ObjectDrop: Bootstrapping Counterfactuals for Photorealistic Object Removal and Insertion Supplementary Materials

1 Training and inference

Object removal In our object removal training process, we utilize a pre-trained text-to-image latent diffusion model (LDM) that was further trained for inpainting. Given an image of an object ("factual") and its mask, we fine-tune the LDM to denoise an image of the same scene without the object (the counterfactual image). We performed 50,000 optimization steps with batch size of 128 images and learning rate of $1e$–5.

Object insertion To train our object insertion model, we first fine-tune the model using a synthetic dataset as described in Section 5. This initial training phase consists of 100,000 optimization steps, employing a batch size of 512 images and a learning rate of $5e$–5. Subsequently, we fine-tune the model on our original counterfactual dataset for an additional 40,000 steps, with a batch size of 128 and decaying learning rates.
The denoiser function $D_\theta(x_t, x_{cond}, m, t, p)$ receives the following inputs:

- x_t: Noised latent representation of the image containing the object.
- x_{cond}: Latent representation of the object pasted onto a background image as is, without its effects on the scene.
- m: Mask indicating the object's location.
- t: Timestamp.
- p: Encoding of an empty string (text prompt).

1.1 Inference

All images in this paper were generated at a resolution of 512×512, with 50 denoising steps.

2 Bootstrapping

The bootstrapping procedure for creating the object insertion training set, as outlined in Section 5, follows these steps: We begin with an external dataset of 14 million images and extract foreground segmentation for each. We filter out images where the foreground mask covers less than 5% or more than 50% of the total image area, aiming to exclude objects that are either too small or too large. Additionally, we eliminate images where the foreground object extends

across more than 20% of the lower image boundary, as the shadow or reflection of these objects is often not visible within the image. This filtering process results in 700,000 images potentially containing suitable objects for removal. Using our object removal model, we generate predicted background images. However, in many of the original images, the object does not have a significant shadow or reflection, so that the difference between the synthesized input and output pairs consists of noise. To address this, we further discard images where the area showing significant differences between the object image and the predicted background is too small. This yields our final bootstrapped dataset of 350,000 examples.

3 Evaluation datasets

To assess our object insertion model, we employed two datasets. The first, referred to as the held-out dataset, comprises 51 triplets of photos taken after the completion of the project. Each triplet consists of: (1) a scene without the object, (2) the same scene with an added object, and (3) another image of the same scene and the same object placed elsewhere. We automatically segmented [6] the added object and relocated it within the image by naively pasting it on the background scene image. The model's inputs consist of the image with the pasted object and its corresponding mask. This dataset, along with our results, is presented in Fig. 4. With ground truth images illustrating how object movement should appear, we conducted quantitative metric assessments and user studies. Additionally, we used this dataset for evaluating the object removal model. In this test, we removed the object and compared the generated image to the ground truth background image.

The second test set, utilized for object insertion, comprises 50 examples, including some out-of-distribution images intended for moving large objects, as shown in Fig. 3. As this dataset lacks ground truth images, we used it solely for user study.

4 User study

To assess the effectiveness of our object removal model, we conducted a user study using the test set provided by Emu Edit of 264 examples, as shown in Fig. 2. We compared our results separately with those of Emu Edit and MGIE. Utilizing the CloudResearch platform, we collected user preferences from 50 randomly selected participants. Each participant reviewed 30 examples consisting of an original image, removal instructions, and the outcomes produced by both our method and the baseline. We randomized both the order of the examples shown and the order of each model in each example. To improve the reliability of the responses, we duplicated a few questions, and removed questionnaires that showed inconsistency for those repeated questions. A similar user study was carried out to compare our object insertion model with AnyDoor and Paint-by-Example, using the datasets described in Section 3. Different participants were

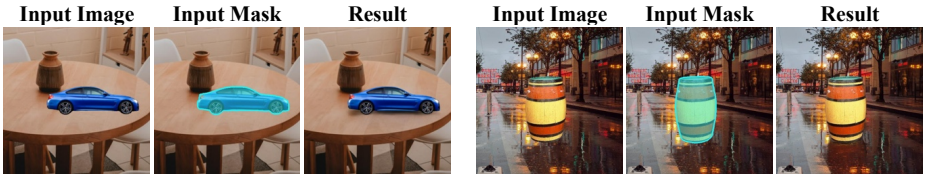

Fig. 1: Limitations of ObjectDrop: we focus on simulating the effect that an object has on the scene, but not the effect of the scene on the object. Consequently, we do not change the pose or lighting of the inserted object.

used for each dataset and comparison with baselines. The majority of participants were located in the United States. Participants were compensated above the minimum wage.

5 Self-supervision limitations

Attention-based methods, such as prompt-to-prompt [3], use a sophisticated heuristic based on cross-attention which sometimes overcomes the failure modes of inpainting. However, as they bias the generative model $P(X_o|X_e)$, they can result in unrealistic edits, sometimes removing the object but not its shadows. Also, the attention masks often fail to capture all scene pixels affected by the object, resulting in similar failures as inpainting. Note that Emu Edit [7] uses synthetic data created by an attention-based method for object removal and can therefore suffer from similar failure modes.

While class-guided disentanglement methods [1,2] attempt to solve the disentanglement task from observational data by assuming perfect knowledge of one of the hidden variables, and assuming that the object and scene are independent. Both assumptions are not sound in this setting, as the properties of the physical object and scene are not known perfectly, and only some objects are likely in a particular scene. Note that the generative mechanism is not perfectly identifiable even when the assumptions are satisfied [4, 5]. This motivates our search for a more grounded approach as will be described in the following sections.

Fig. 2: Additional examples for comparison with general editing methods, Emu Edit and MGIE. In this comparison, we utilized a text-based segmentation model to generate a mask for the object based on given instructions, which was then used as input for our model.

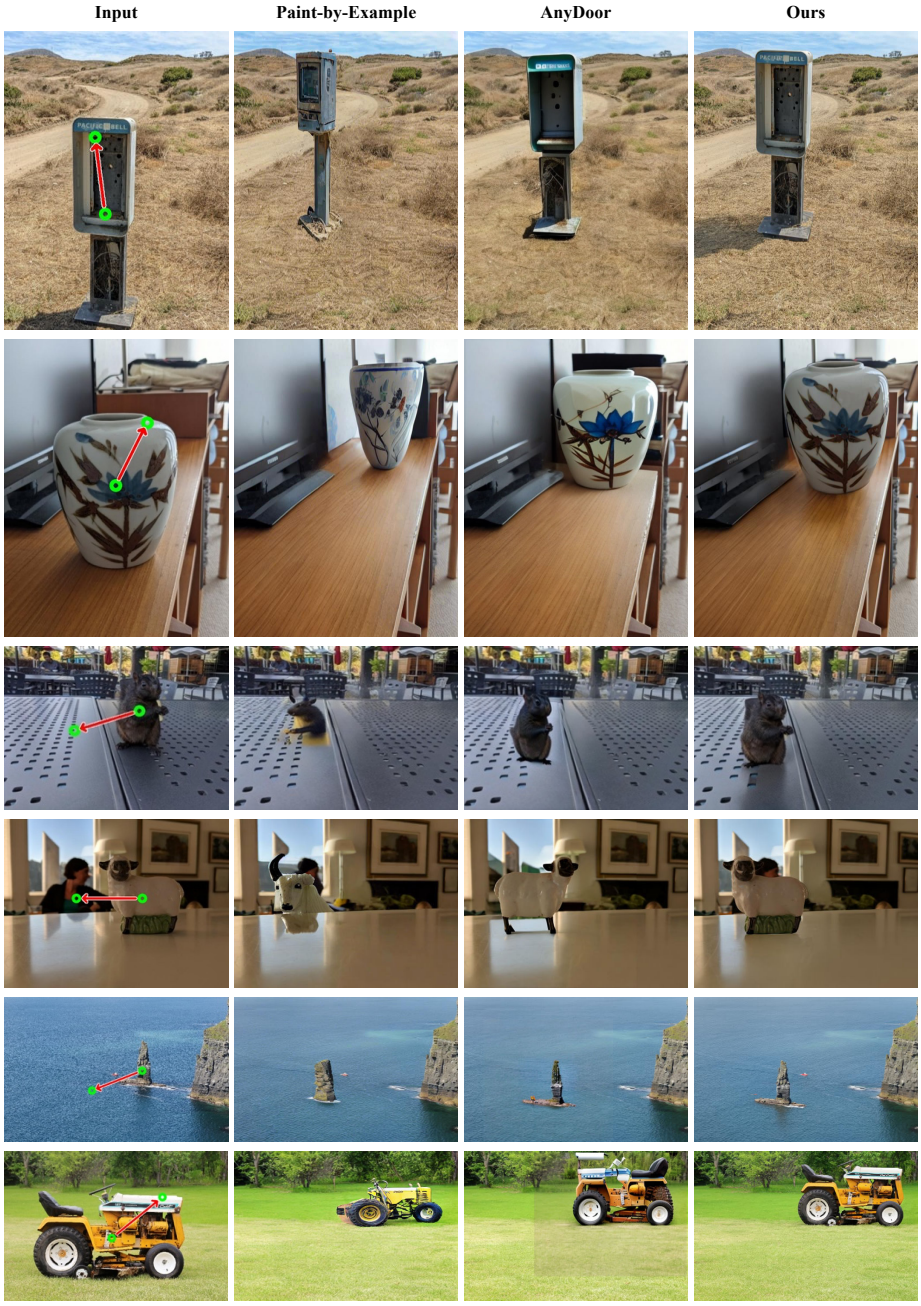

Fig. 3: Additional examples of intra-image object insertion.

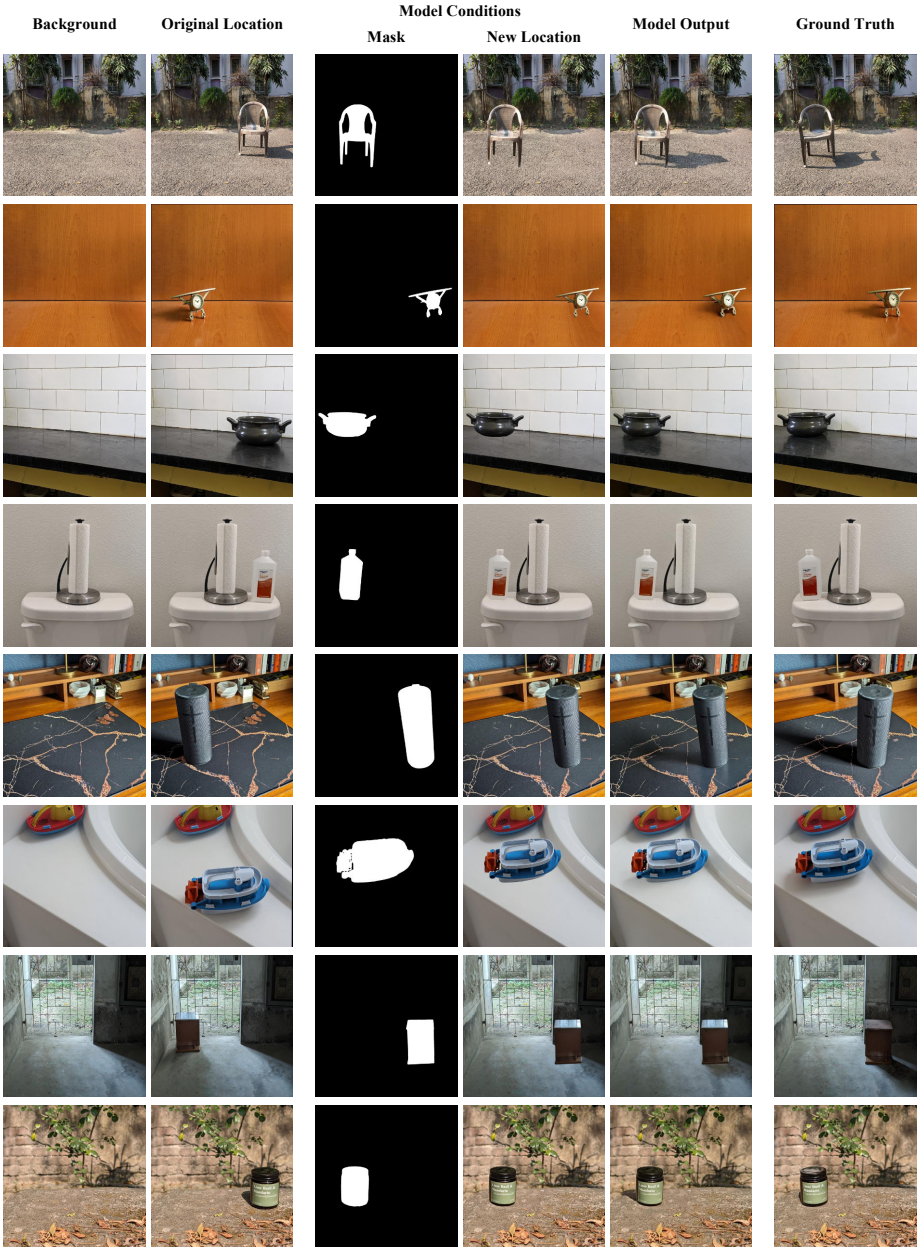

Fig. 4: Our held-out test set. The object insertion model uses two conditions: (1) an image where the object was pasted naively on the background and (2) a mask of that object.

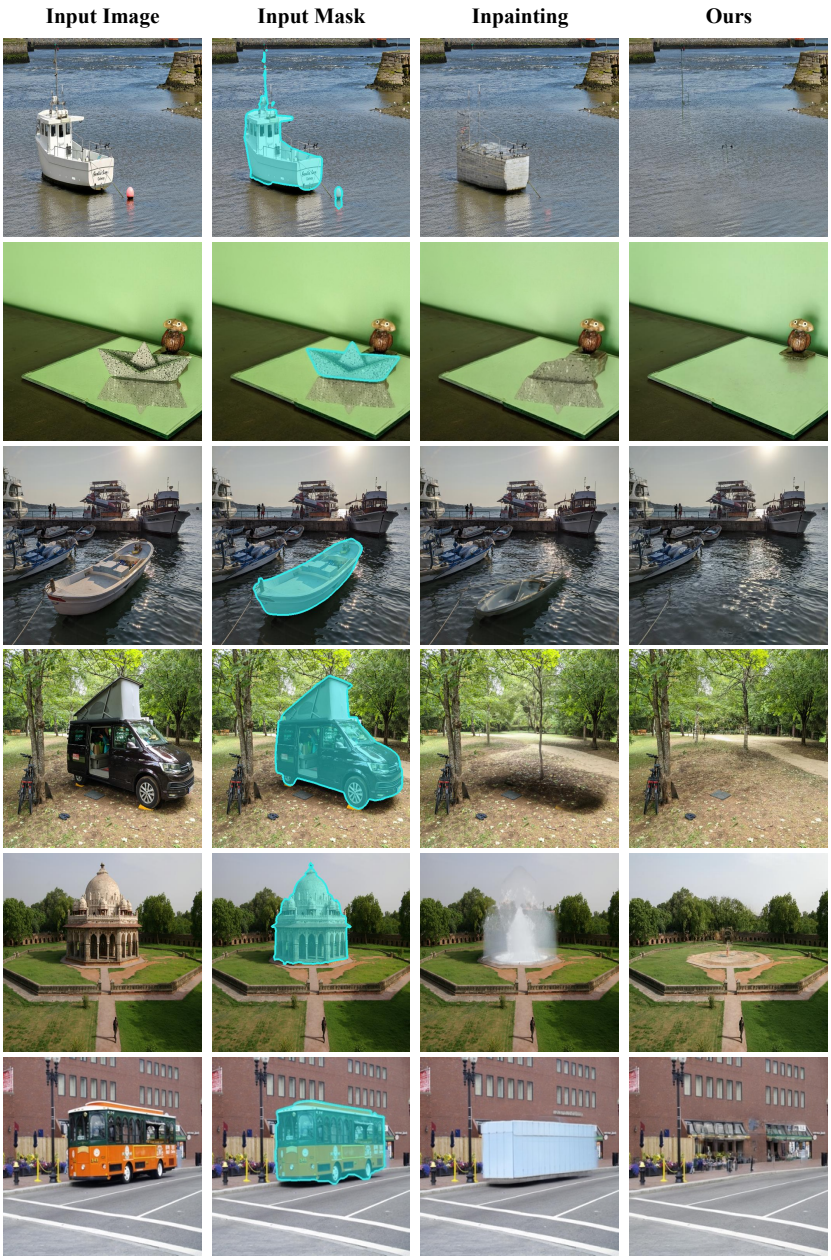

Fig. 5: Additional examples showcasing the performance of our object removal model compared to the inpainting model we initialized our model with.

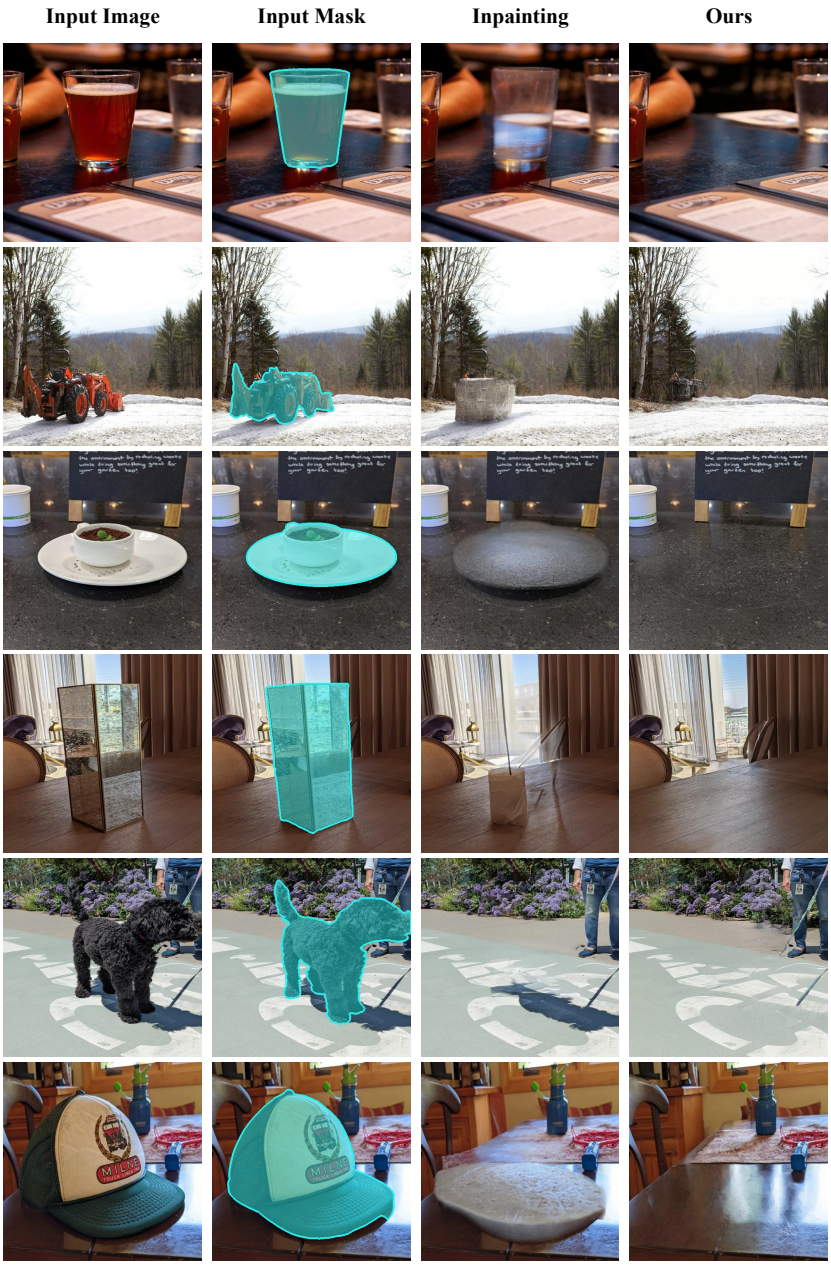

Fig. 6: Additional examples showcasing the performance of our object removal model compared to the inpainting model we initialized our model with.

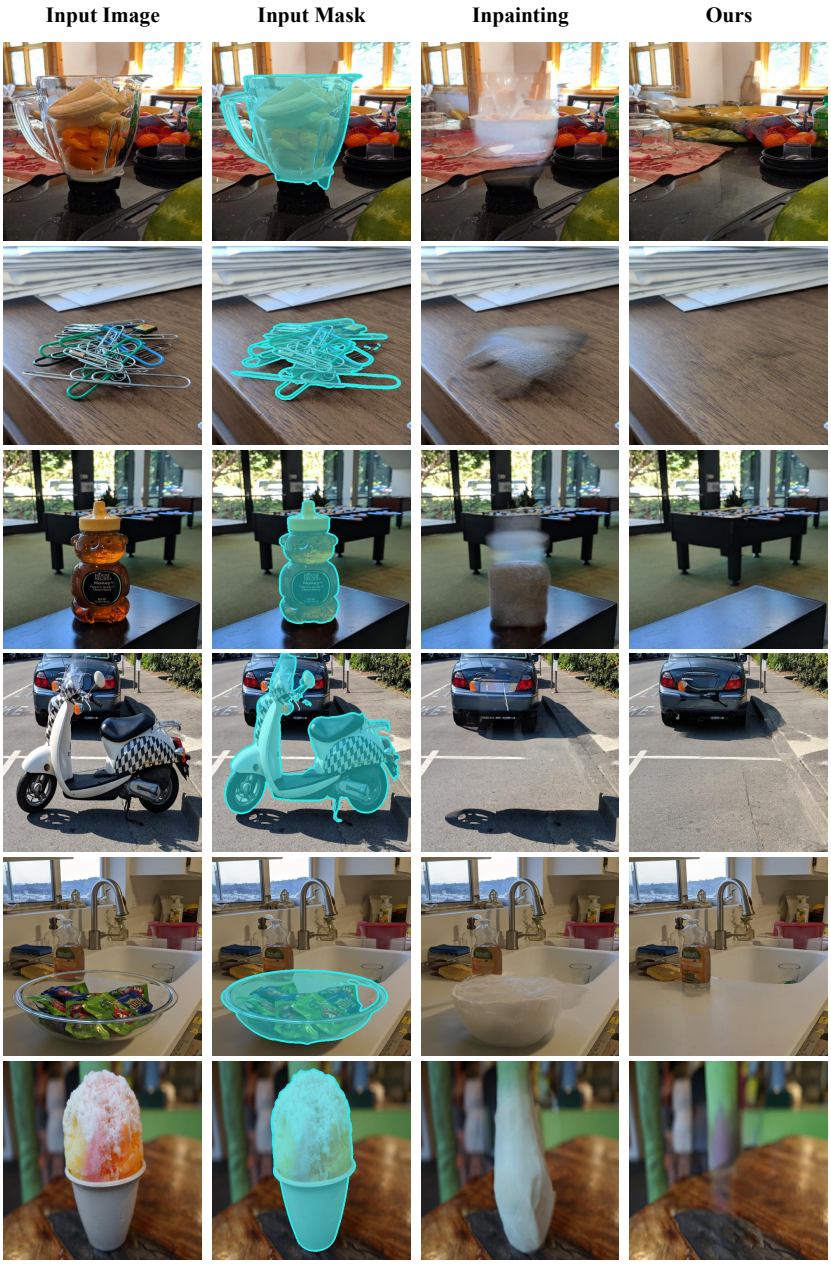

Fig. 7: Additional examples showcasing the performance of our object removal model compared to the inpainting model we initialized our model with.

References

1. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: Stargan v2: Diverse image synthesis for multiple domains. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8188–8197 (2020)
2. Gabbay, A., Cohen, N., Hoshen, Y.: An image is worth more than a thousand words: Towards disentanglement in the wild. Advances in Neural Information Processing Systems **34**, 9216–9228 (2021)
3. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626 (2022)
4. Kahana, J., Hoshen, Y.: A contrastive objective for learning disentangled representations. In: European Conference on Computer Vision. pp. 579–595. Springer (2022)
5. Khemakhem, I., Kingma, D., Monti, R., Hyvarinen, A.: Variational autoencoders and nonlinear ica: A unifying framework. In: International Conference on Artificial Intelligence and Statistics. pp. 2207–2217. PMLR (2020)
6. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
7. Sheynin, S., Polyak, A., Singer, U., Kirstain, Y., Zohar, A., Ashual, O., Parikh, D., Taigman, Y.: Emu edit: Precise image editing via recognition and generation tasks. arXiv preprint arXiv:2311.10089 (2023)