Supplementary Materials for Context Diffusion: In-Context Aware Image Generation

Ivona Najdenkoska[1,2⋆] Animesh Sinha[1] Abhimanyu Dubey[1]
Dhruv Mahajan[1] Vignesh Ramanathan[1] Filip Radenovic[1]

[1]Meta GenAI [2]University of Amsterdam

Appendix

The appendix consists of the following sections: A Additional ablations, B Limitations, C Architecture details & comparison and D Additional qualitative results.

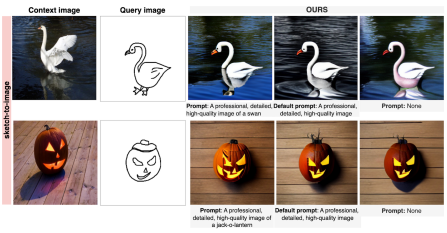
A Additional ablations

Effect of different text prompt settings. In this section, we analyze the behavior of our model with several possible text prompts for the sketch tasks. We consider (i) the given prompt (ii) default prompt "A professional, detailed, high-quality image", similar to [3] (iii) empty string as a prompt. In Figure A.1a we can observe that our model can generate images capturing the visual cues from the context example, across all text prompts. Note that having the text prompt mentioning the name of the object in the query image helps in generating finer details (like a more detailed surface of the pumpkin), however, even without it, our model can generate reasonable images.

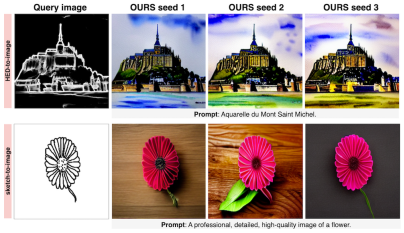
Comparison using different seeds. We investigate whether our model preserves the ability of LDMs [1] to generate diverse images. As shown in Figure A.1b, given three different seeds the model generates various output images while still preserving the layouts provided by the query image.

Choice of vision encoder We examined other vision encoders for encoding the visual context images, such as DINO and ConvNets (same configuration as the baseline) amongst others. In Figure A.1c we present examples to compare CLIP ViT-L/14 and DINO for encoding the context images. The key observation is that vision encoders already aligned with text, such as CLIP, perform better in capturing the nuanced details in the context images.

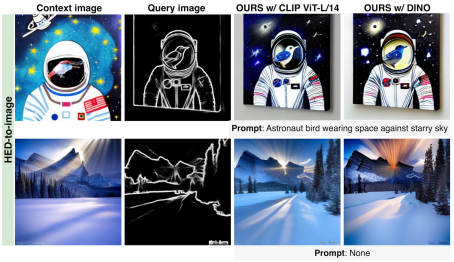
⋆ Work done during an internship at Meta GenAI. Correspondence at i.najdenkoska@uva.nl.



(a) Different text prompt settings. Our model succeeds in all three scenarios while capturing visual signals from the context.



(b) Comparison using different seeds. Our model preserves the ability of LDMs for diverse output generation.



(c) Comparison using different vision encoders. CLIP ViT-L/14 is better than DINO since it is already aligned with the text.

Fig. A.1: Additional ablations. We provide additional ablations regarding: (a) Different text prompt settings, (b) Comparison using different seeds. (c) Comparison using different vision encoders.

B Limitations

In the scenario where both the visual context and prompt are present, the current design assumes that the examples in the context are representative of the prompt. These embeddings create a stronger representation of the conditioning during the generation process. However, to build an even more flexible architecture, the visual context and prompt should ideally provide complementary information. Another limitation is the generation of images containing fine-grained details indicated in the text prompts or the visual context. For instance, image editing is such a challenging task, especially for fine-grained, local edits, as shown in Figure D.16.

Ethical considerations Our model is built using pre-trained models, such as CLIP, both for the visual and textual conditioning as well as for the image generation process. This means that it inherits any biases and limitations that may exist in these pre-trained models. Therefore, a careful analysis of the risks and societal implications should be considered before building any real-world application using these models.

C Architecture details & comparison

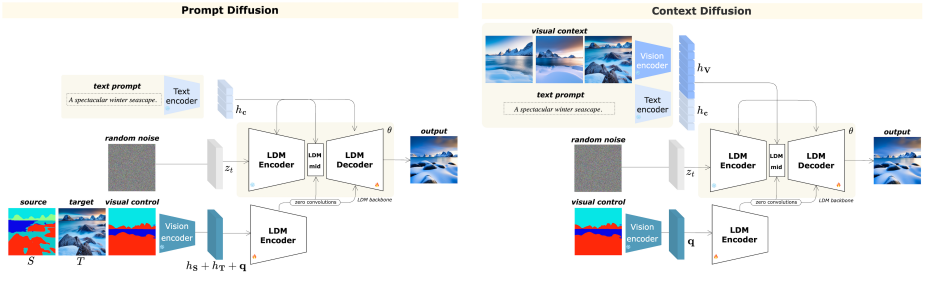


Fig. C.2: Comparison between Prompt Diffusion (left) and our model architecture (right). We propose to use k-context examples via visual conditioning which allows the model to learn the visual characteristics of the context separated from the layout of the image $\mathbf{q}$. On the other hand, Prompt Diffusion is summing the source S and target T examples directly to the visual control *i.e.* the query image $\mathbf{q}$.

In this section, we provide additional details of the implementation of our model and compare the architecture to Prompt Diffusion. In Figure C.1 we include the pseudo-code of our implementation showing how the cross-attention block is modified by using multiple images in the visual context. Furthermore, in Figure C.2 we compare the two architectures showing the difference in the visual conditioning using the context examples. We propose to stack the visual embeddings of the context examples - $\mathbf{h^V}$ next to the text embeddings $\mathbf{h^c}$. In this manner, the model is able to balance textual and visual conditioning. Moreover, it learns how to handle the structure of $\mathbf{q}$ separately from the context examples. This is different from Prompt Diffusion [2] which directly sums the examples embeddings to the query image.

```
# vision_encoder - CLIP ViT-L/14 Vision Encoder
# text_encoder - CLIP ViT-L/14 Text Encoder
# V[bs, k, H, W, C] - batch of preprocessed images
# c[bs, L] - batch of tokenized prompts
# z_t - noisy sample at time step t

# extract embeddings of the last encoder layers for each modality
h^V = vision_encoder(V).last_hidden_state   #[bs, k, N^V, d^V]
h^c = text_encoder(c).last_hidden_state     #[bs, N^c, d^c]

# sum the k-visual embeddings
h^V = torch.sum(h^V, dim=1)   #[bs, N^V, d^V]
h^V = linear_to_dc(h^V)

# stack the embeddings of both modalities
context = torch.cat((h^V, h^c), dim=1)    #[bs, N^V+N^C, d^C]

z_t = attention(norm(z_t)) + z_t  # self-attention
z_t = attention(norm(z_t), context) + z_t  # modified cross-attention
z_t = linear(norm(z_t)) + z_t
```

Fig. C.1: Pseudo-code for a torch-like implementation of the modified cross-attention block in our model, by using k-images as visual context examples.

Another difference is in the context examples. Different from Prompt Diffusion, we do not provide a source image, since it can be derived from the target (context) image itself, meaning it does not provide any additional information. Note that in our early experiments, we did use a pair of source and target images, however, it showed to not bring any improvements (see Ablation in the main paper). Furthermore, the ControlNet framework [3] is capable of controlling image generation based on image structures, making the source image unnecessary. We also provide the flexibility to include more than one context example, to learn stronger visual representations as conditioning, as shown in Figure C.2.

D Additional qualitative results

In the following sections, we provide additional qualitative results, spanning from in-domain tasks, such as handling HED, segmentation, and depth maps to out-of-domain tasks, such as editing sketches, canny edges and scribbles, as well as examples of few-shot settings.

References

1. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
2. Wang, Z., Jiang, Y., Lu, Y., Shen, Y., He, P., Chen, W., Wang, Z., Zhou, M.: In-context learning unlocked for diffusion models. arXiv preprint arXiv:2305.01115 (2023)
3. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2023)

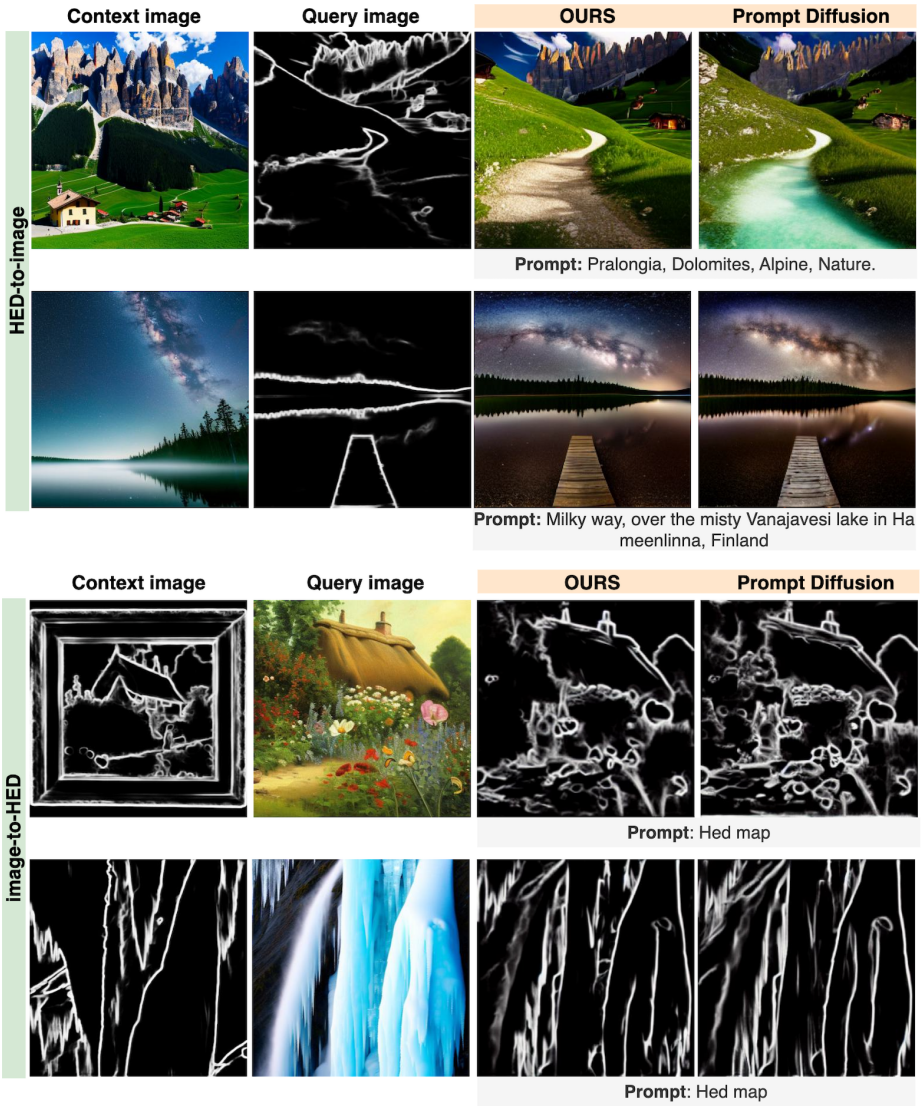


Fig. D.1: HED-to-image and vice versa, with visual context and prompt as conditioning, in-domain comparison to Prompt Diffusion.

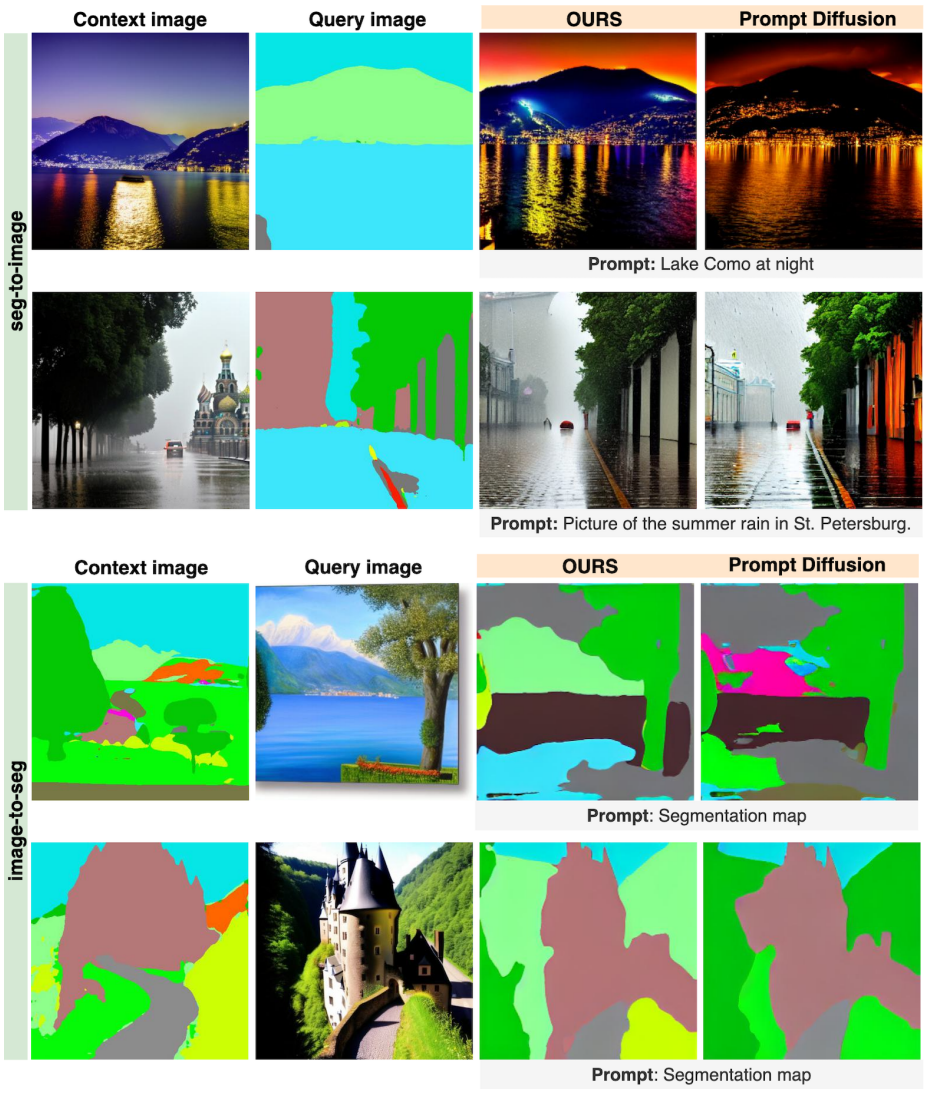


Fig. D.2: Seg-to-image and vice versa, with visual context and prompt as conditioning, in-domain comparison to Prompt Diffusion.

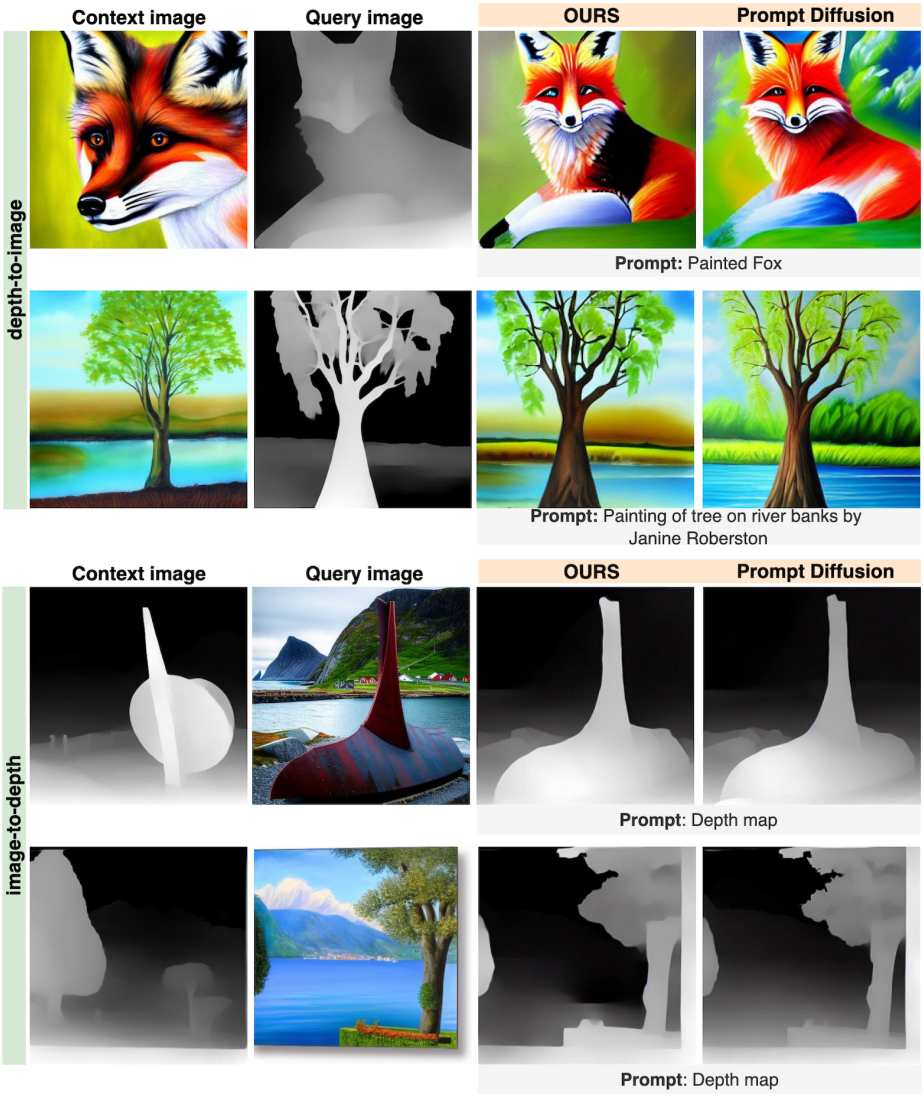


Fig. D.3: Depth-to-image and vice versa, with visual context and prompt as conditioning, in-domain comparison to Prompt Diffusion.

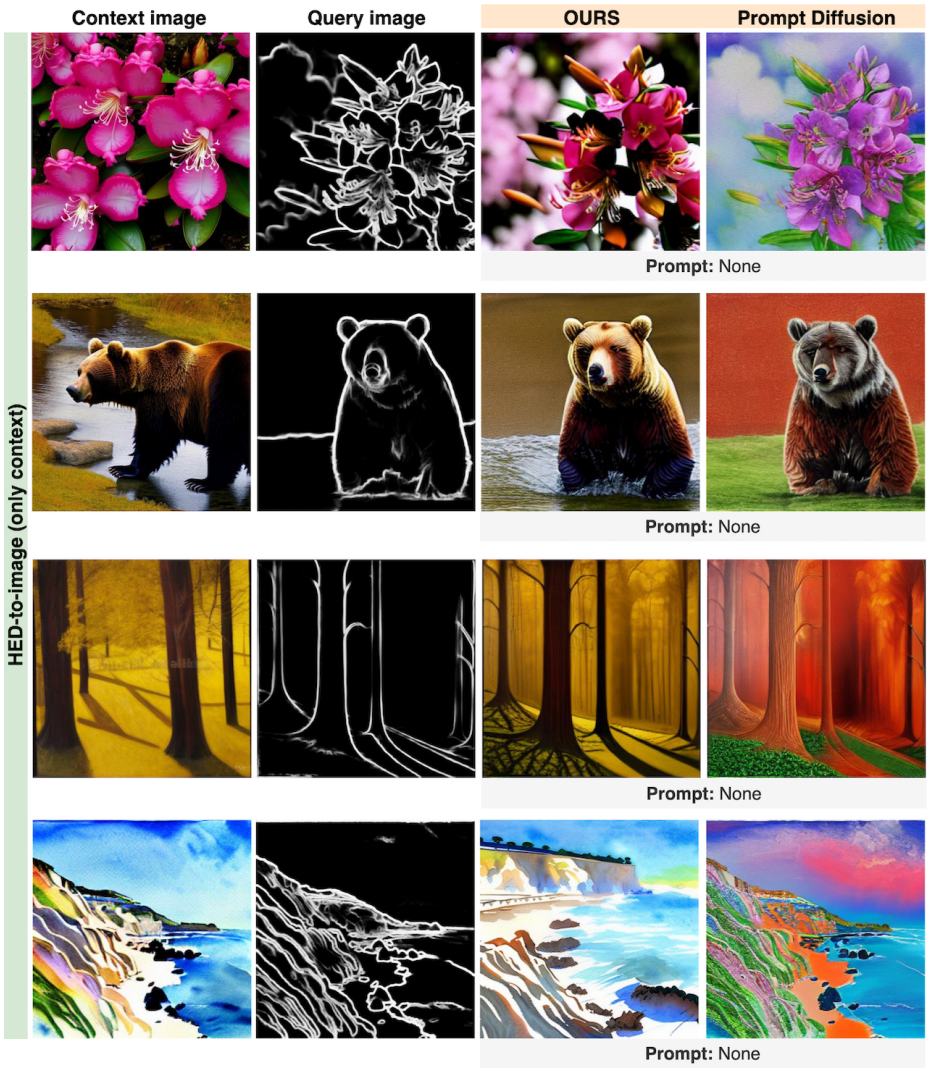


Fig. D.4: HED-to-image, only with context images as conditioning, in-domain comparison to Prompt Diffusion [2].

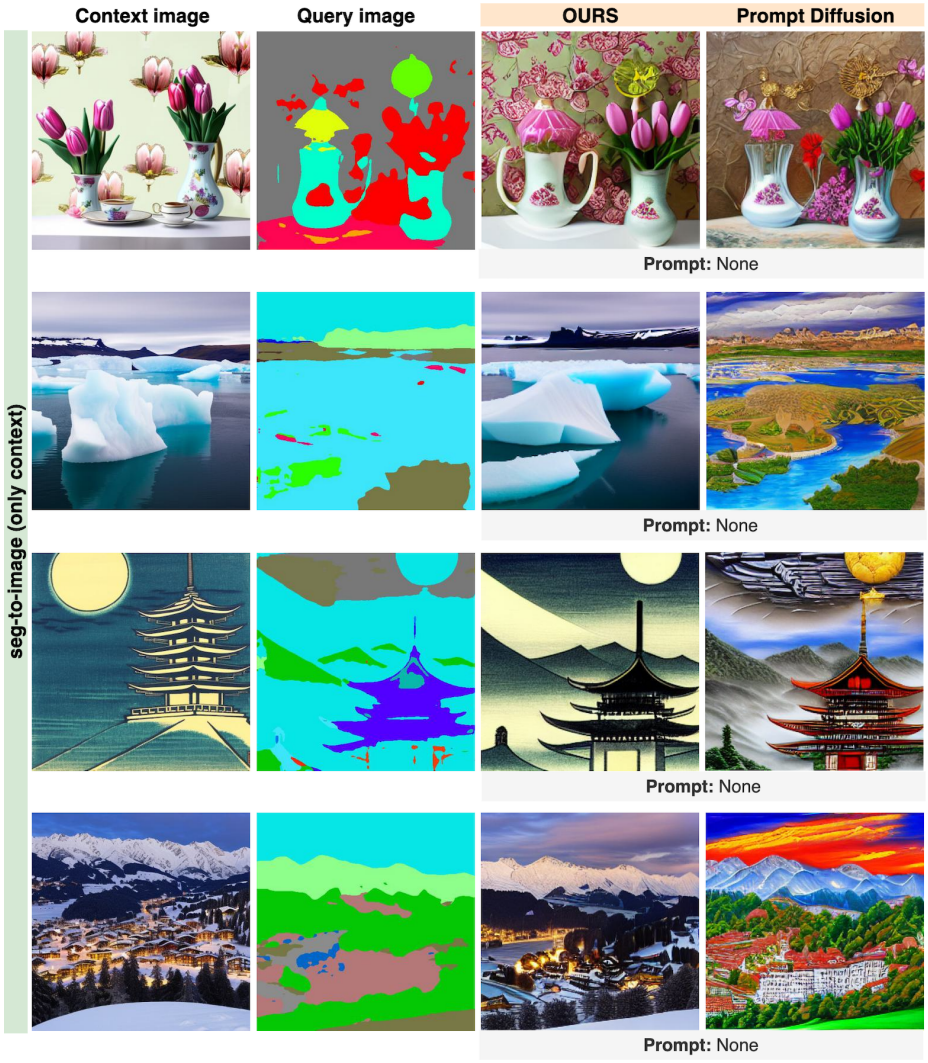


Fig. D.5: Seg-to-image, only with context images as conditioning, in-domain comparison to Prompt Diffusion [2].

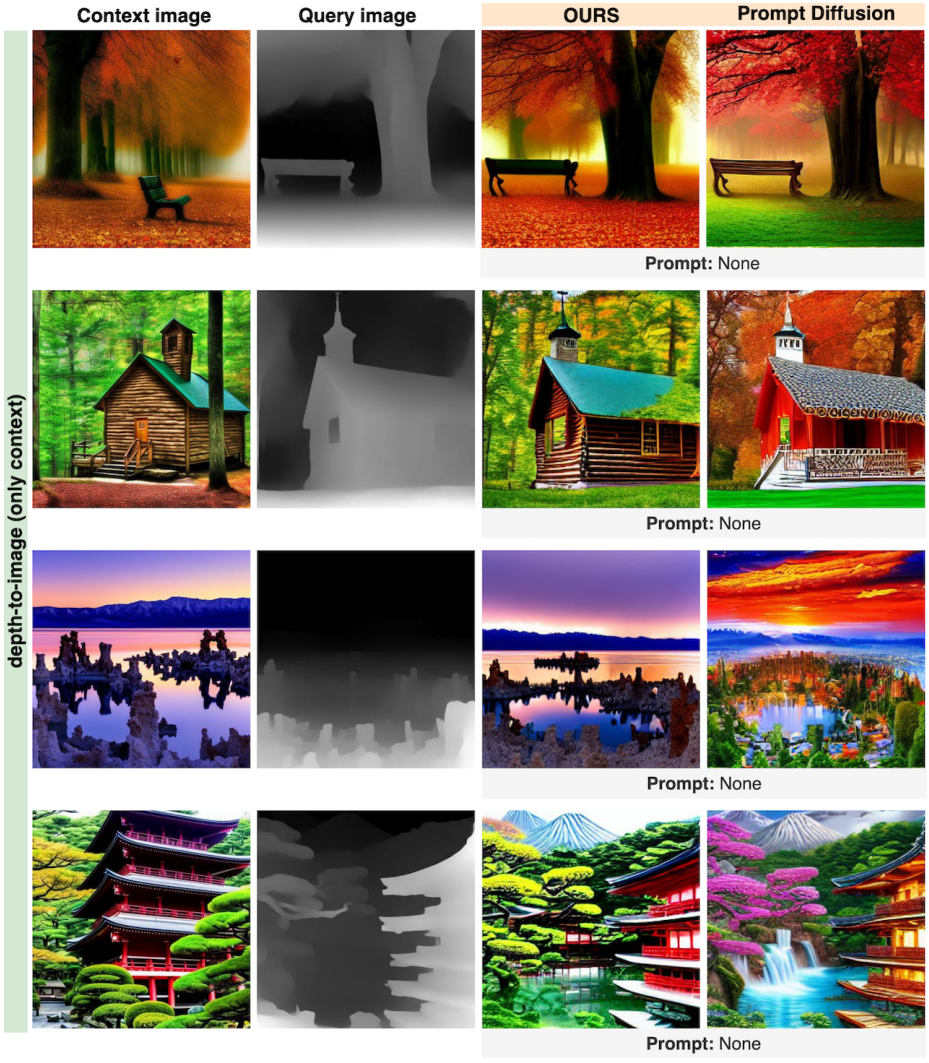


Fig. D.6: Depth-to-image, only with context images as conditioning, in-domain comparison to Prompt Diffusion [2].

Fig. D.7: HED-to-image, only with text prompt as conditioning, in-domain comparison to Prompt Diffusion [2] and Control Net [3].

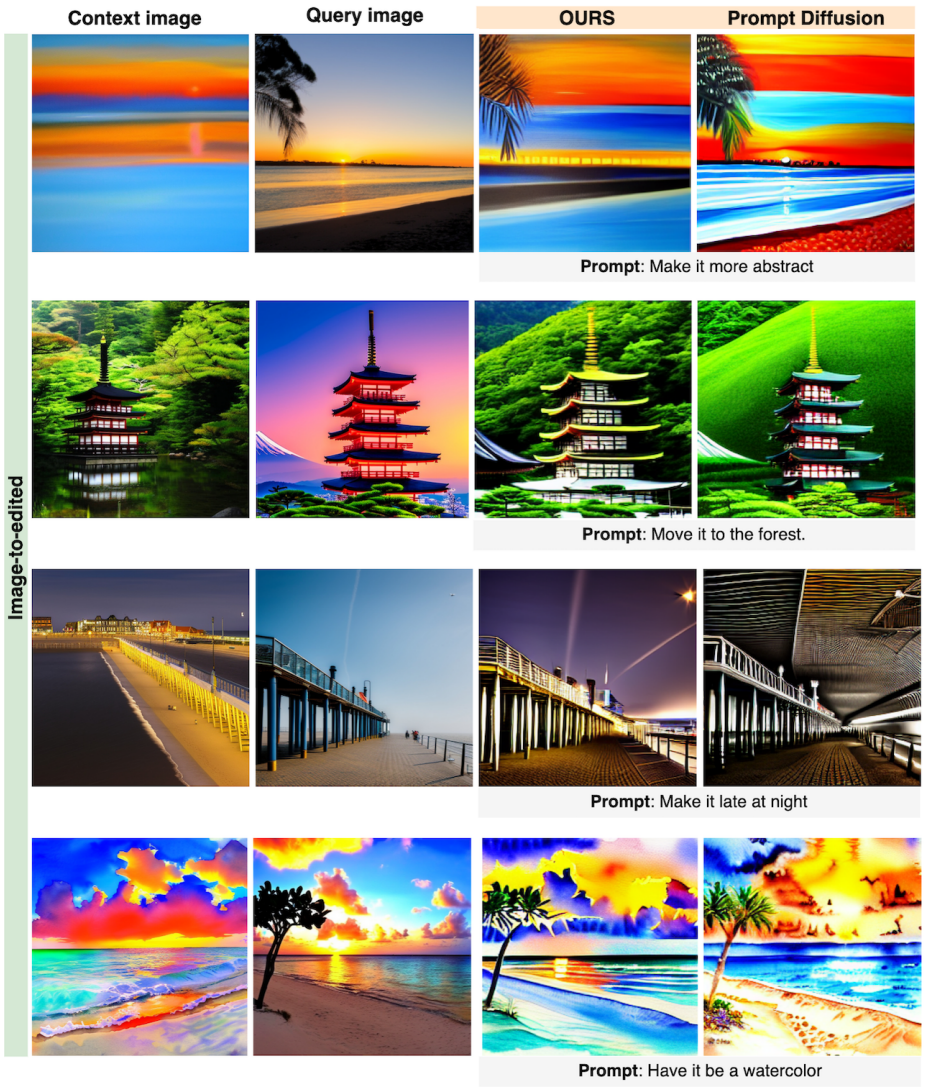


Fig. D.8: Image edit, with visual context and prompt as conditioning, out-of-domain comparison to Prompt Diffusion [2].

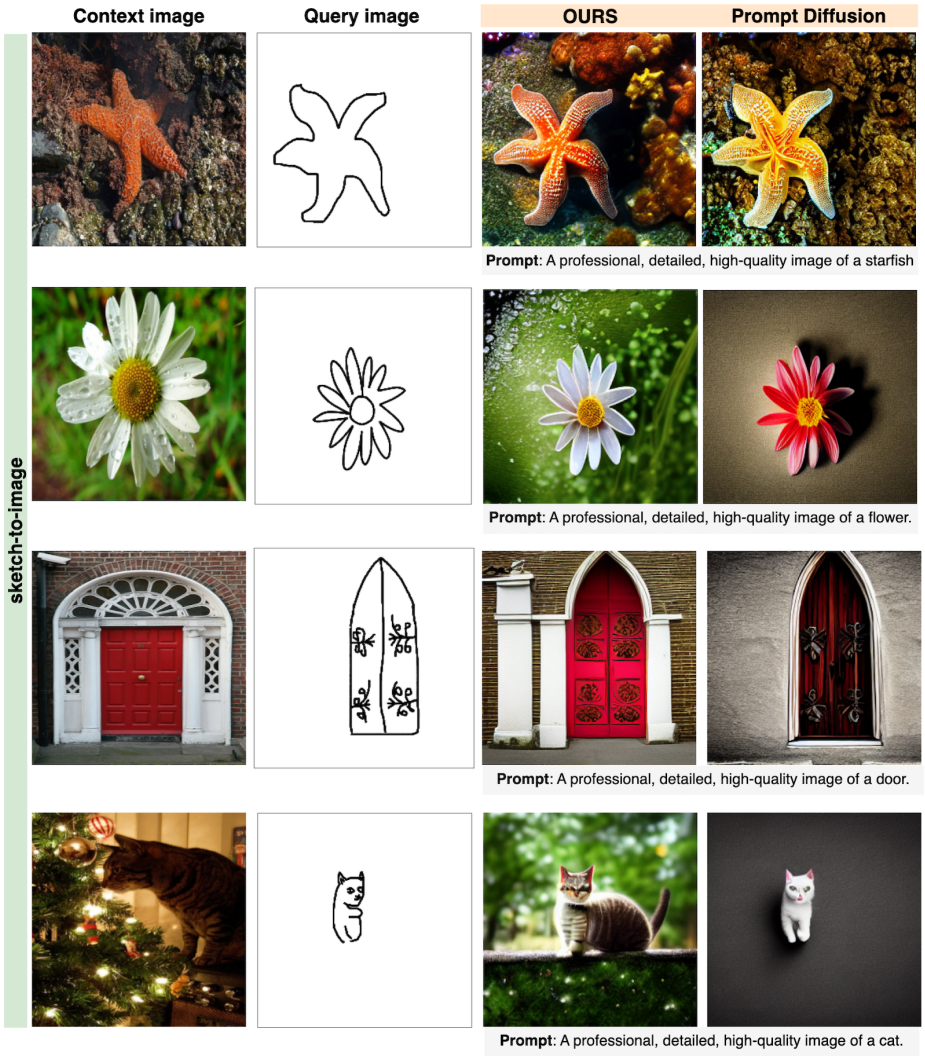


Fig. D.9: Sketch-to-image, with visual context and prompt as conditioning, out-of-domain comparison to Prompt Diffusion [2].

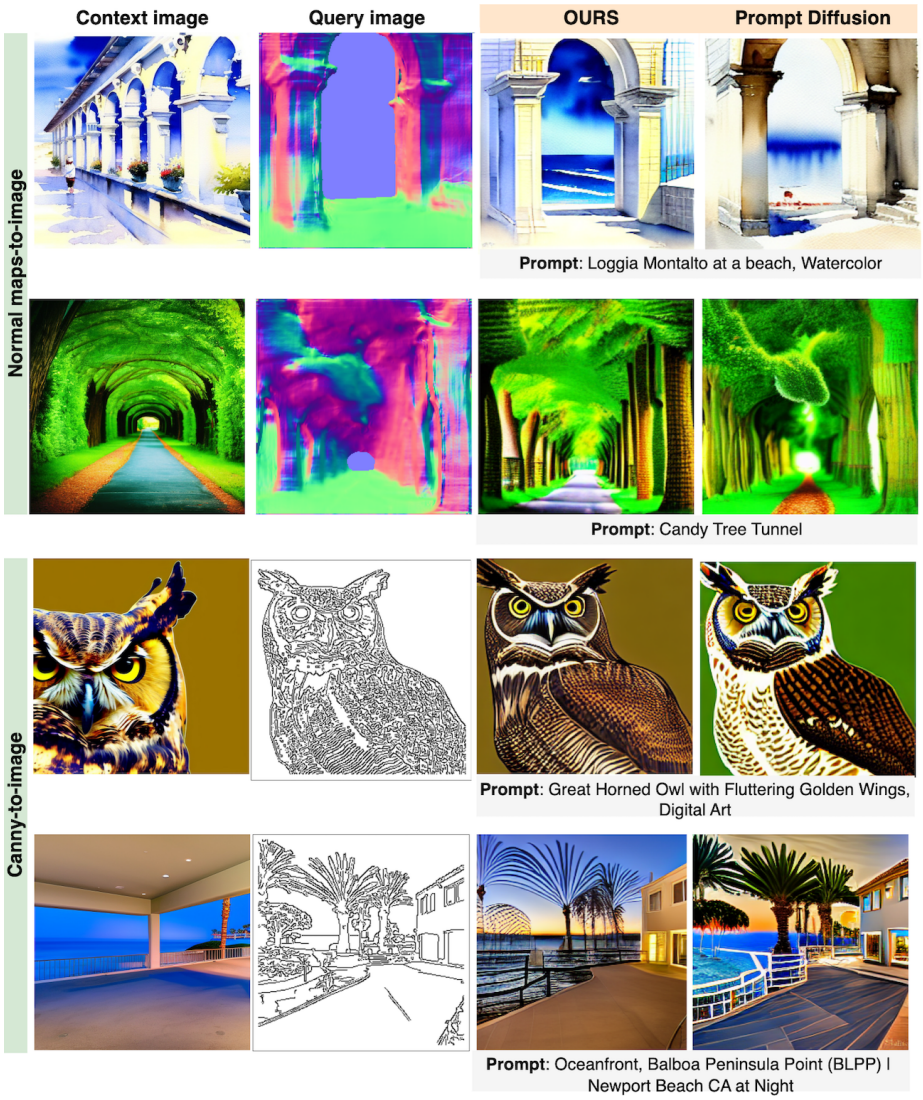


Fig. D.10: Normal map-to-image (rows 1-2) and canny-to-image (rows 3-4), both with visual context and prompt as conditioning, out-of-domain comparison to Prompt Diffusion [2].

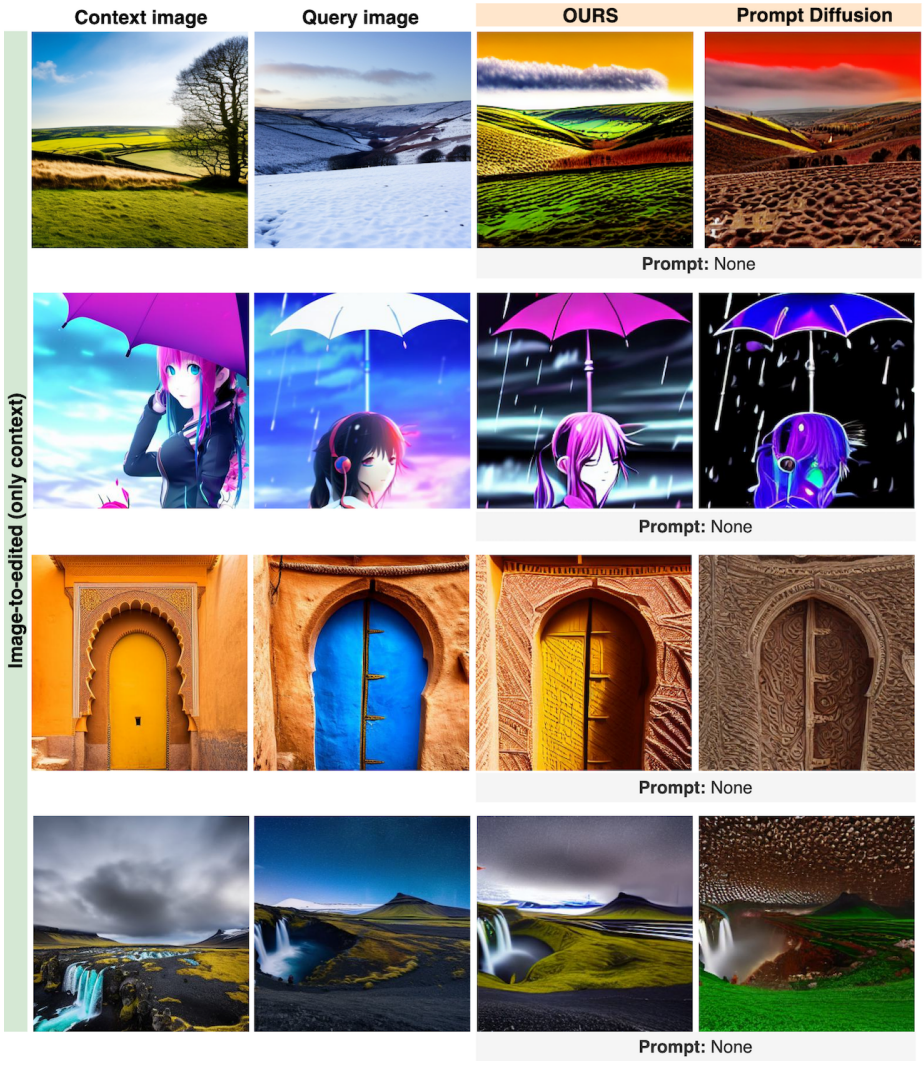


Fig. D.11: Image editing, only with context images as conditioning, out-of-domain comparison to Prompt Diffusion [2].

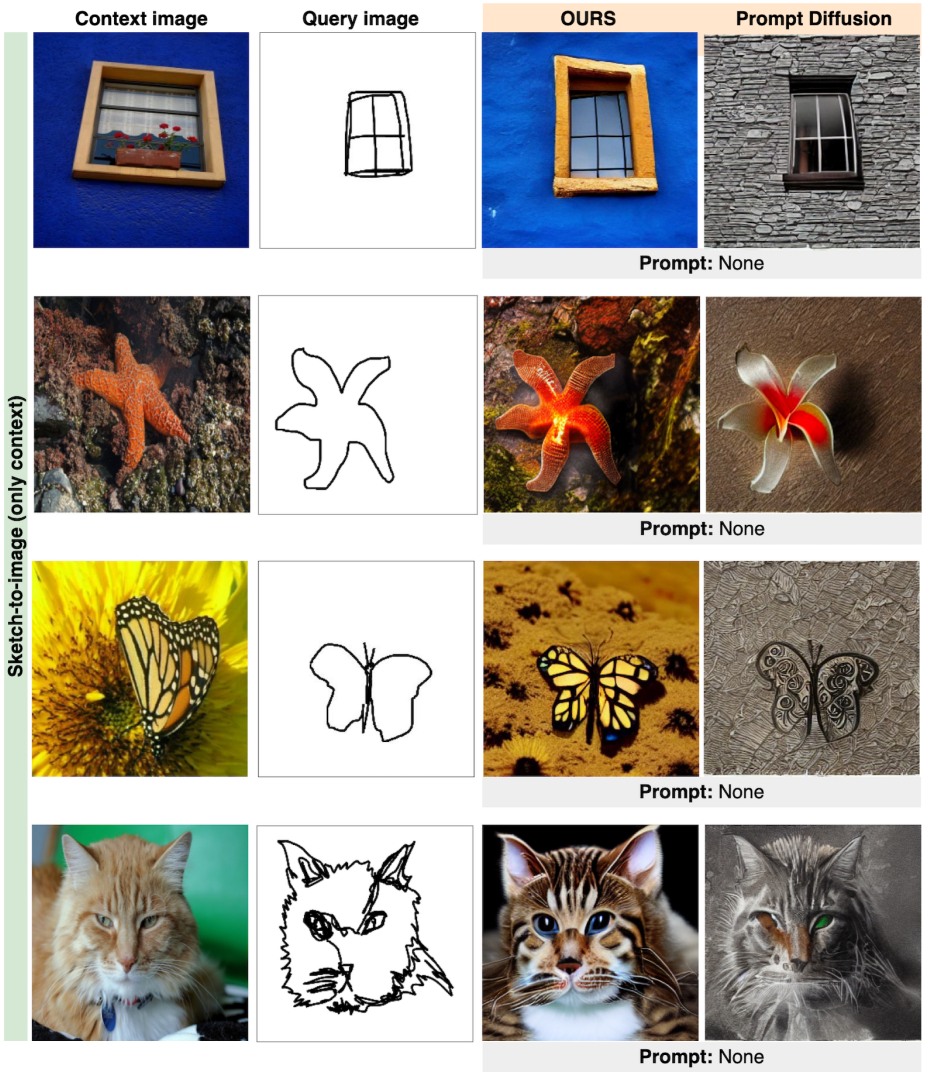


Fig. D.12: Sketch-to-image, only with context images as conditioning, out-of-domain comparison to Prompt Diffusion [2].

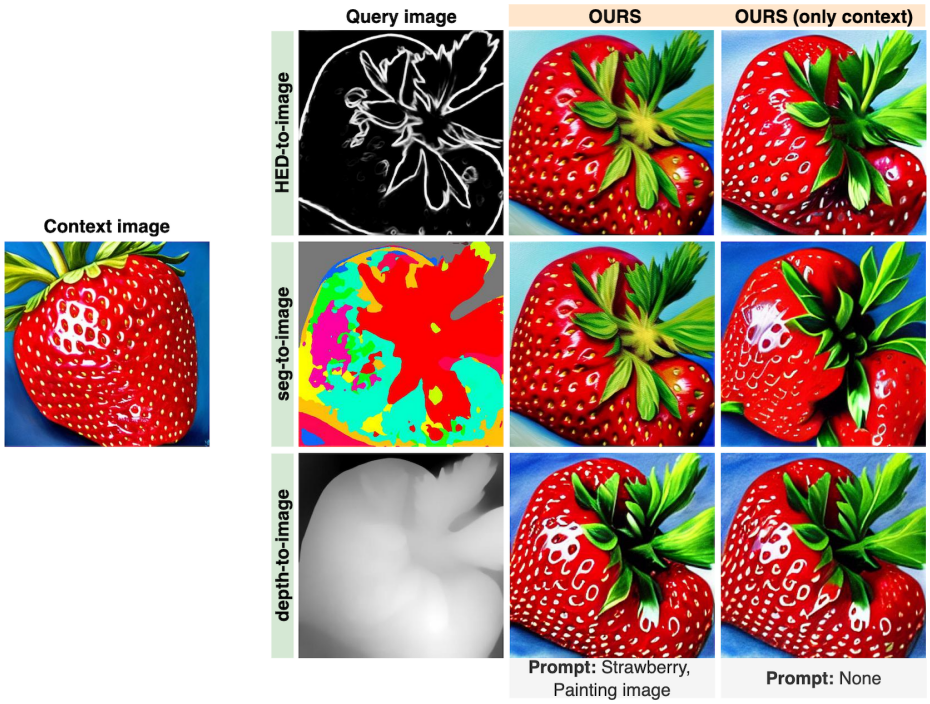


Fig. D.13: Examples using different types of query images, with visual context and prompt, and only using context images as conditioning. Our model can successfully generate images that match the visual context and/or prompt, independent of the type of visual control provided by the query image.

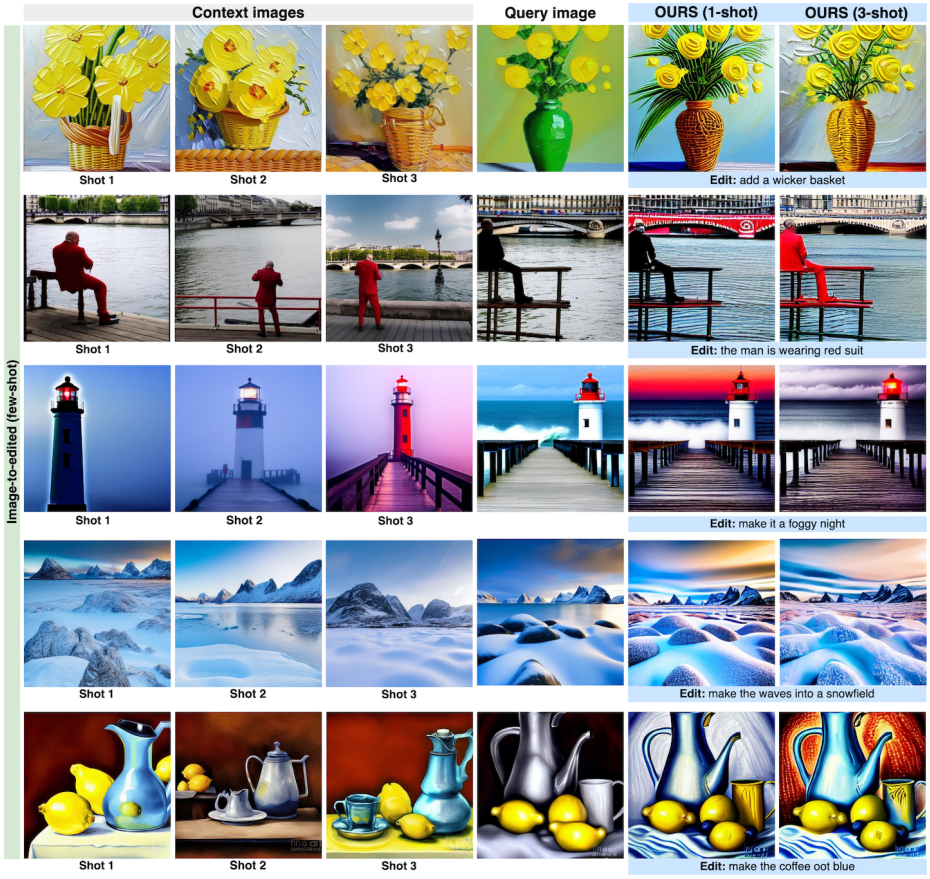


Fig. D.14: Few-shot examples: image edit. Comparison using one, two, and three shots of context examples with text prompts.

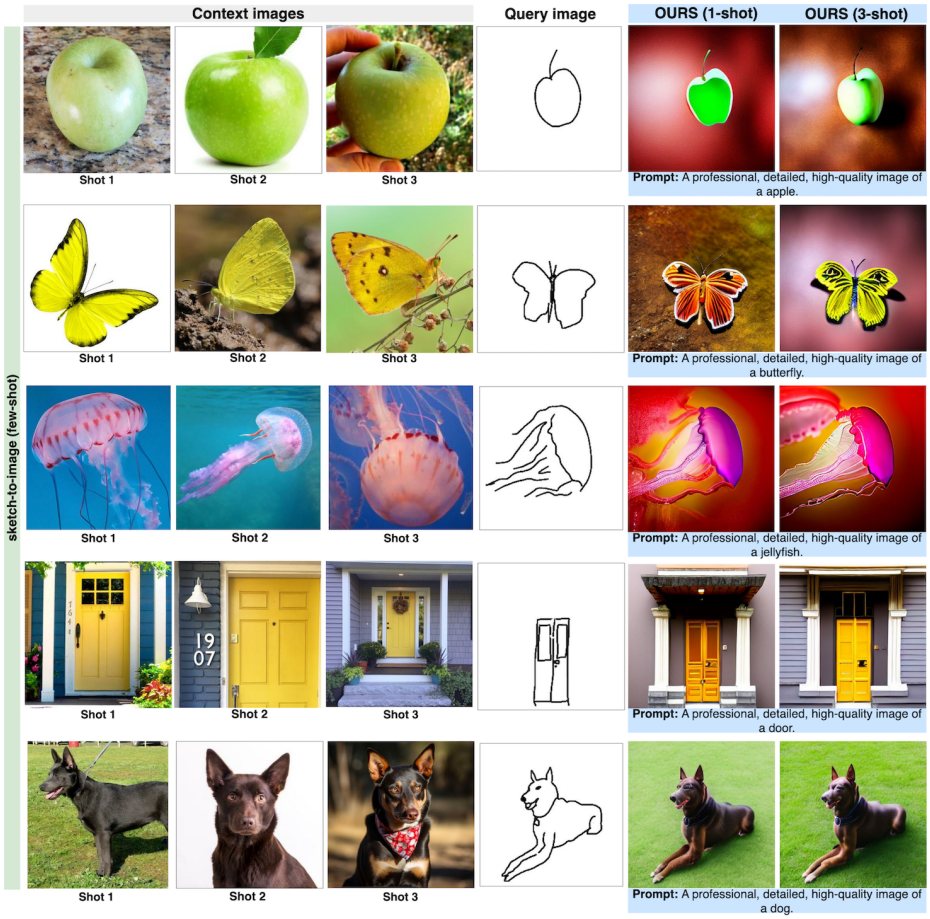


Fig. D.15: Few-shot examples: sketch-to-image. Comparison by using one, two, and three shots of context examples with text prompts.

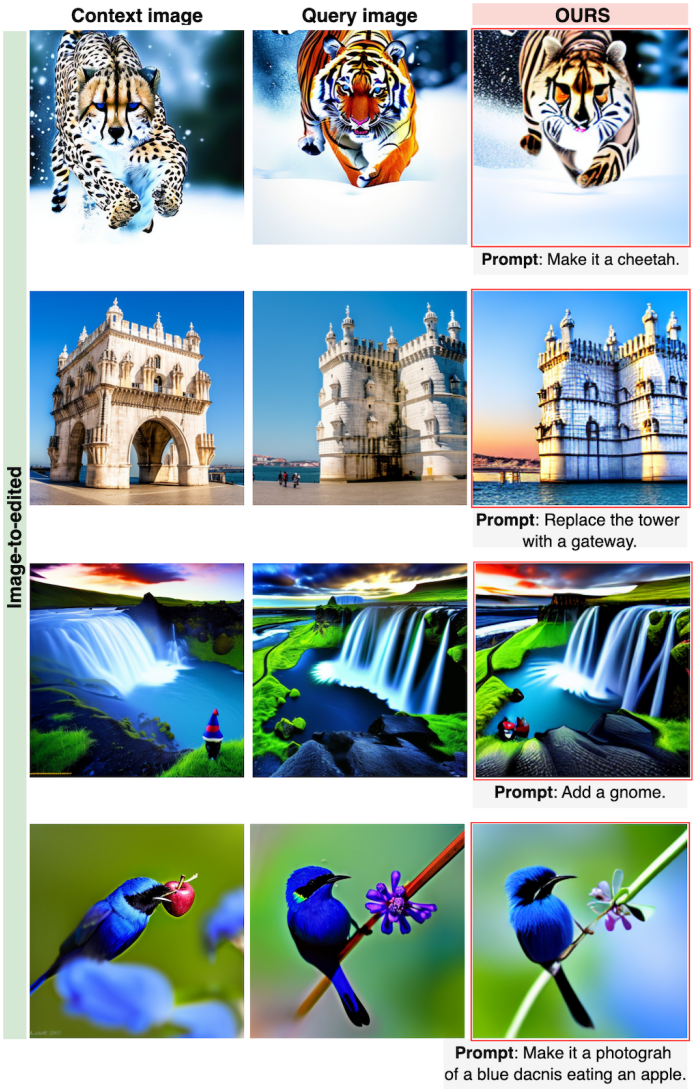


Fig. D.16: Failure examples on editing tasks (local edits), using visual context and prompt as conditioning.