ObjectDrop: Bootstrapping Counterfactuals for Photorealistic Object Removal and Insertion

Daniel Winter[1,2], Matan Cohen[1], Shlomi Fruchter[1], Yael Pritch[1], Alex Rav-Acha[1], and Yedid Hoshen[1,2]

[1] Google Research
[2] The Hebrew University of Jerusalem
https://objectdrop.github.io

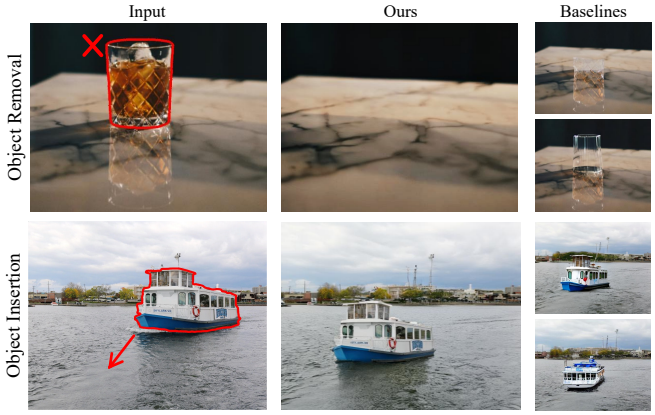

Fig. 1: Object removal and insertion. Our method models the effects of an object on the scene including occlusions, reflections, and shadows, enabling photorealistic object removal and insertion. It significantly outperforms state-of-the-art baselines.

Abstract. Diffusion models have revolutionized image editing but often generate images that violate physical laws, particularly the effects of objects on the scene, e.g., occlusions, shadows, and reflections. By analyzing the limitations of self-supervised approaches, we propose a practical solution centered on a "counterfactual" dataset. Our method involves capturing a scene before and after removing a single object, while minimizing other changes. By fine-tuning a diffusion model on this dataset, we are able to not only remove objects but also their effects on the scene. However, we find that applying this approach for photorealistic object insertion requires an impractically large dataset. To tackle this challenge, we propose bootstrap supervision; leveraging our object removal model trained on a small counterfactual dataset, we synthetically expand this dataset considerably. Our approach significantly outperforms prior methods in photorealistic object removal and insertion, particularly in modeling the effects of objects on the scene.

1 Introduction

Photorealistic image editing requires both visual appeal and physical plausibility. While diffusion-based editing models have significantly enhanced aesthetic quality, they often fail to generate physically realistic images. For instance, object removal methods must not only replace pixels occluded by the object but also model how the object affected the scene, e.g., removing shadows and reflections. Current diffusion methods frequently struggle with this, highlighting the need for better modeling of the effects of objects on their scenes.

Object removal and insertion is a long-standing but challenging task. As classical image editing methods could not tackle the full task, they targeted specific aspects, e.g., removing hard shadows. The advent of text-to-image diffusion models enabled new image editing techniques that perform more general edits.

We analyze the limitations of self-supervised editing approaches through the lens of counterfactual inference. A counterfactual statement [24] takes the form "if the object did not exist, this reflection would not occur". Accurately adding or removing the effect of an object on its scene requires understanding what the scene would look like with and without the object. Self-supervised approaches rely solely on observations of existing images, lacking access to counterfactual images. Disentanglement research [15,19,30] highlights that it is difficult to identify and learn the underlying physical processes from this type of data alone, leading to incorrect edits. This often manifests as either incomplete object removal or physically implausible changes to the scene.

Here, we propose a practical approach that trains a diffusion model on a meticulously curated "counterfactual" dataset. Each sample includes: i) a factual image depicting the scene, and ii) a counterfactual image depicting the scene after an object change (e.g., adding/removing it). We create this dataset by physically altering the scene; a photographer captures the factual image, alters the scene (e.g., removes an object), and then captures the counterfactual image. This approach ensures that each example reflects only the scene changes related to the presence of the object instead of other nuisance factors of variation.

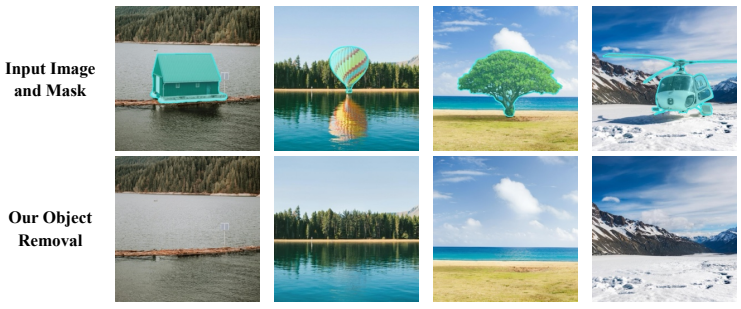

Fig. 2: Generalization. Our counterfactual dataset is relatively small and was captured in controlled settings, yet the model generalizes exceptionally well to out-of-distribution scenarios, such as removing buildings and large objects.

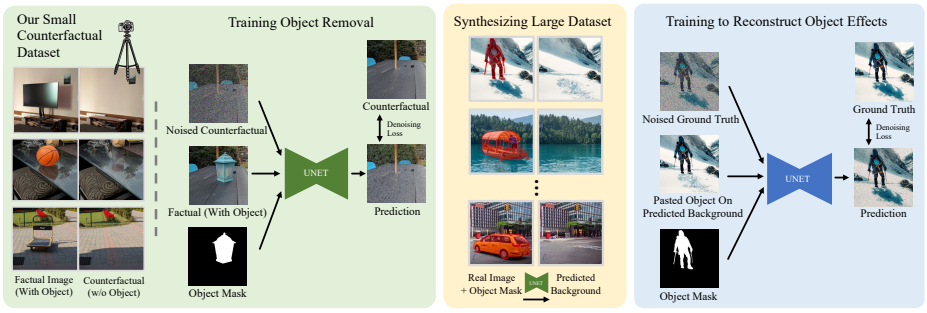

Fig. 3: Overview of our method. We collect a *counterfactual* dataset consisting of photos of scenes before and after removing an object, while keeping everything else fixed. We used this dataset to fine-tune a diffusion model to remove an object and all its effects from the scene. For the task of object insertion, we *bootstrap* a bigger dataset by removing selected objects from a large unsupervised image dataset, resulting in a vast, synthetic counterfactual dataset. Training on this synthetic dataset and then fine tuning on a smaller, supervised dataset yields a high-quality object insertion model.

We find this technique highly effective for object removal, surprisingly even for large or inaccessible objects that were not seen in training (see Fig. 2). However, given its limited size, the same dataset proved insufficient for training the reverse task of modeling how a newly inserted object affects the scene. We hypothesize that object insertion, which requires synthesizing shadows and reflections rather than merely removing them, is inherently more complex. We expect this to require a dataset too large for us to collect.

To address this, we propose a two-step approach. First, we train an object removal model using a smaller counterfactual dataset. Second, we apply the removal model on a large unlabeled image dataset to create a vast synthetic dataset. We finetune a diffusion model on this large dataset to add realistic shadows and reflections around newly inserted objects. We term this approach *bootstrap supervision*.

Our approach, ObjectDrop, achieves unprecedented results for both adding and removing the effects of objects. We show that it compares favorably to recent approaches such as Emu Edit, AnyDoor, and Paint-by-Example. Our contributions are:

1. An analysis of the limitations of self-supervised training for editing the effects of objects on scenes, such as shadows and reflections.
2. A counterfactual supervised training method for photorealistic object removal.
3. Bootstrap supervision to mitigate the labeling burden for object insertion.

1.1 Related work

Image inpainting. The task of inpainting missing image pixels has been widely explored in the literature. For several years, deep learning methods used genera-

tive adversarial networks [10], e.g., [14,28,35,37,40,55]. Several works use end-to-end learning methods [16,26,47,53]. More recently, the impressive advancements of diffusion models [39,41,44,45], have helped spur significant progress in inpainting [1,32,34,42,51]. We show (Sec. 3) that despite the great progress in the field and using very powerful diffusion models, these methods are not sufficient for photorealistic object removal.

Shadow removal methods. Another line of work focuses on the sub-task of shadow removal. In this task, the model aims to remove the shadow from an image given the shadow mask. Various methods [5,7,8,13,18,22,23,29,49,50,59,60] have been proposed for shadow removal. More recent methods [11,33] used latent diffusion models. Unlike these methods that remove only shadows, our method aims to remove all effects of the object on the scene, including occlusions and reflections. Also, these methods require a shadow segmentation map [5,52], while our method only requires an object segmentation map, which is easy to extract automatically, e.g., [20]. OmniMatte [31] aimed to recover both shadows and reflections; however, it requires video, whereas this paper deals with images.

General image editing models. An alternative approach for removing objects from photos is to use a general-purpose text-based image editing model [2,3,9,43,57]. For example, Emu Edit [43] trains a diffusion model on a large synthetic dataset to perform different editing tasks given a task embedding. MGIE [9] utilizes a diffusion model coupled with a Multimodal Large Language Model (MLLM) [27,48] to enhance the model's cross-modal understanding. While the breadth of the capabilities of these methods is impressive, our method outperformed them convincingly on object removal.

Object insertion. Earlier methods for inserting an object into a new image used end-to-end Generative Adversarial Networks (GANs) such as Pix2Pix [17], ShadowGAN [58], ARShadowGAN [25], and SGRNet [12]. Recent studies used diffusion models. Paint-by-Example [54] and ObjectStitch [46] insert a reference object into an image using the guidance of an image-text encoder [38], but only preserve semantic resemblance to the inserted object. AnyDoor [4] used a self-supervised representation [36] of the reference object alongside its high-frequency map as conditions to enhance object identity preservation. While the fidelity of generated images produced by AnyDoor improved over former methods, it sometimes changes object identity entirely, while we keep it unchanged by design. Furthermore, previous methods often do not model object reflections and shadows accurately, leading to unrealistic outcomes.

2 Task definition

We consider the input image X depicting a physical 3D scene S. We want our model to generate how the scene would have looked, had an object O been added to or removed from it. We denote this the *counterfactual* outcome. In other words, the task is to re-render the counterfactual image X^{cf}, given the physical change. The challenge is to model the effects of the object change on the scene, such as occlusions, shadows, and reflections. We denote the physical

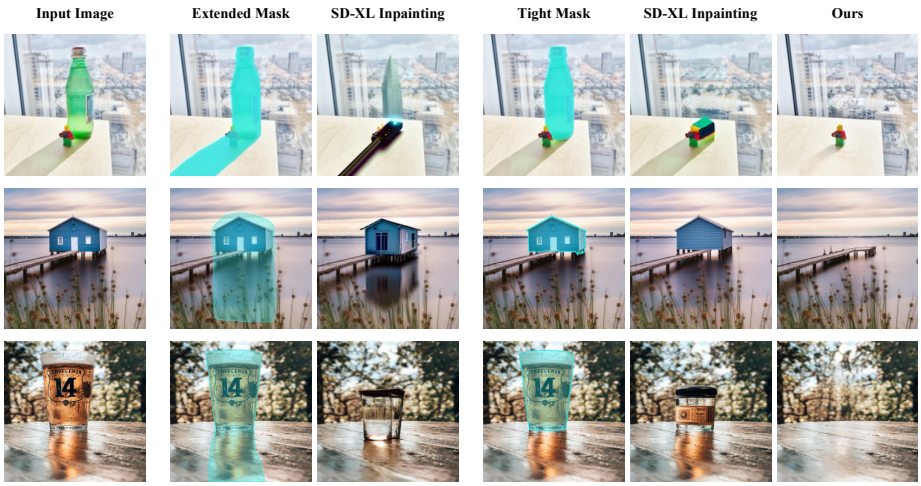

Fig. 4: Object removal - comparison with inpainting. Our model successfully removes the masked object, while the baseline inpainting model [6] replaces it with a different one. Using a mask that covers the reflections (extended mask) may obscure important details from the model.

rendering mechanism as $G_{physics}$; the input image X is given by:

$$X = G_{physics}(O, S) \tag{1}$$

The desired output is the counterfactual image X^{cf} s.t.,

$$X^{cf} = G_{physics}(O^{cf}, S) \tag{2}$$

For object removal, an object o is originally present ($O = o$) and we wish to remove the object so that $O^{cf} = \phi$ (ϕ is the empty object). For object insertion, the object is initially absent ($O = \phi$) and we wish to add it ($O^{cf} = o$).

While the physical rendering mechanism is relatively well understood, we cannot directly use the formulation in Eq. 1 for editing as it requires perfect knowledge of the physical object and scene, which is rarely available.

3 Self-supervision is not enough

Diffusion models generate photorealistic image distributions. However, they are not a natural fit for adding or removing objects from images, as they do not provide a direct way to intervene on their physical variables. Here, these variables are the physical object O and the properties of the scene S. The task of inferring hidden variables (e.g., O and S) and the generative mechanism (e.g., $G_{physics}$) is called disentanglement. Several influential works [15, 19, 30] established that unsupervised disentanglement from observational data is generally impossible without strong priors.

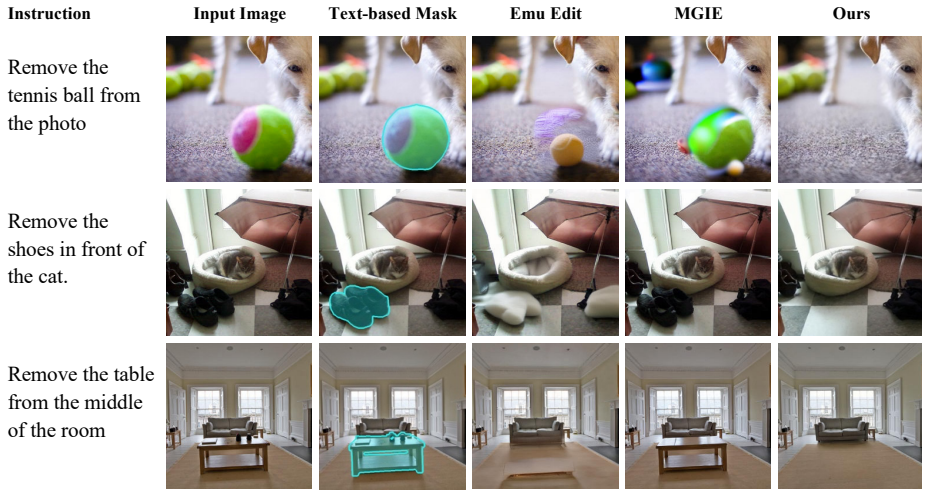

Fig. 5: Object removal - comparison with general editing methods. We compare to general editing methods: Emu Edit [43] and MGIE [9]. These methods often replace the object with a new one and introduce unintended changes to the input image. For this comparison, we used a text-based segmentation model to mask the object according to the instruction and passed the mask as input to our model.

Self-supervised methods attempt to perform disentanglement using heuristic schemes. One common heuristic for object removal is to use diffusion-based inpainting. Such methods use a segmentation map to split the image pixels into two disjoint subsets: a subset of pixels that contain the object X_o and a subset of those that does not X_s, such that $X = X_o \cup X_s$. Inpainting resamples the values of the object pixels given the scene:

$$x_o \sim P(X_o|x_s) \tag{3}$$

This approach depends on the segmentation mask, which is a limitation because: i) if the mask is chosen too tightly around the object, then X_s includes object shadows and reflections and thus has information about the object. The most likely value of $P(X_o|x_s)$ will contain an object that renders similar shadows and reflections as the original, which is likely a similar object to the original. ii) If the mask is wide and removes all scene pixels that are affected by the object, it will not preserve the original scene. We show both failure modes in Fig. 4.

4 Object removal

In this section, we propose ObjectDrop, a new approach based on counterfactual supervision for object removal. As mentioned in Sec. 3, it is insufficient to merely model the observed images; we must also take into account their causal hidden variables, i.e., the object and the scene. Since we cannot learn these purely from

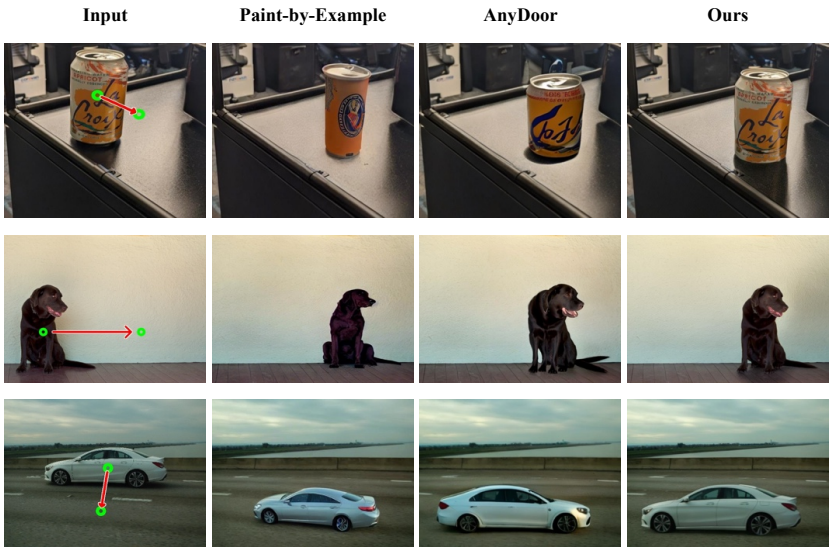

Fig. 6: Intra-image object insertion - baseline comparison. We preserve the object identity better and achieve more photorealistic shadows and reflections than baselines Paint-by-Example and AnyDoor.

observing images, we propose to directly act in the physical world to estimate the counterfactual effect of removing objects.

4.1 Collecting a counterfactual dataset

The key to unlocking such models is by creating a counterfactual dataset. The procedure consists of three steps:

1. Capture an image X ("factual") containing the object O in scene S.
2. Physically remove the object O while avoiding camera movement, lighting changes, or motion of other objects.
3. Capture another image X^{cf} ("counterfactual") of the same scene but without the object O.

We use an off-the-shelf segmentation model [20] to create a segmentation map M_o for the object O removed from the factual image X. The final dataset contains input pairs of factual images and binary object masks $(X, M_o(X))$, and the output counterfactual images X^{cf}.

In practice, we collected 2,500 such counterfactual pairs. This number is relatively small due to the high cost of data collection. The images were collected by professional photographers using a tripod-mounted camera to keep the camera pose as stable as possible. Since the counterfactual pairs have (almost) exactly the same camera pose, lighting, and background objects, the only difference between the factual and counterfactual images is the removal of the object.

Advantages over video supervision. Previous approaches, such as AnyDoor [4] and [21], reconstruct a masked object in one video frame by observing the object in another frame. While this procedure is cheaper, it has serious limitations: i) In a counterfactual dataset, removing the object should be the only difference between the frames, but in video, many other attributes also change, such as camera view. This leads to spurious correlations between object removal and other attributes. ii) This procedure only works for dynamic objects (cars, animals, etc.) and cannot collect samples for inanimate objects.

4.2 Counterfactual distribution estimation

Given our high-quality counterfactual dataset, our goal is to estimate the distribution of the counterfactual images $P(X^{cf}|X = x, M_o(x))$, given the factual image x and segmentation mask. We achieve this by fine-tuning a diffusion model on our counterfactual dataset. We investigate the impact of using different foundational diffusion models in Sec. 6. The estimation is done by minimizing:

$$\mathcal{L}(\theta) = \mathop{\mathbb{E}}_{t\sim U([0,T]),\epsilon\sim\mathcal{N}(0,I)} \left[\sum_{i=1}^{N} \| D_\theta(\alpha_t x_i^{cf} + \sigma_t \epsilon, x_i, M_o(x_i), t, p) - \epsilon \|^2 \right] \quad (4)$$

where $D_\theta(\tilde{x_t}, x_{cond}, m, t, p)$ is a denoiser network with following inputs: noised latent representation of the counterfactual image $\tilde{x_t}$, latent representation of the image containing the object we want to remove x_{cond}, mask m indicating the object's location, timestamp t, and encoding of an empty string (text prompt) p. Here, x_t is calculated based on the forward process equation:

$$\tilde{x_t} = \alpha_t \cdot x + \sigma_t \cdot \epsilon \quad (5)$$

where x represents the image without the object (the counterfactual), α_t and σ_t are determined by the noising schedule, and $\epsilon \sim \mathcal{N}(0, I)$.

Importantly, unlike traditional inpainting methods, we avoid replacing the pixels of the object with uniform gray or black pixels. This approach allows our model to leverage information preserved within the mask, which is particularly beneficial in scenarios involving partially transparent objects or imperfect masks.

5 Object insertion

We extend ObjectDrop to object insertion. In this task, we are given an image of an object, a desired position, and a target image. The objective is to predict how the target image would look if it had been photographed with the given object. While collecting a relatively small-scale (2,500 samples) counterfactual dataset was successful for object removal, we observed that this dataset is insufficient for training an object insertion model (see Fig. 8). We hypothesize that this requires more examples, as synthesizing the shadows and reflections of the object may be more challenging than removing them.

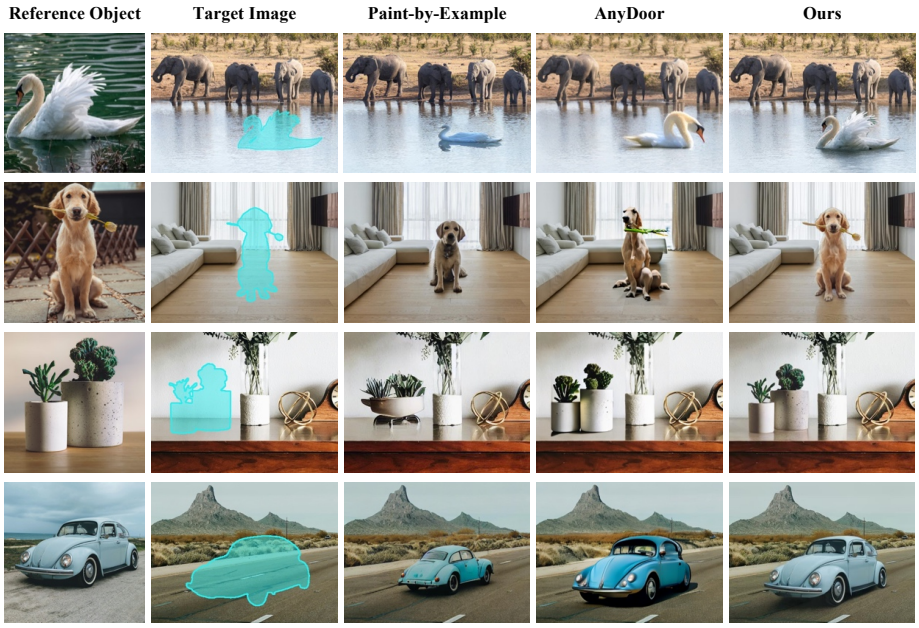

Fig. 7: Cross-image object insertion. Similar to the results of intra-image object insertion, our method preserves object identity better and synthesizes more photorealistic shadows and reflections than the baselines.

5.1 Bootstrapping counterfactual dataset

We propose to leverage our small counterfactual dataset towards creating a large-scale counterfactual object insertion dataset. We take a large external image dataset $x_1, x_2, \ldots, x_n$ and detect object masks $M_o(x_1), M_o(x_2), \ldots, M_o(x_n)$ using a foreground detector. We remove each object and its effects on the scene using our removal model, denoting the output by $z_1, z_2, \ldots, z_n$:

$$z_i \sim P(X^{cf} | x_i, M_o(x_i)) \tag{6}$$

Finally, we paste each object into the object-less scenes z_i, resulting in images without shadows and reflections:

$$y_i = M_o(x_i) \odot x_i + (1 - M_o(x_i)) \odot z_i. \tag{7}$$

The synthetic dataset consists of a set of input pairs $(y_i, M_o(x_i))$. The corresponding targets are the original images x_i. To clarify, both the input and output images contain the object o_i, but the input images do not contain the effects of the object on the scene, while the output images do. The task of the model is to generate the effects as illustrated in Fig. 3.

In practice, we start with a dataset consisting of 14M images and select 700k images with suitable objects. We run object removal on each image and further

Fig. 8: Bootstrapping ablation. Bootstrap supervision improves model quality.

filter approximately half of them that did not have significant object effects on the scene. The final bootstrapped dataset consists of 350K images, around 140 times larger than the manually labeled dataset. More details about the filtering process are provided in the supplementary materials.

5.2 Diffusion model training

We use the bootstrapped counterfactual dataset to train an object insertion model with the diffusion objective presented in Eq. 4.2. In contrast to the object removal process, we use a pre-trained text-to-image model $D_\theta(x, t, p)$ that did not undergo inpainting pre-training. As the input mask increases input dimension, we add new channels to the input of the pre-trained text-to-image model, initializing their weights with 0.

5.3 Fine-tuning on the ground truth counterfactual dataset

The synthetic dataset is not realistic enough for training the final model and is only used for pre-training. In the last stage, we fine-tune the model on the original ground truth counterfactual dataset that was manually collected. While this dataset is not large, pre-training on the bootstrapped dataset is powerful enough to enable effective fine-tuning using this small ground truth dataset.

6 Experiments

6.1 Implementation details

Counterfactual dataset. We created a counterfactual dataset of 2,500 pairs of photos using the procedure detailed in Sec. 4. Each pair contains a "factual" image of a scene and a second "counterfactual" image of exactly the same scene except that it was photographed after removing one object. We also held out

Table 1: Object removal - reconstruction metrics. A comparison with the inpainting baseline on the held-out test set.

Model	PSNR ↑	DINO ↑	CLIP ↑	LPIPS ↓
RePaint [32]	21.177	0.805	0.855	0.118
Inpainting	21.192	0.876	0.897	0.056
Ours	**23.153**	**0.948**	**0.959**	**0.048**

Table 2: Object insertion - reconstruction metrics. A comparison with baselines: Paint-by-Example and AnyDoor, on the held-out test set. Furthermore, we ablate the contribution of the bootstrap supervision.

Model	PSNR ↑	DINO ↑	CLIP ↑	LPIPS ↓
Paint-by-Example	17.523	0.755	0.862	0.138
AnyDoor	19.500	0.889	0.890	0.095
Ours w/o Bootstrap	20.178	0.929	0.945	0.066
Ours	**21.625**	**0.939**	**0.950**	**0.057**

100 counterfactual test examples, captured after the completion of the research, depicting new objects and scenes.

Model architecture. We train a latent diffusion model (LDM) [41] for the object removal task. We initialize it using a pre-trained inpainting model, which takes as input a factual image, an object mask, and a noisy counterfactual image. We perform inference using default settings. We use an internal model with a similar architecture to Stable-Diffusion-XL. Unlike other inpainting models, we do not replace the removed object pixels with gray pixels.

Quantitative metrics. We compared the results of our method and the baselines on the held-out counterfactual test set. As this dataset has ground truth (see supplementary), we used standard reconstruction metrics: both classical (PSNR) and deep perceptual similarity metrics using DINO [36], CLIP [38], and LPIPS [56] (AlexNet) features.

6.2 Object removal

Qualitative results. We evaluated our results on the benchmark published by Emu Edit [43]. As seen in Fig. 5, our model removes objects and their effects in a photorealistic manner. The baselines failed to remove shadows and reflections and sometimes adversely affected the image in other ways.

Quantitative results. Tab. 1 compares our method to the inpainting pre-trained model on the held-out test set using quantitative reconstruction metrics. Our method outperformed the baseline substantially.

User study. We conducted a user study on the benchmark by Emu Edit [43] between our method and baselines: Emu Edit and MGIE. As the benchmark does not have ground truth, a user study is the most viable comparison. We

Table 3: Object removal - user study. A comparison to Emu Edit [43] and MGIE [9] on the Emu Edit dataset.

Which model did a better job at following the object removal editing instruction?			
Preferred Emu Edit	35.9%	Preferred MGIE	13.5%
Preferred ours	64.1%	Preferred ours	86.5%

Table 4: Object insertion - user study. A comparison on in-distribution (ID) and out-of-distribution (OOD) intra-image object insertion datasets with the baselines AnyDoor [4] and PbE [54].

Held-Out Set (ID)				In-the-Wild (OOD)			
AnyDoor	11.1%	PbE	3.3%	AnyDoor	5.0%	PbE	2.8%
Ours	88.9%	Ours	96.7%	Ours	95.0%	Ours	97.2%

used the CloudResearch platform to gather user preferences from 50 randomly selected participants. Each participant was presented with 30 examples of an original image, removal text instructions, and results generated by our method and the baseline. Tab. 3 displays the results. Notably, our method surpassed both baseline methods in user preference.

6.3 Object insertion

Qualitative results. We compare our object insertion model with state-of-the-art image editing techniques, Paint-by-Example [54] and AnyDoor [4]. Fig. 6 shows intra-image insertions, i.e., when the objects are repositioned within the same image. For achieving intra-image insertions, we first use our object removal model to remove the object from its original position, obtaining the background image. For equitable comparisons, we used the same background image, obtained by our model, when comparing to the baselines. Fig. 7 shows inter-image insertions, i.e., when the objects come from different images. In both cases, our method synthesizes the shadows and reflections of the object better than the baselines. It also preserves the identity of the object, while other methods modify it freely and in many cases lose the original identity entirely.

Quantitative results. We compare to Paint-by-Example and AnyDoor on the held-out counterfactual test dataset. The results are presented in Tab. 2. Our method outperforms the baselines by a significant margin on all metrics.

User study. We also conducted a user study on 2 intra-image insertion datasets. The first is the held-out test set. The second is a set of 50 out-of-distribution images depicting more general scenes, some of which are very different from those seen in training, e.g., inserting boats and buildings. Tab. 4 shows that users overwhelmingly preferred our method over the baselines.

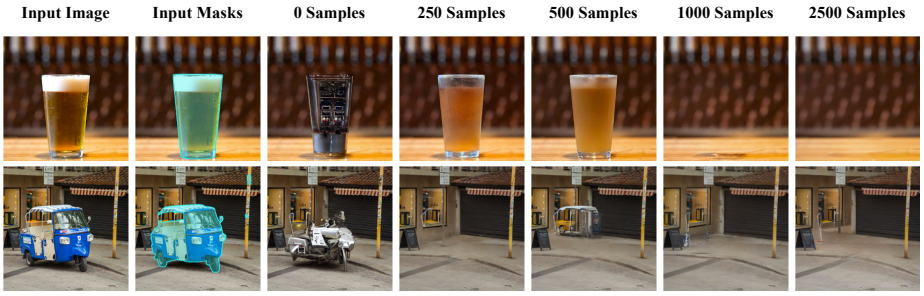

Fig. 9: Qualitative dataset size ablation. Increasing the size of the training dataset improves object removal performance.

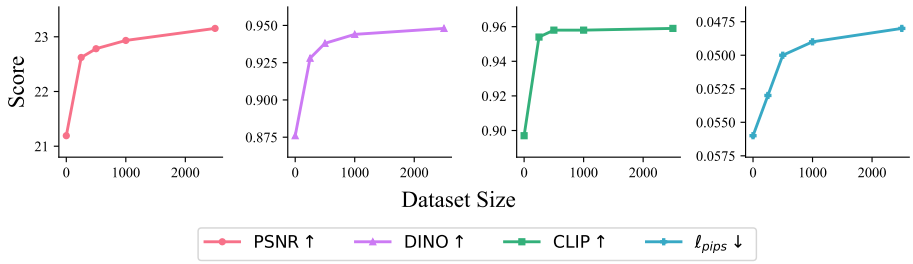

Fig. 10: Quantitative dataset size ablation. Even a small counterfactual dataset improves performance greatly.

6.4 Ablation study

Bootstrapping. We ablated the contribution of our bootstrapping method for object insertion. Here, we bootstrapped 2,500 real images into $350K$ synthetic images. In the ablation, we compare our full method to fine-tuning the original backbone on the original counterfactual dataset without bootstrapping. Both models used the same pre-training backbone. Both the qualitative results in Fig. 8 and the quantitative results in Tab. 2 clearly support bootstrapping.

Dataset size. Collecting large counterfactual datasets is expensive. We evaluate the influence of dataset size on the performance of our object removal method. We fine-tune the base model on subsets of the full counterfactual dataset with varying sizes. Fig. 9 and Fig. 10 illustrate the qualitative and quantitative effects of dataset size, respectively. Using the pre-trained inpainting model without fine-tuning ("0 samples") merely replaces the target object with another similar one, and its effects on the scene remain. The results start looking attractive around 1,000 examples, with more examples further improving performance.

Text-to-image vs. inpainting pre-trained model. Fig. 11 demonstrates that using a text-to-image (T2I) model instead of an inpainting model for pre-training obtains comparable quality for removing shadows and reflections. This shows that inpainting models do not have a better inductive bias for modeling object effects on scenes than T2I models. Unsurprisingly, the inpainting model

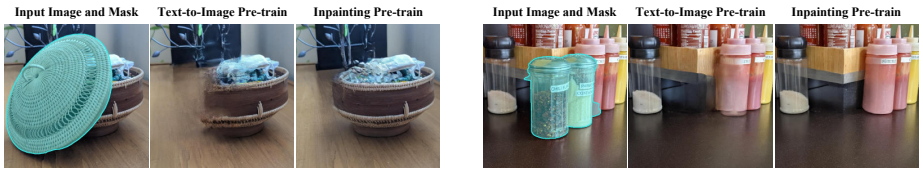

Fig. 11: Inpainting vs. Text-to-Image pre-training. The inpainting model has better results on pixels occluded by the objects, but results in comparable quality for removing or adding photorealistic reflections and shadows.

Table 5: Stable-Diffusion results. Our method works well on the public Stable-Diffusion-Inpainting v1 [41] model.

Model	PSNR ↑	DINO ↑	CLIP ↑	LPIPS ↓
SD Inpainting	19.198	0.775	0.884	0.083
Our fine-tuned SD	**21.363**	**0.876**	**0.930**	**0.076**

is better at inpainting the pixels occluded by the objects. Furthermore, we compared the pre-trained models (not shown) on object insertion. Consequently, we used the inpainting backbone for the object removal experiments and the T2I backbone for the object insertion experiments in the paper.

Public models. To verify that our method works on public pre-trained models, we trained our model using Stable-Diffusion-Inpainting v1 [41]. We then computed the quantitative metrics for object removal as in Sec. 6.2. Our results in Tab. 5 show that our method improves this pre-trained model significantly.

7 Limitations

This work focuses on simulating the effect that an object has on the scene, but not the effect of the scene on the object. As a result, our method may yield unrealistic results in scenarios where the orientation and lighting of the object are incompatible with the scene. This can be solved independently using existing harmonization methods, but this was not explored in the context of this work. Additionally, since our model does not know the physical 3D scene and lighting perfectly, it may produce realistic-looking but incorrect shadow directions. Fig. 1 in SM presents a visualization of these limitations.

8 Conclusion

We introduced ObjectDrop, a supervised approach for object removal and insertion. Our method relies on a counterfactual dataset, i.e., pairs of images taken before and after the physical manipulation of the object. We proposed bootstrap supervision to reduce the cost of counterfactual dataset collection. A comprehensive evaluation shows that our approach outperforms the state-of-the-art.

Acknowledgement

We would like to thank Gitartha Goswami, Soumyadip Ghosh, Reggie Ballesteros, Srimon Chatterjee, Michael Milne and James Adamson for providing the photographs that made this project possible. We thank Yaron Brodsky, Dana Berman, Amir Hertz, Moab Arar, and Oren Katzir for their invaluable feedback and discussions. We also appreciate the insights provided by Dani Lischinski and Daniel Cohen-Or, which helped improve this work.

References

1. Avrahami, O., Lischinski, D., Fried, O.: Blended diffusion for text-driven editing of natural images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18208–18218 (2022)
2. Bar-Tal, O., Ofri-Amar, D., Fridman, R., Kasten, Y., Dekel, T.: Text2live: Text-driven layered image and video editing. In: European conference on computer vision. pp. 707–723. Springer (2022)
3. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18392–18402 (2023)
4. Chen, X., Huang, L., Liu, Y., Shen, Y., Zhao, D., Zhao, H.: Anydoor: Zero-shot object-level image customization. arXiv preprint arXiv:2307.09481 (2023)
5. Cun, X., Pun, C.M., Shi, C.: Towards ghost-free shadow removal via dual hierarchical aggregation network and shadow matting gan. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 10680–10687 (2020)
6. Diffusers: Stable diffusion xl inpainting 0.1. `https://huggingface.co/diffusers/stable-diffusion-xl-1.0-inpainting-0.1` (2023)
7. Ding, B., Long, C., Zhang, L., Xiao, C.: Argan: Attentive recurrent generative adversarial network for shadow detection and removal. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10213–10222 (2019)
8. Fu, L., Zhou, C., Guo, Q., Juefei-Xu, F., Yu, H., Feng, W., Liu, Y., Wang, S.: Auto-exposure fusion for single-image shadow removal. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10571–10580 (2021)
9. Fu, T.J., Hu, W., Du, X., Wang, W.Y., Yang, Y., Gan, Z.: Guiding instruction-based image editing via multimodal large language models. arXiv preprint arXiv:2309.17102 (2023)
10. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)
11. Guo, L., Wang, C., Yang, W., Huang, S., Wang, Y., Pfister, H., Wen, B.: Shadowdiffusion: When degradation prior meets diffusion model for shadow removal. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14049–14058 (2023)
12. Hong, Y., Niu, L., Zhang, J.: Shadow generation for composite image in real-world scenes. In: Proceedings of the AAAI conference on artificial intelligence. vol. 36, pp. 914–922 (2022)
13. Hu, X., Jiang, Y., Fu, C.W., Heng, P.A.: Mask-shadowgan: Learning to remove shadows from unpaired data. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2472–2481 (2019)

14. Hui, Z., Li, J., Wang, X., Gao, X.: Image fine-grained inpainting. arXiv preprint arXiv:2002.02609 (2020)
15. Hyvärinen, A., Pajunen, P.: Nonlinear independent component analysis: Existence and uniqueness results. Neural networks **12**(3), 429–439 (1999)
16. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. ACM Transactions on Graphics (ToG) **36**(4), 1–14 (2017)
17. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)
18. Jin, Y., Sharma, A., Tan, R.T.: Dc-shadownet: Single-image hard and soft shadow removal using unsupervised domain-classifier guided network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5027–5036 (2021)
19. 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)
20. 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)
21. Kulal, S., Brooks, T., Aiken, A., Wu, J., Yang, J., Lu, J., Efros, A.A., Singh, K.K.: Putting people in their place: Affordance-aware human insertion into scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17089–17099 (2023)
22. Le, H., Samaras, D.: Shadow removal via shadow image decomposition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8578–8587 (2019)
23. Le, H., Samaras, D.: From shadow segmentation to shadow removal. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16. pp. 264–281. Springer (2020)
24. Lewis, D.K.: Counterfactuals. Blackwell, Malden, Mass. (1973)
25. Liu, D., Long, C., Zhang, H., Yu, H., Dong, X., Xiao, C.: Arshadowgan: Shadow generative adversarial network for augmented reality in single light scenes. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8139–8148 (2020)
26. Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: Proceedings of the European conference on computer vision (ECCV). pp. 85–100 (2018)
27. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36** (2024)
28. Liu, H., Jiang, B., Song, Y., Huang, W., Yang, C.: Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. pp. 725–741. Springer (2020)
29. Liu, Z., Yin, H., Wu, X., Wu, Z., Mi, Y., Wang, S.: From shadow generation to shadow removal. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4927–4936 (2021)
30. Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., Bachem, O.: Challenging common assumptions in the unsupervised learning of disentangled representations. In: international conference on machine learning. pp. 4114–4124. PMLR (2019)

31. Lu, E., Cole, F., Dekel, T., Zisserman, A., Freeman, W.T., Rubinstein, M.: Omnimatte: Associating objects and their effects in video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4507–4515 (2021)
32. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: Repaint: Inpainting using denoising diffusion probabilistic models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11461–11471 (2022)
33. Mei, K., Figueroa, L., Lin, Z., Ding, Z., Cohen, S., Patel, V.M.: Latent feature-guided diffusion models for shadow removal. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 4313–4322 (2024)
34. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: Sdedit: Guided image synthesis and editing with stochastic differential equations. arXiv preprint arXiv:2108.01073 (2021)
35. Ntavelis, E., Romero, A., Bigdeli, S., Timofte, R., Hui, Z., Wang, X., Gao, X., Shin, C., Kim, T., Son, H., et al.: Aim 2020 challenge on image extreme inpainting. In: Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16. pp. 716–741. Springer (2020)
36. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
37. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2536–2544 (2016)
38. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
39. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 **1**(2), 3 (2022)
40. Ren, Y., Yu, X., Zhang, R., Li, T.H., Liu, S., Li, G.: Structureflow: Image inpainting via structure-aware appearance flow. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 181–190 (2019)
41. 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 (CVPR). pp. 10684–10695 (June 2022)
42. Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., Norouzi, M.: Palette: Image-to-image diffusion models. In: ACM SIGGRAPH 2022 Conference Proceedings. pp. 1–10 (2022)
43. 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)
44. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International conference on machine learning. pp. 2256–2265. PMLR (2015)
45. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)

46. Song, Y., Zhang, Z., Lin, Z., Cohen, S., Price, B., Zhang, J., Kim, S.Y., Aliaga, D.: Objectstitch: Object compositing with diffusion model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18310–18319 (2023)
47. Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., Lempitsky, V.: Resolution-robust large mask inpainting with fourier convolutions. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 2149–2159 (2022)
48. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
49. Wan, J., Yin, H., Wu, Z., Wu, X., Liu, Y., Wang, S.: Style-guided shadow removal. In: European Conference on Computer Vision. pp. 361–378. Springer (2022)
50. Wang, J., Li, X., Yang, J.: Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1788–1797 (2018)
51. Wang, S., Saharia, C., Montgomery, C., Pont-Tuset, J., Noy, S., Pellegrini, S., Onoe, Y., Laszlo, S., Fleet, D.J., Soricut, R., et al.: Imagen editor and editbench: Advancing and evaluating text-guided image inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18359–18369 (2023)
52. Wang, T., Hu, X., Wang, Q., Heng, P.A., Fu, C.W.: Instance shadow detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1880–1889 (2020)
53. Wu, C., Liang, J., Hu, X., Gan, Z., Wang, J., Wang, L., Liu, Z., Fang, Y., Duan, N.: Nuwa-infinity: Autoregressive over autoregressive generation for infinite visual synthesis. arXiv preprint arXiv:2207.09814 (2022)
54. Yang, B., Gu, S., Zhang, B., Zhang, T., Chen, X., Sun, X., Chen, D., Wen, F.: Paint by example: Exemplar-based image editing with diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18381–18391 (2023)
55. Zeng, Y., Fu, J., Chao, H., Guo, B.: Learning pyramid-context encoder network for high-quality image inpainting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1486–1494 (2019)
56. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
57. Zhang, S., Yang, X., Feng, Y., Qin, C., Chen, C.C., Yu, N., Chen, Z., Wang, H., Savarese, S., Ermon, S., et al.: Hive: Harnessing human feedback for instructional visual editing. arXiv preprint arXiv:2303.09618 (2023)
58. Zhang, S., Liang, R., Wang, M.: Shadowgan: Shadow synthesis for virtual objects with conditional adversarial networks. Computational Visual Media **5**, 105–115 (2019)
59. Zhu, Y., Huang, J., Fu, X., Zhao, F., Sun, Q., Zha, Z.J.: Bijective mapping network for shadow removal. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5627–5636 (2022)
60. Zhu, Y., Xiao, Z., Fang, Y., Fu, X., Xiong, Z., Zha, Z.J.: Efficient model-driven network for shadow removal. In: Proceedings of the AAAI conference on artificial intelligence. vol. 36, pp. 3635–3643 (2022)