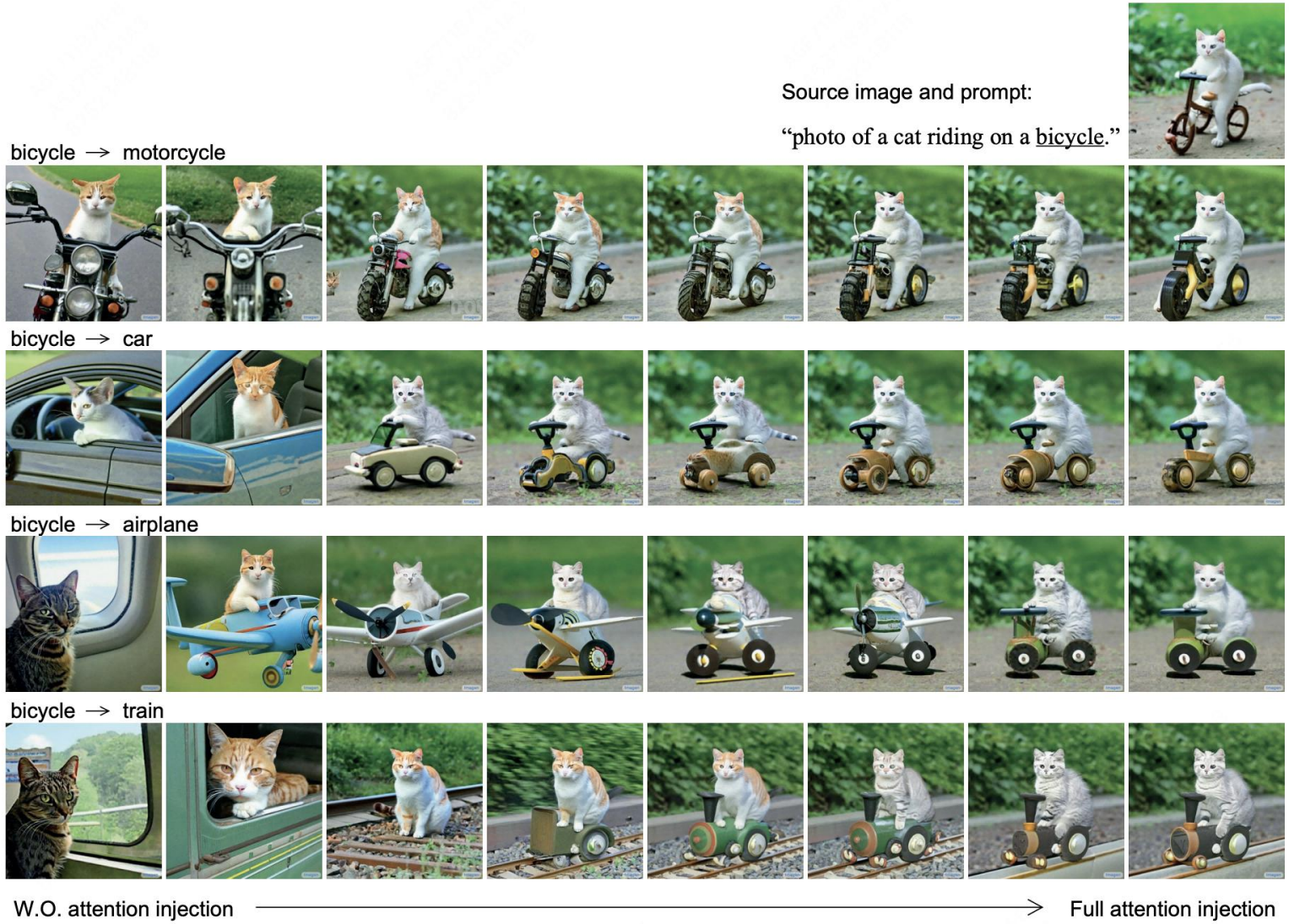


Figure 6: Attention injection through a varied number of diffusion steps. On the top, we show the source image and prompt. In each row, we modify the content of the image by replacing a single word in the text and injecting the cross-attention maps of the source image ranging from 0% (on the left) to 100% (on the right) of the diffusion steps. Notice that on one hand, without our method, none of the source image content is guaranteed to be preserved. On the other hand, injecting the cross-attention throughout all the diffusion steps may over-constrain the geometry, resulting in low fidelity to the text prompt, e.g., the car (3rd row) becomes a bicycle with full cross-attention injection.

Figure 7: Editing by prompt refinement. By extending the description of the initial prompt, we can make local edits to the car (top rows) or global modifications (bottom rows).

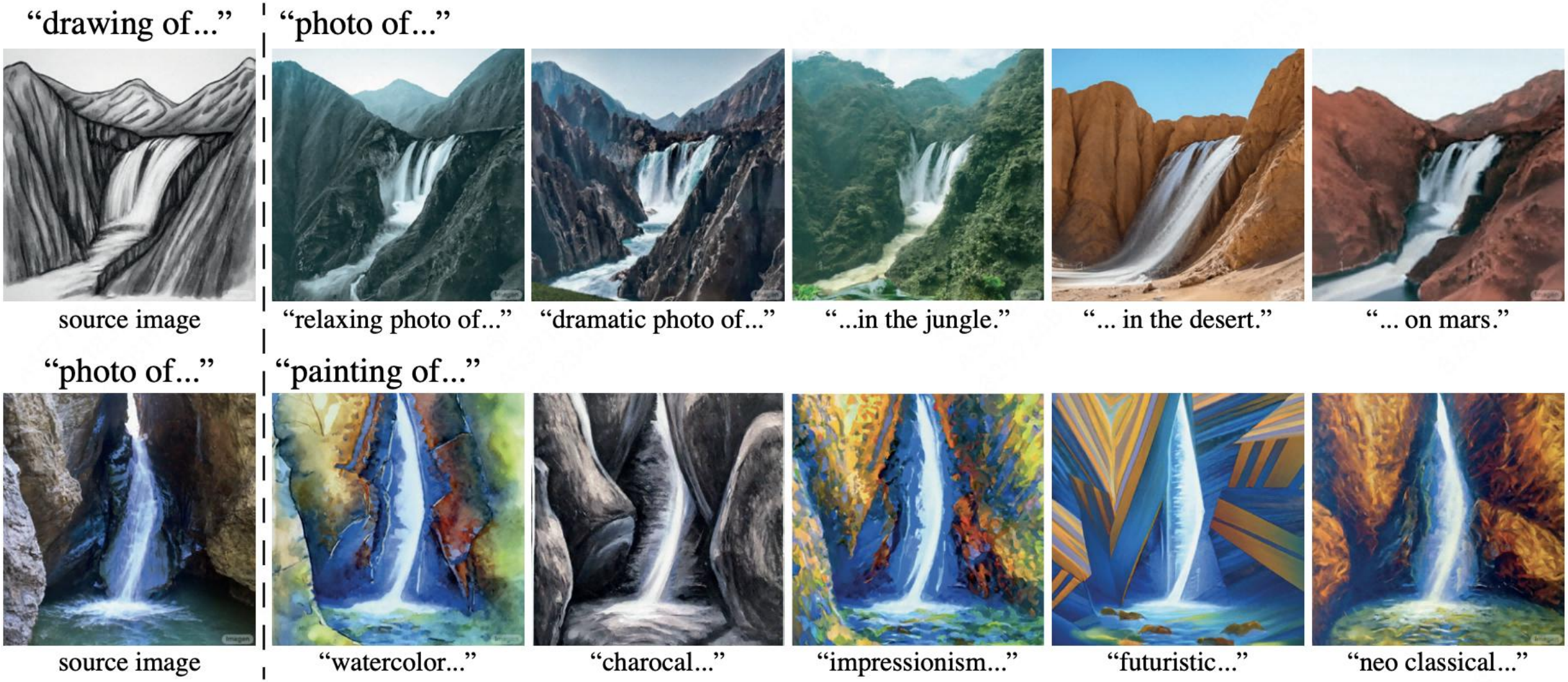


Figure 8: Image stylization. By adding a style description to the prompt while injecting the source attention maps, we can create various images in the new desired styles that preserve the structure of the original image.

"A photo of a birthday(↓) cake next to an apple."

"The picnic is ready under a blossom(↓) tree."

"A photo of a house on a snowy(↑) mountain."

"My fluffy(↑) bunny doll.

Figure 9: Text-based image editing with fader control. By reducing (top rows) or increasing (bottom) the cross-attention of the specified words (marked with an arrow), we can control the extent to which it influences the generated image.

InstructPix2Pix: Learning to Follow Image Editing Instructions

Tim Brooks* Aleksander Holynski* Alexei A. Efros

University of California, Berkeley

Figure 1. Given **an image** and **an instruction** for how to edit that image, our model performs the appropriate edit. Our model does not require full descriptions for the input or output image, and edits images in the forward pass without per-example inversion or fine-tuning.

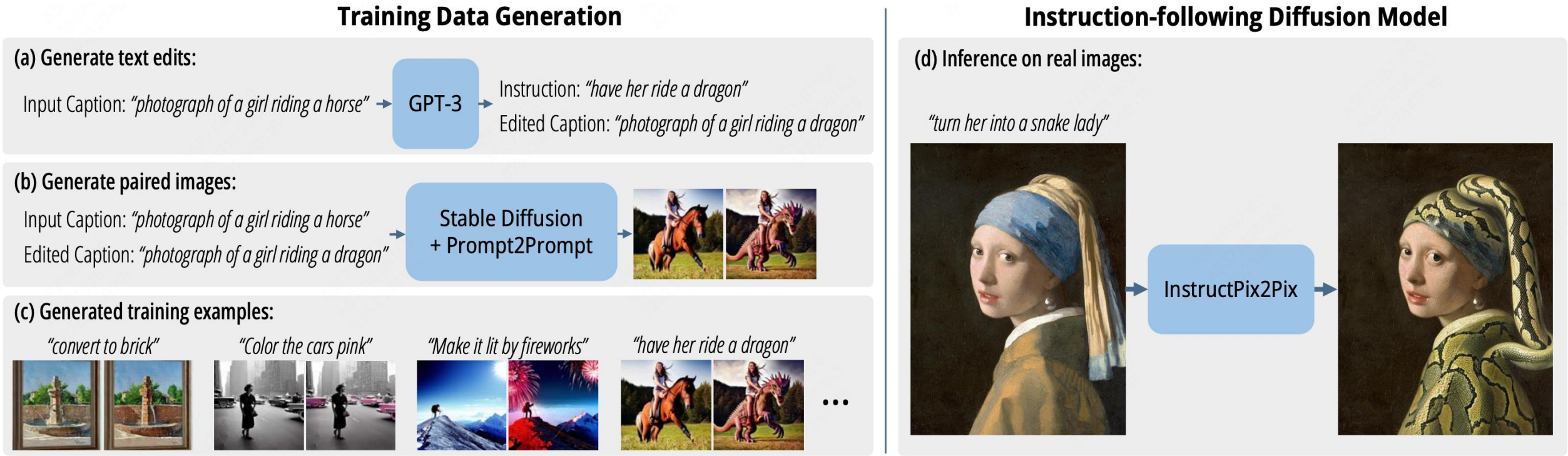


Figure 2. Our method consists of two parts: generating an image editing dataset, and training a diffusion model on that dataset. (a) We first use a finetuned GPT-3 to generate instructions and edited captions. (b) We then use StableDiffusion [52] in combination with Prompt-to-Prompt [17] to generate pairs of images from pairs of captions. We use this procedure to create a dataset (c) of over 450,000 training examples. (d) Finally, our InstructPix2Pix diffusion model is trained on our generated data to edit images from instructions. At inference time, our model generalizes to edit real images from human-written instructions.

	Input LAION caption	Edit instruction	Edited caption
Human-written (700 edits)	*Yefim Volkov, Misty Morning*	*make it afternoon*	*Yefim Volkov, Misty Afternoon*
	girl with horse at sunset	*change the background to a city*	*girl with horse at sunset in front of city*
	painting-of-forest-and-pond	*Without the water.*	*painting-of-forest*
	...	...	...
GPT-3 generated (>450,000 edits)	*Alex Hill, Original oil painting on canvas, Moonlight Bay*	*in the style of a coloring book*	*Alex Hill, Original coloring book illustration, Moonlight Bay*
	The great elf city of Rivendell, sitting atop a waterfall as cascades of water spill around it	*Add a giant red dragon*	*The great elf city of Rivendell, sitting atop a waterfall as cascades of water spill around it with a giant red dragon flying overhead*
	Kate Hudson arriving at the Golden Globes 2015	*make her look like a zombie*	*Zombie Kate Hudson arriving at the Golden Globes 2015*
	...	...	...

Table 1. We label a small text dataset, finetune GPT-3, and use that finetuned model to generate a large dataset of text triplets. As the input caption for both the labeled and generated examples, we use real image captions from LAION. Highlighted text is generated by GPT-3.

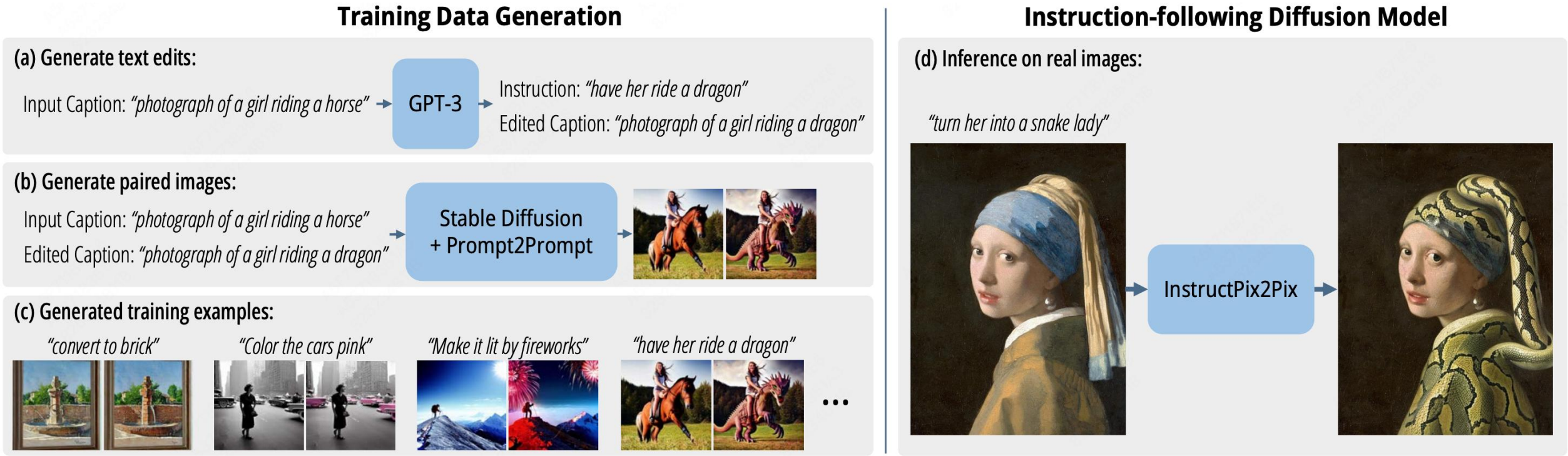


Figure 2. Our method consists of two parts: generating an image editing dataset, and training a diffusion model on that dataset. (a) We first use a finetuned GPT-3 to generate instructions and edited captions. (b) We then use StableDiffusion [52] in combination with Prompt-to-Prompt [17] to generate pairs of images from pairs of captions. We use this procedure to create a dataset (c) of over 450,000 training examples. (d) Finally, our InstructPix2Pix diffusion model is trained on our generated data to edit images from instructions. At inference time, our model generalizes to edit real images from human-written instructions.

(a) Without Prompt-to-Prompt. (b) With Prompt-to-Prompt.

Figure 3. Pair of images generated using StableDiffusion [52] with and without Prompt-to-Prompt [17]. For both, the corresponding captions are *"photograph of a girl riding a horse"* and *"photograph of a girl riding a dragon"*.

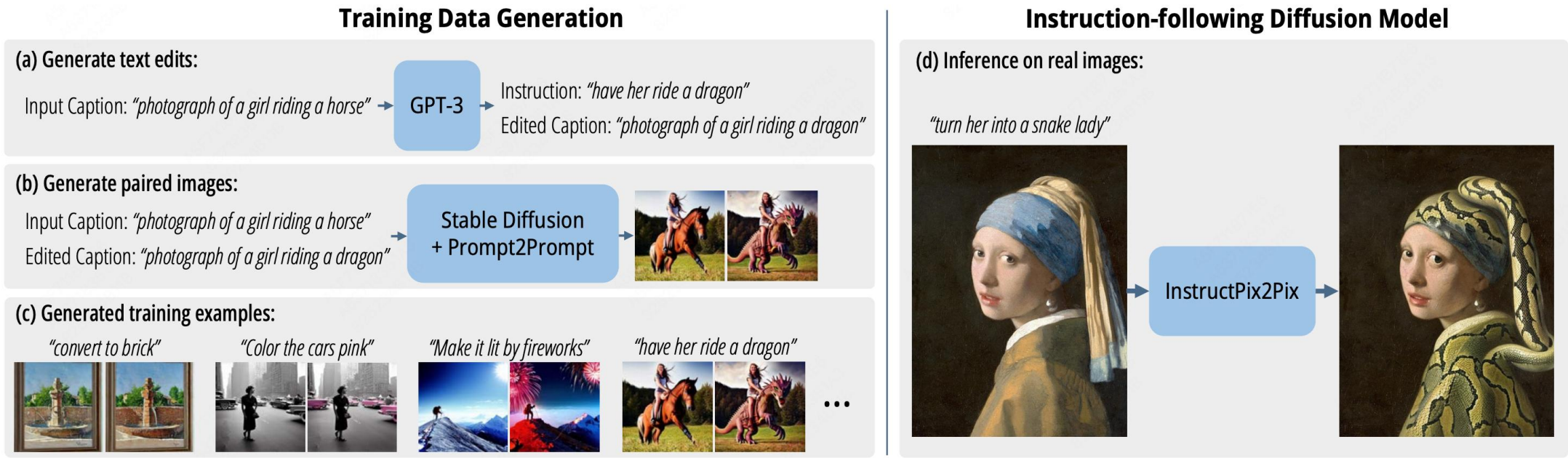
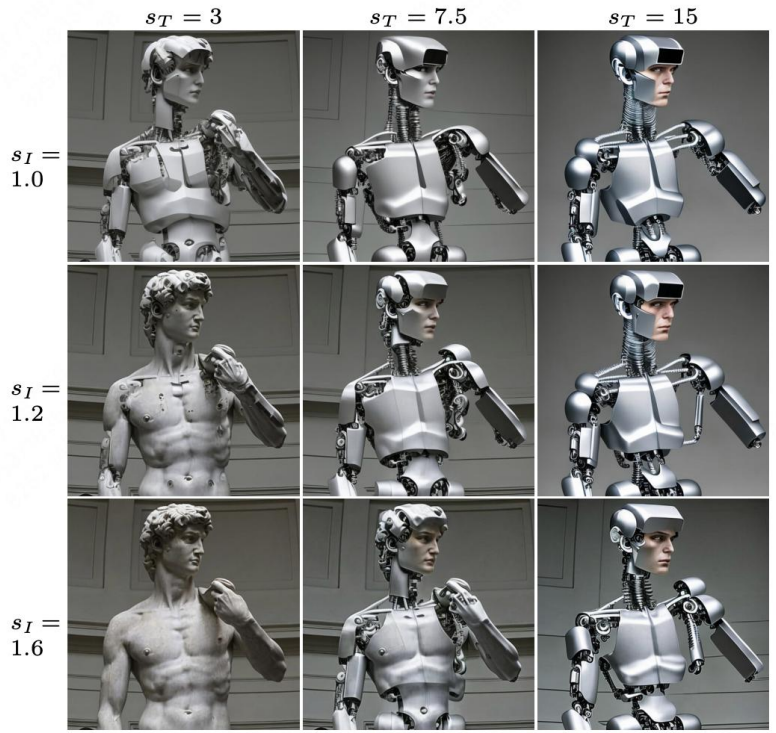


Figure 2. Our method consists of two parts: generating an image editing dataset, and training a diffusion model on that dataset. (a) We first use a finetuned GPT-3 to generate instructions and edited captions. (b) We then use StableDiffusion [52] in combination with Prompt-to-Prompt [17] to generate pairs of images from pairs of captions. We use this procedure to create a dataset (c) of over 450,000 training examples. (d) Finally, our InstructPix2Pix diffusion model is trained on our generated data to edit images from instructions. At inference time, our model generalizes to edit real images from human-written instructions.

$$L = \mathbb{E}_{\mathcal{E}(x),\mathcal{E}(c_I),c_T,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, \mathcal{E}(c_I), c_T))\|_2^2\right]$$

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



Edit instruction: *"Turn him into a cyborg!"*

Figure 4. Classifier-free guidance weights over two conditional inputs. s_I controls similarity with the input image, while s_T controls consistency with the edit instruction.

$$L = \mathbb{E}_{\mathcal{E}(x),\mathcal{E}(c_I),c_T,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, \mathcal{E}(c_I), c_T))\|_2^2\right]$$

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

Input | *"Make it a Modigliani painting"* | *"Make it a Miro painting"* | *"Make it an Egyptian sculpture"* | *"Make it a marble roman sculpture"*

Figure 5. *Mona Lisa* transformed into various artistic mediums.

Input | *"Put them in outer space"* | *"Turn the humans into robots"*

Figure 6. *The Creation of Adam* with new context and subjects (generated at 768 resolution).

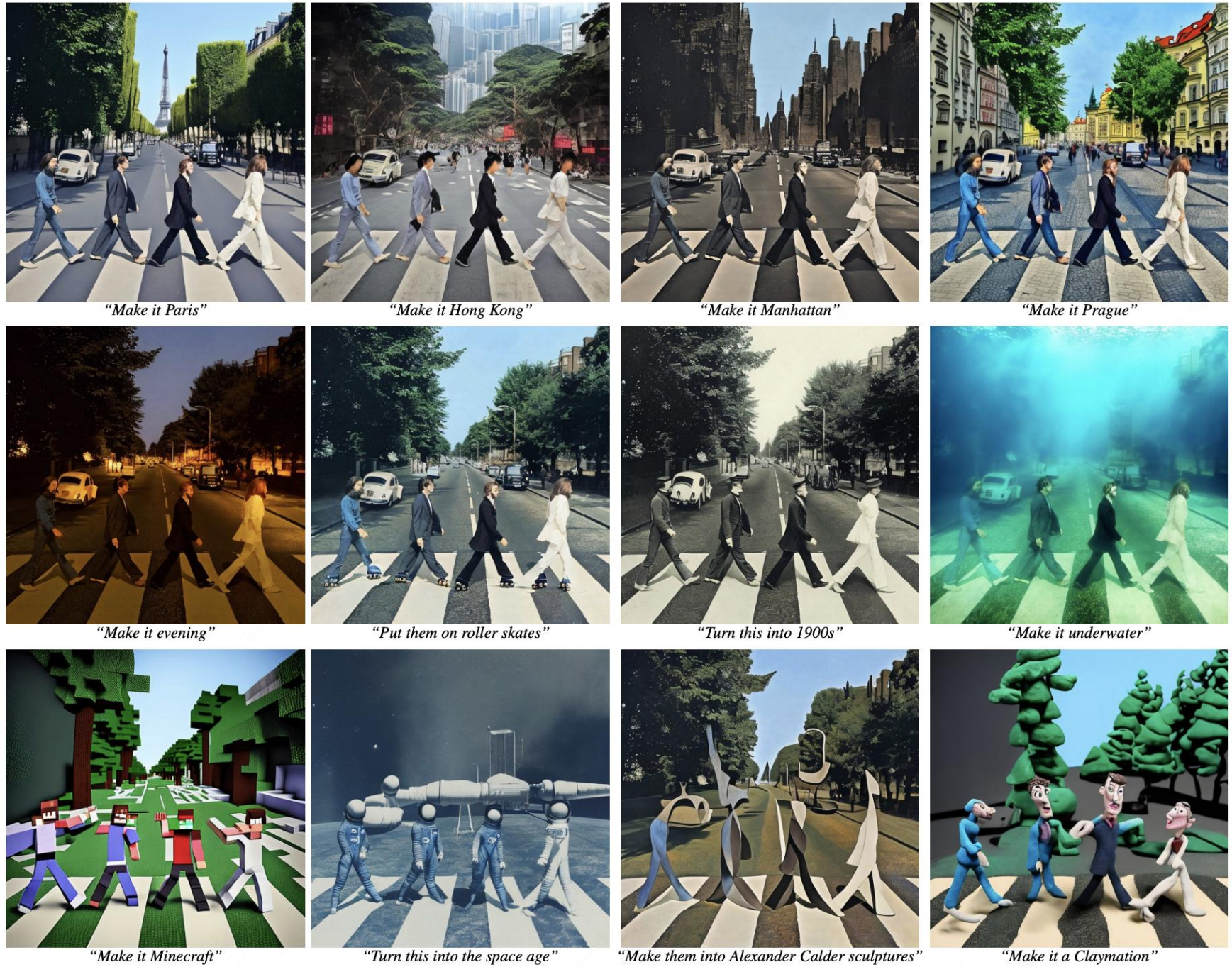


Figure 7. The iconic Beatles *Abbey Road* album cover transformed in a variety of ways.

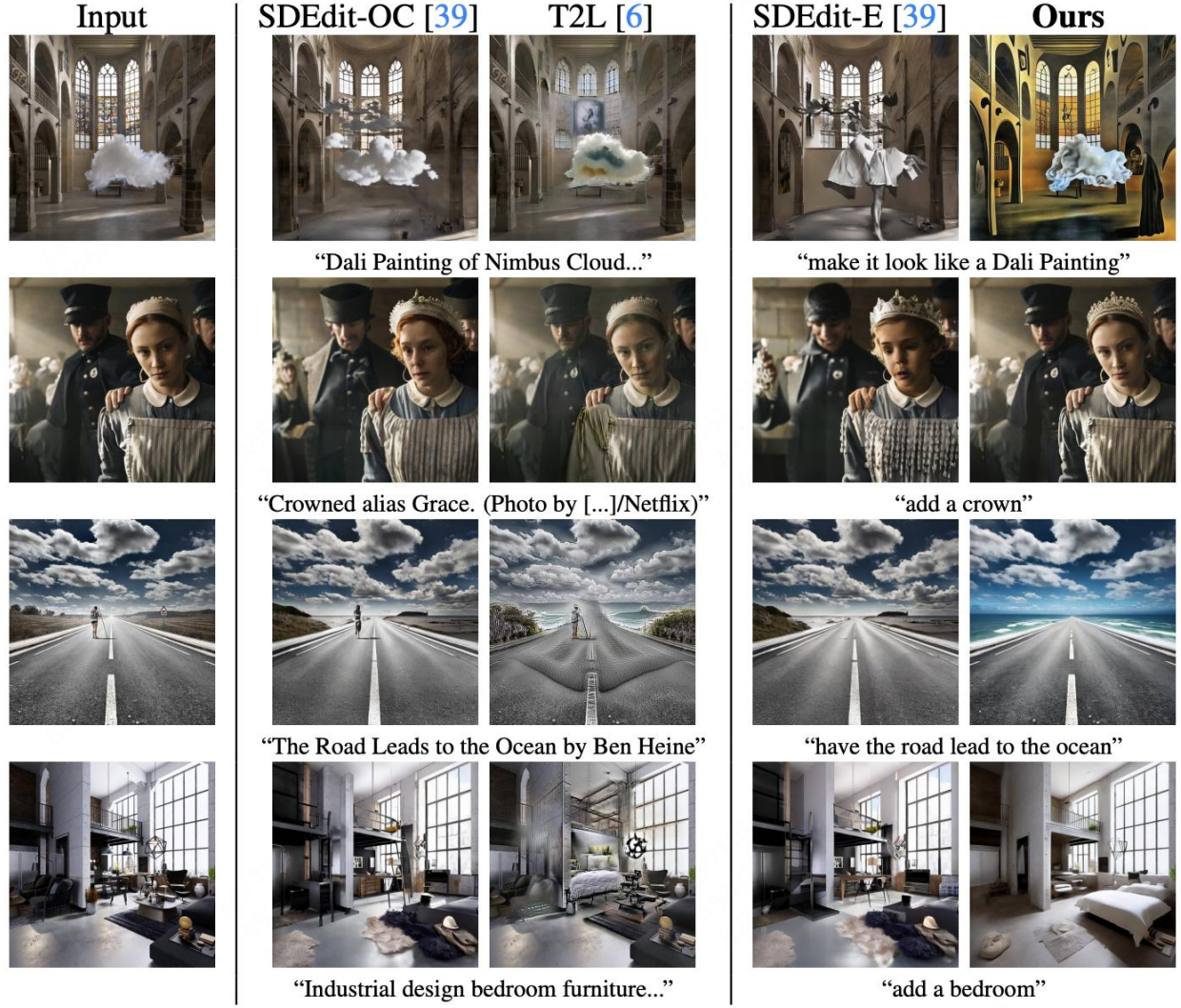


Figure 9. Comparison with other editing methods. The input is transformed either by edit string (last two columns) or the ground-truth output image caption (middle two columns). We compare our method against two recent works, SDEdit [39] and Text2Live [6]. We show SDEdit in two configurations: conditioned on the output caption (OP) and conditioned on the edit string (E).

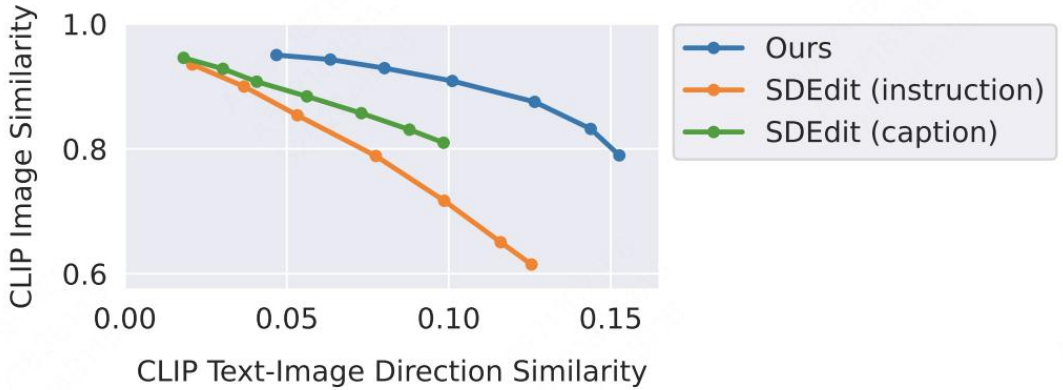


Figure 8. We plot the trade-off between consistency with the input image (Y-axis) and consistency with the edit (X-axis). For both metrics, higher is better. For both methods, we fix text guidance to 7.5, and vary our $s_I \in [1.0, 2.2]$ and SDEdit's strength (the amount of denoising) between $[0.3, 0.9]$.

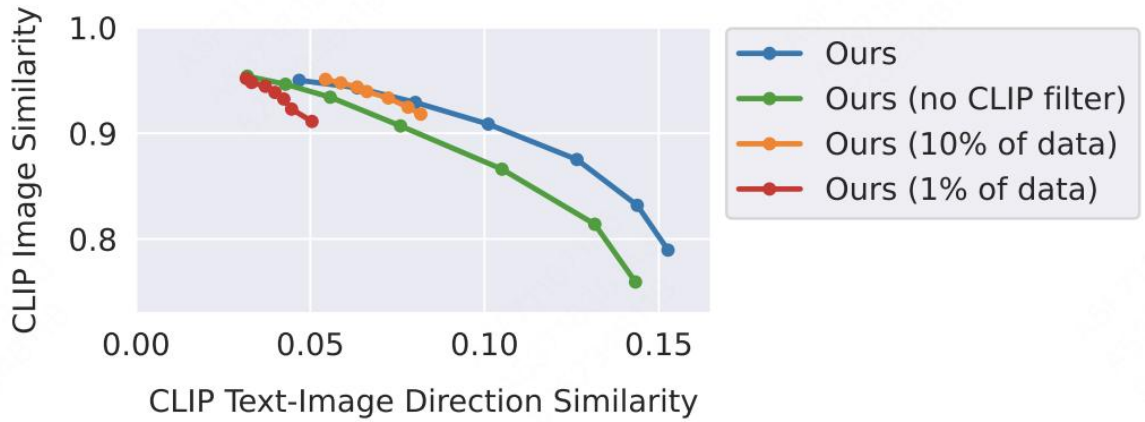


Figure 10. We compare ablated variants of our model (smaller training dataset, no CLIP filtering) by fixing s_T and sweeping values of $s_I \in [1.0, 2.2]$. Our proposed configuration performs best.