Supplementary File for ObjectMover: Generative Object Movement with Video Prior

Xin Yu[1,*] Tianyu Wang[2] Soo Ye Kim[2] Paul Guerrero[2] Xi Chen[1] Qing Liu[2]
Zhe Lin[2] Xiaojuan Qi[1,†]
[1]The University of Hong Kong [2]Adobe Research
https://xinyu-andy.github.io/ObjMover

1. Details on Compared Methods

We provide more implementation details of two comparison methods, i.e., DragAnything and 3DiT.

DragAnything [4] DragAnything is a trajectory-based video generation model that allows specifying one or more objects, enabling the corresponding objects to move according to the trajectory in the generated video. However, while this method can control the positions of the generated objects to match the coordinates in the trajectory, it cannot control other elements, such as keeping the background stationary. To maintain the background as much as possible, we need to select a point in the background region and assign it a stationary trajectory to control the relative stability of the background. In our implementation, we strive to keep the background stationary. Specifically, we design two trajectories. The first is the foreground trajectory, which controls the movement of the target object. The start point is the original center position of the object, and the end point is the center of the target position of the object. The key points of the trajectory are obtained through linear interpolation. The second is the background trajectory, where we set the trajectories of a background point in the background region to be stationary, thereby maintaining the stability of the background. To automatically select a background point, we identify the location within the background mask that is farthest from both the foreground object and the image boundaries. This is achieved by computing the Euclidean distance of each background pixel to the nearest foreground pixel and the image borders, then selecting the point with the maximum minimum distance. Nonetheless, despite our effort, this can still result in detail-level jitter or control failures, as shown in the last example of Fig. 10.

3DiT [1] 3DiT is a text-conditioned image editing method that cannot use bounding boxes to precisely control the objects to be moved and their target positions. Instead, it requires a textual prompt to describe the objects and the coordinates of the target positions. To address this, we employ an image caption model to generate text labels for each cropped object in our evaluation set, which are then used as prompt instructions. For the target position coordinates, we use the center points of the target bounding boxes.

2. Video Dataset Pipeline

We use an internal video dataset as the real-world video source. Note that the dataset pipeline can be applied to any video dataset. For processing, we utilize SAM2 [2] to segment the videos and obtain consistent object labels across frames. Then, we filter out objects with masks that are too small and those that do not appear simultaneously in both frames. Finally, we obtain approximately 800,000 image groups, each containing two frames and the corresponding mask image for one object.

Fig. 1 shows some sample data from the video dataset. As mentioned, in the video dataset, other objects or backgrounds, except for the main subject, mostly change, making it difficult to directly use them for training movement tasks. However, we use the video data to train on a mask-based object insertion task. As demonstrated in Fig. 2, during the training process, the object from frame #1 is extracted using mask #1 and serves as the object image, coupled with a foreground-masked frame #2 as the input, to predict a complete frame #2.

Note that our model is trained on video and CG data, while it is evaluated on image data. Hence the training and evaluation data are completely different with no overlap or data leakage issues. *ObjMove-A* is manually captured using DSLR while *ObjMove-B* is web data **without ground-truth**, which theoretically prevents data leakage.

[*] Work done during an internship at Adobe.
[†] Corresponding authors.

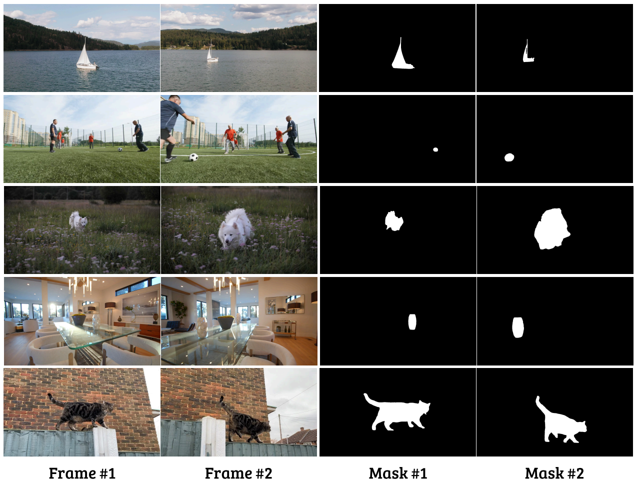

Figure 1. **Video dataset samples.** Frames and corresponding masks from our video dataset, processed with the SAM2 [2] to ensure consistent object labeling across frames.

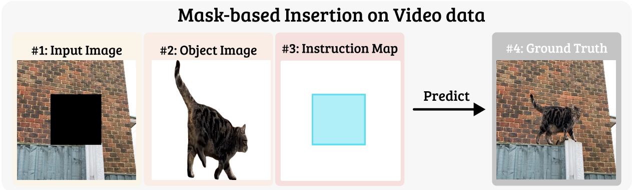

Figure 2. **Mask-based insertion on video data.** #1: Input image with the object masked out. #2: Isolated object image. #3: Instruction map indicating where to place the object. #4: Ground-truth image for prediction.

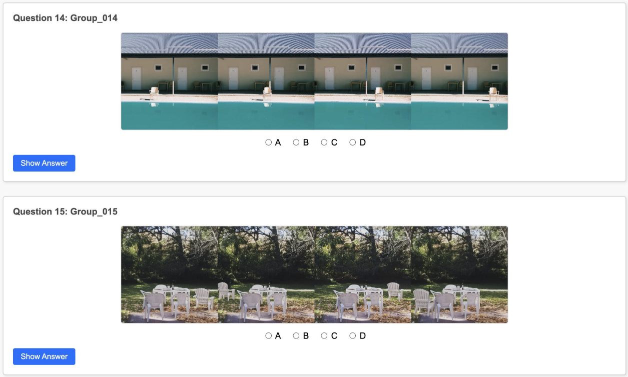

Figure 3. **Interface of the additional user study.** Samples from our "find-the-real-image" game designed to assess the realism of synthesized images. Each participant is shown a set of images and asked to identify the original.

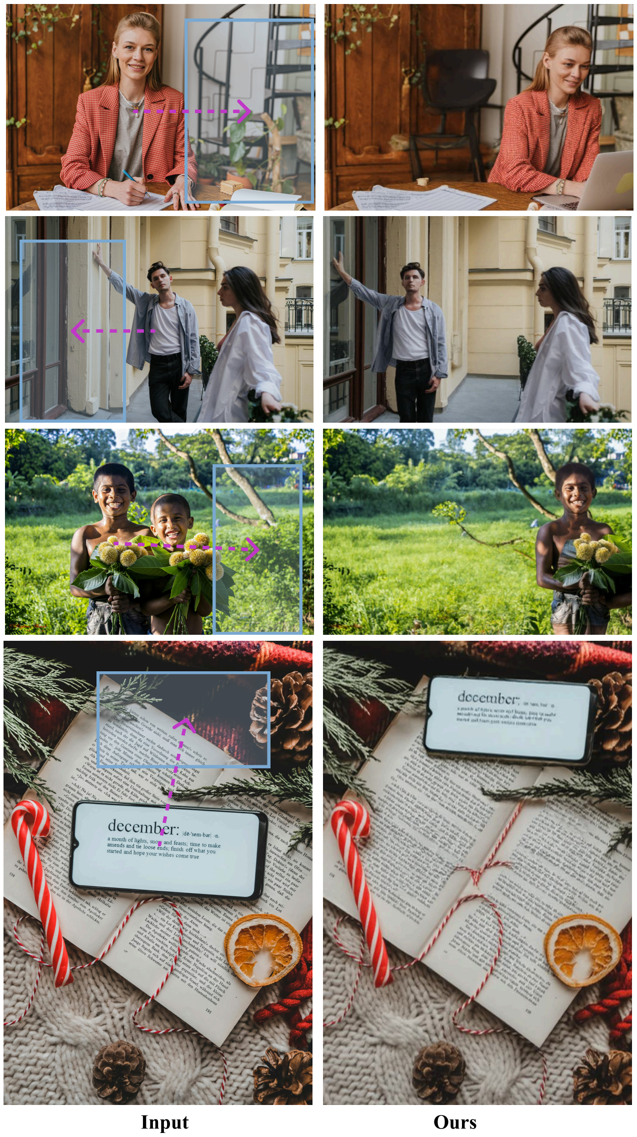

Figure 4. **Illustration of representative failure cases.** Rows 1 and 2 show unintended pose alterations when moving non-rigid objects (e.g., humans), where the generated pose significantly deviates from the original. Row 3 illustrates the disappearance of nearby objects when one object is moved closely past another. Row 4 shows text distortion after object movement, a common limitation inherent in latent diffusion models.

3. Samples of Synthetic Data

Fig. 5 shows some rendered images of different objects in background scenes with varying camera views. We also display their corresponding clean background images where no object is located in the region of interest. These clean background images support our mask-free object removal and insertion training. Fig. 6 presents examples of the full sequence rendering. The first four rows illustrate sequences of the same scene and object in different positions, albeit under different lighting conditions. The last two rows display the corresponding object masks, which are directly obtained through rendering. Notably, these masks represent

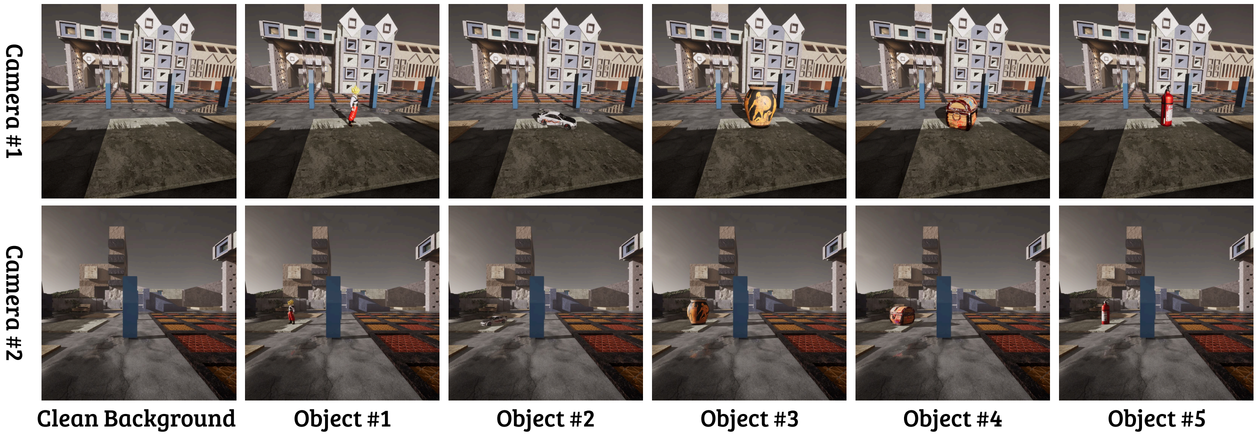

Figure 5. **Image samples of synthetic data.** Display of synthetic scenes with object placements across varying camera angles.

the amodal extent of the objects without considering occlusion relationships. This approach aids the model in learning to determine whether an object should be occluded when a mask overlaps with another object.

4. Additional User Study: Find the Real Image

We conduct an additional user study where users are asked to find the real image among four images, of which three are generated by our method by moving the object of interest to different locations. The results reveal that users incorrectly identify the real (input) image 70% of the time, demonstrating our method's ability to generate realistic images that effectively obscure artifacts. Samples of this game are illustrated in Fig. 3, and we also provide a web link here to play the game interactively.

5. More Results

We provide additional qualitative results of our model on in-the-wild internet images. Fig. 7, Fig. 8, and Fig. 9 respectively showcase more results of our model on object movement, removal, and insertion. For each result, we annotate key aspects above the images to better demonstrate the capabilities of our model.

6. More Comparisons

We present additional comparison results between our method and other approaches. Fig. 10 and Fig. 11 respectively show the comparison results for the movement and removal tasks on *ObjMove-B*. Fig. 12 displays the insertion results on in-the-wild image pairs. Additionally, Fig. 13, Fig. 14, and Fig. 15 illustrate the movement, removal, and insertion results on *ObjMove-A*, where a reference ground-truth image is also provided.

7. Limitations and Future Work

While our method achieves excellent results across three tasks—object movement, removal, and insertion—it still possesses certain limitations. Figure 4 illustrates several failure cases, categorized into three main scenarios:

1. **Unintended Pose Alterations.** Our design philosophy emphasizes maintaining the object's original pose as consistently as possible, only automatically adjusting the pose when necessary for harmonious integration into the new environment. This strategy generally ensures stable and robust performance. However, for non-rigid objects (e.g., humans), generated results sometimes exhibit significant and unintended pose alterations, occasionally introducing new content (rows 1 and 2 in Figure 4). We suspect this primarily arises from the abundance of human-motion examples in real video datasets, which bias the model towards pose variability. To address this, we plan to incorporate meta-information regarding relative object-camera poses into our synthetic dataset and conditionally train the model based on this information. This enhancement will enable explicit 3D control, allowing precise, user-directed pose manipulation.
2. **Disappearance of Nearby Objects.** When an object is moved closely past another object (row 3 in Figure 4), the nearby object occasionally disappears. We attribute this to a lack of examples where one object explicitly crosses over another within our synthetic training data. This limitation can easily be resolved by augmenting the dataset with relevant scenarios.
3. **Text Distortion.** For objects containing text (row 4 in Figure 4), moving the object often results in distorted text. This is a common limitation in latent diffusion models caused by insufficient reconstruction capabilities of the VAE.

Moreover, our method exhibits relatively slow inference

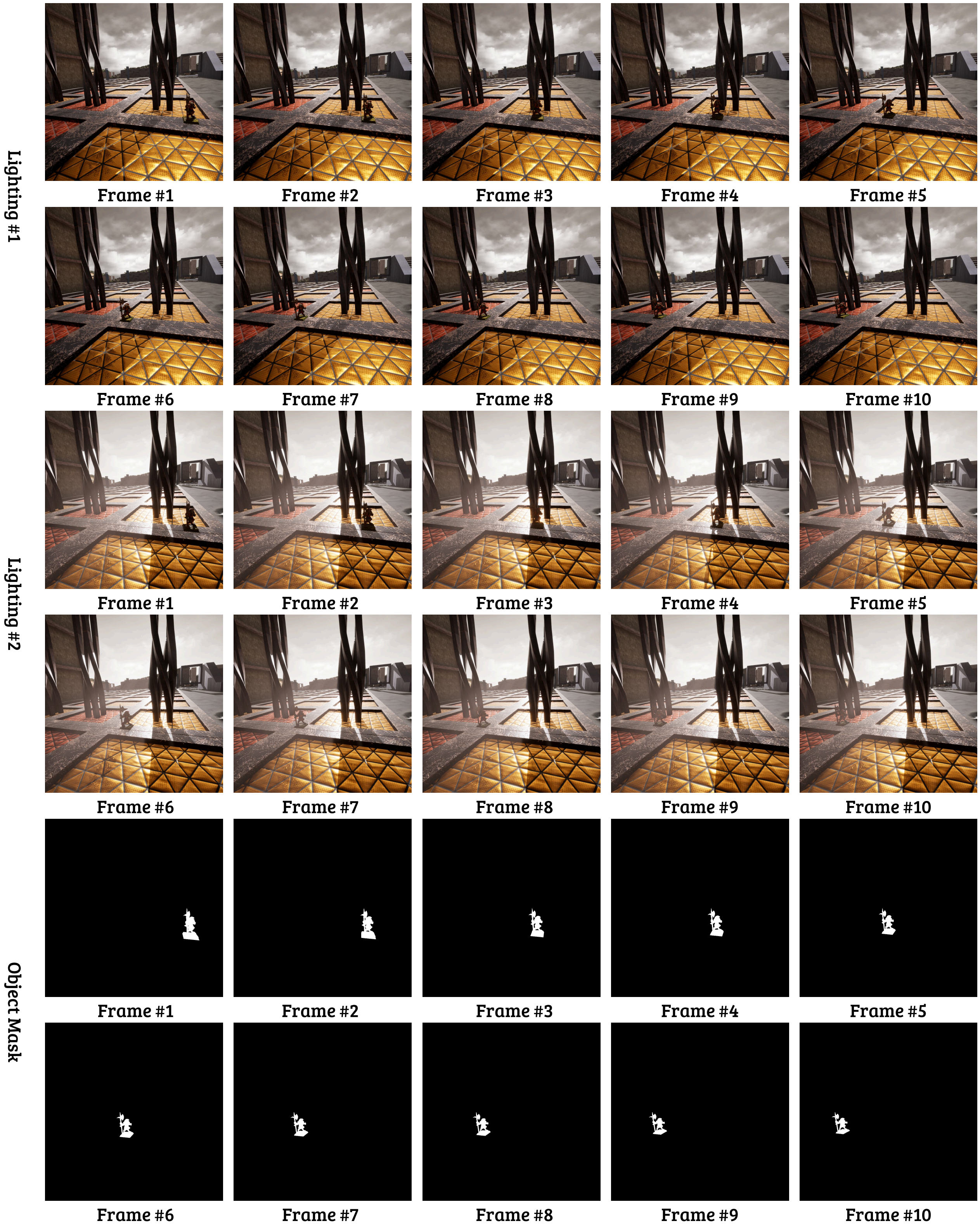

Figure 6. **Full sequence examples of synthetic data.** We show two sequences from our synthetic dataset with an object placed in different locations with two lighting conditions. The last two rows present the object masks.

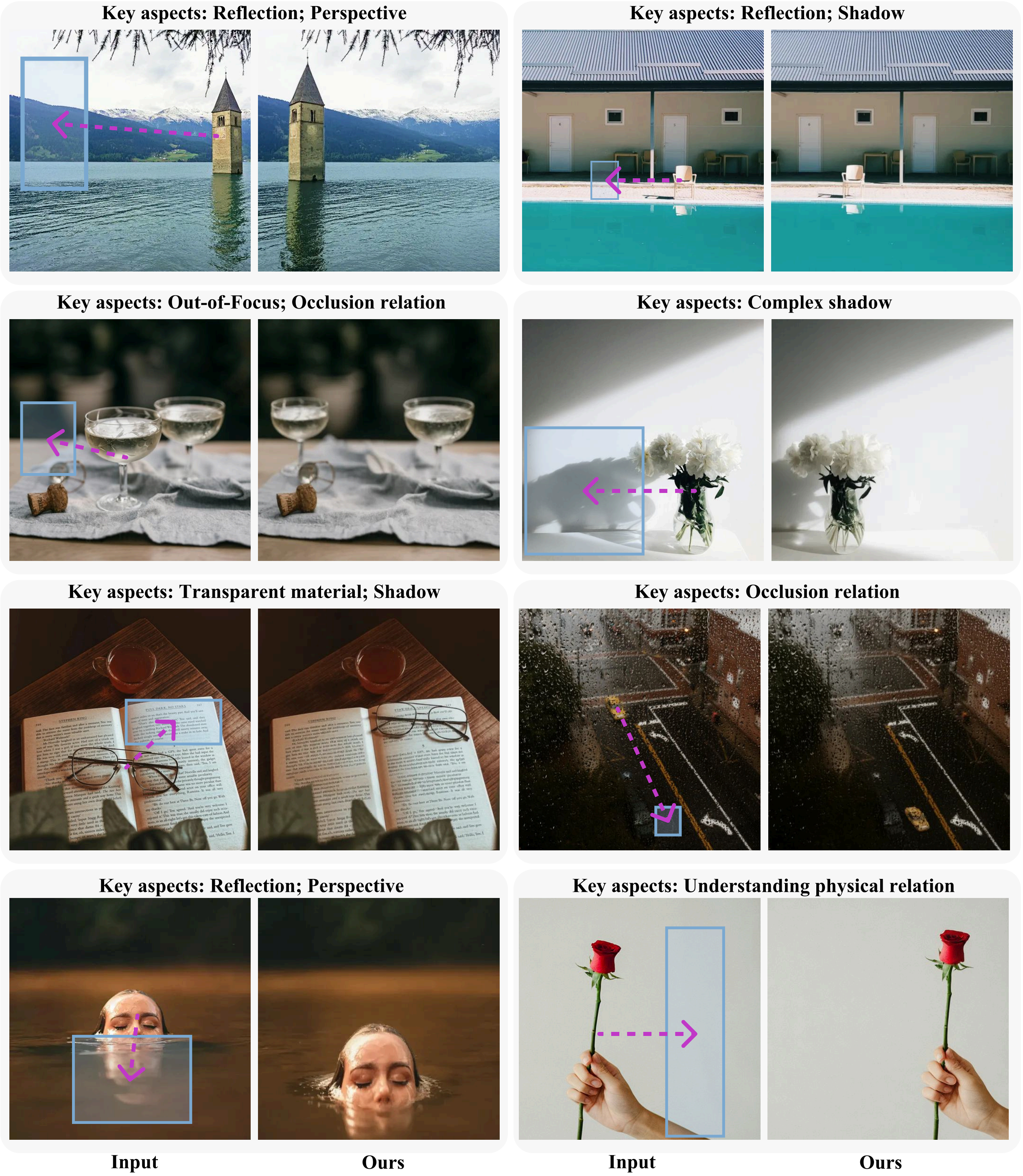
Key aspects: Reflection; Perspective
Key aspects: Reflection; Shadow
Key aspects: Out-of-Focus; Occlusion relation
Key aspects: Complex shadow
Key aspects: Transparent material; Shadow
Key aspects: Occlusion relation
Key aspects: Reflection; Perspective
Key aspects: Understanding physical relation
Input
Ours
Input
Ours

Figure 7. **Qualitative results on object movement.** Key aspects to focus on are annotated above each image to highlight the model's ability.

speed. On a single NVIDIA A100 GPU, inferring an image with a resolution of 512×512 requires approximately 20 seconds, which is slower than other U-Net-based approaches. However, in future work, we aim to reduce the inference cost by employing model distillation and diffusion distillation techniques [3, 5, 6], thereby enhancing the practical applicability of our approach in real-world scenarios.

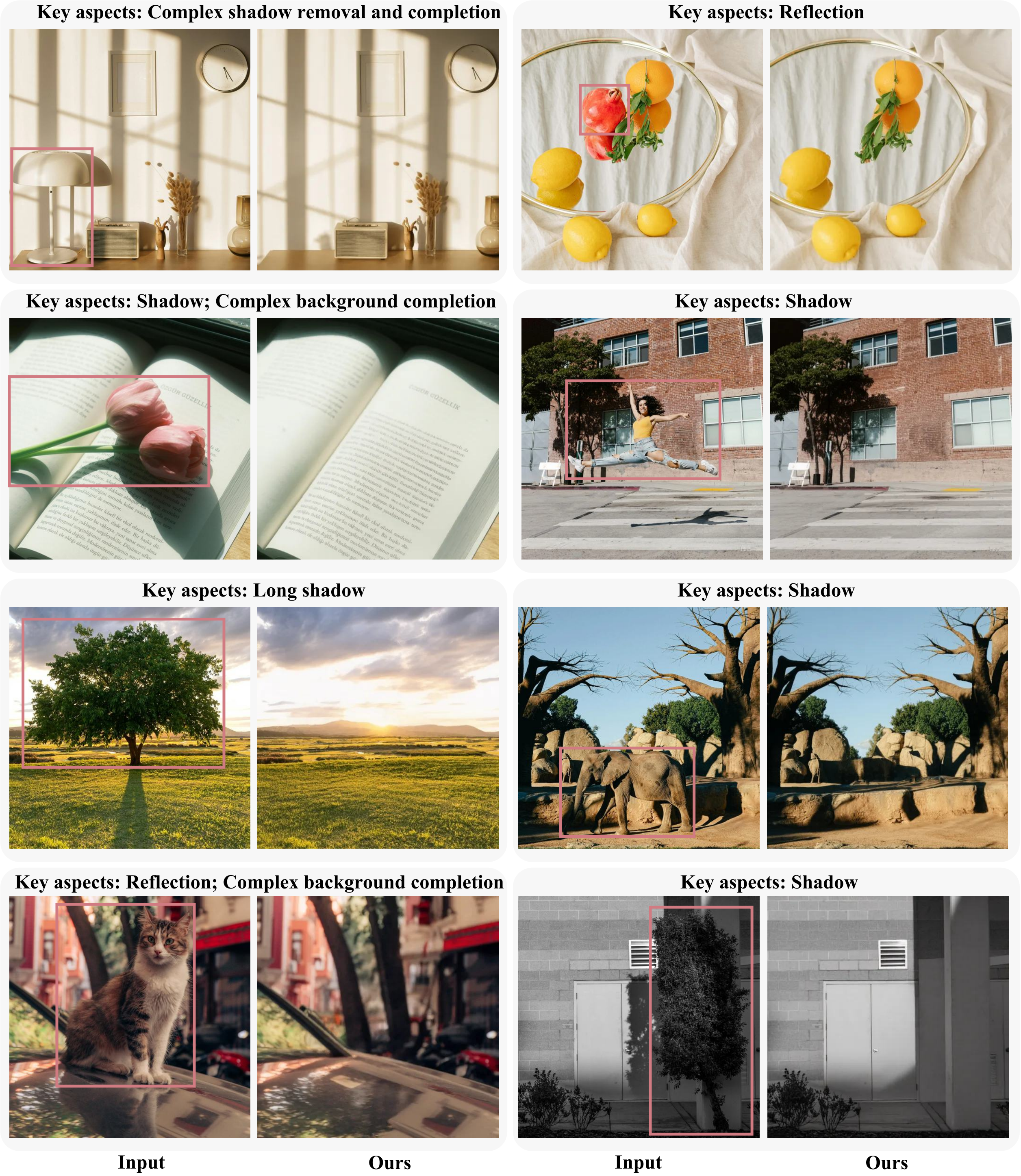
Key aspects: Complex shadow removal and completion
Key aspects: Reflection
Key aspects: Shadow; Complex background completion
Key aspects: Shadow
Key aspects: Long shadow
Key aspects: Shadow
Key aspects: Reflection; Complex background completion
Key aspects: Shadow
Input
Ours
Input
Ours

Figure 8. **Qualitative results on object removal.** Key aspects to focus on are annotated above each image to highlight the model's ability.

References

[1] Oscar Michel, Anand Bhattad, Eli VanderBilt, Ranjay Krishna, Aniruddha Kembhavi, and Tanmay Gupta. Object 3dit: Language-guided 3d-aware image editing. *Advances in Neural Information Processing Systems*, 36, 2024. 1

[2] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu,

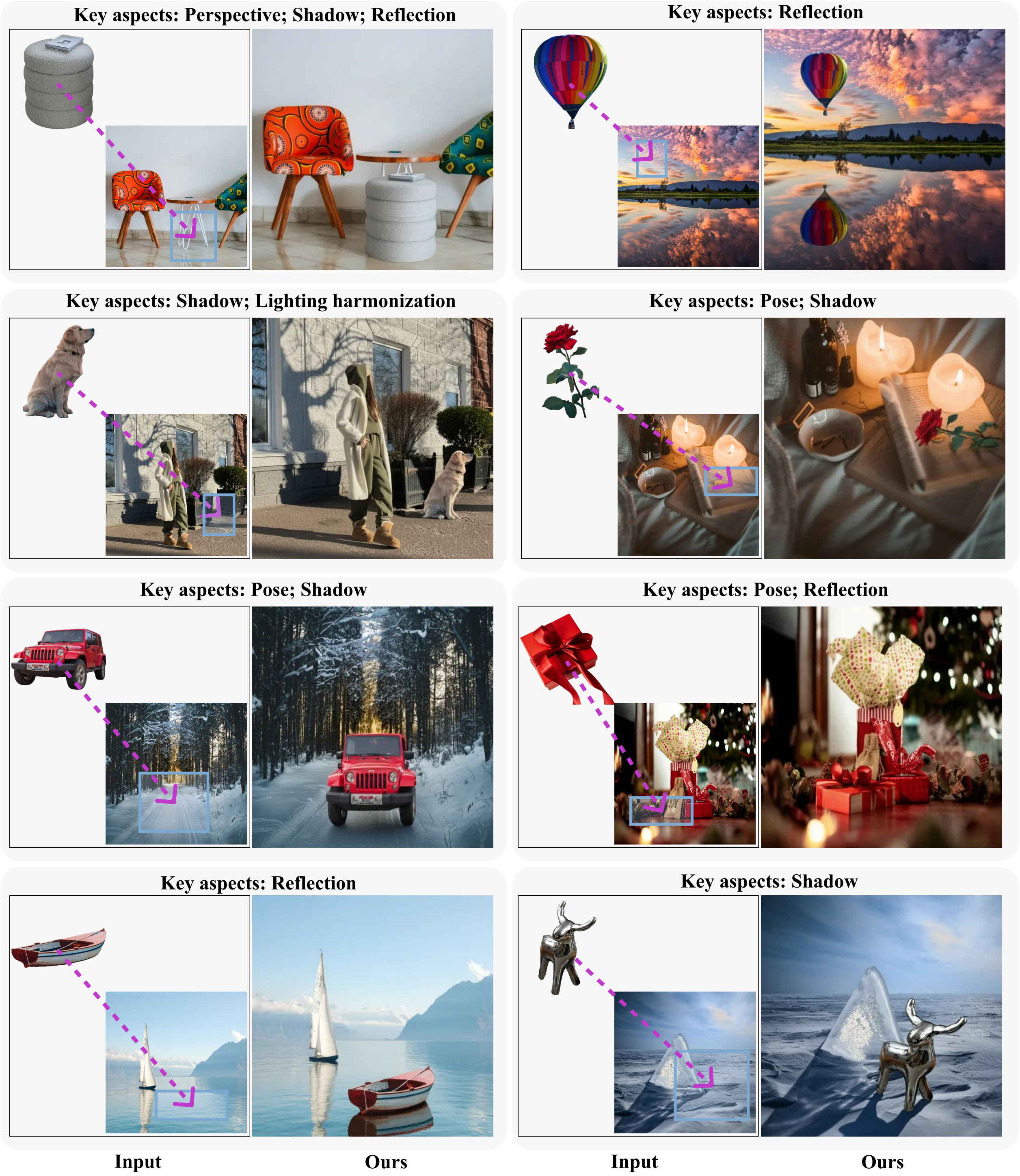
Key aspects: Perspective; Shadow; Reflection
Key aspects: Reflection
Key aspects: Shadow; Lighting harmonization
Key aspects: Pose; Shadow
Key aspects: Pose; Shadow
Key aspects: Pose; Reflection
Key aspects: Reflection
Key aspects: Shadow
Input
Ours
Input
Ours

Figure 9. **Qualitative results on object insertion.** Key aspects to focus on are annotated above each image to highlight the model's ability.

Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 1, 2

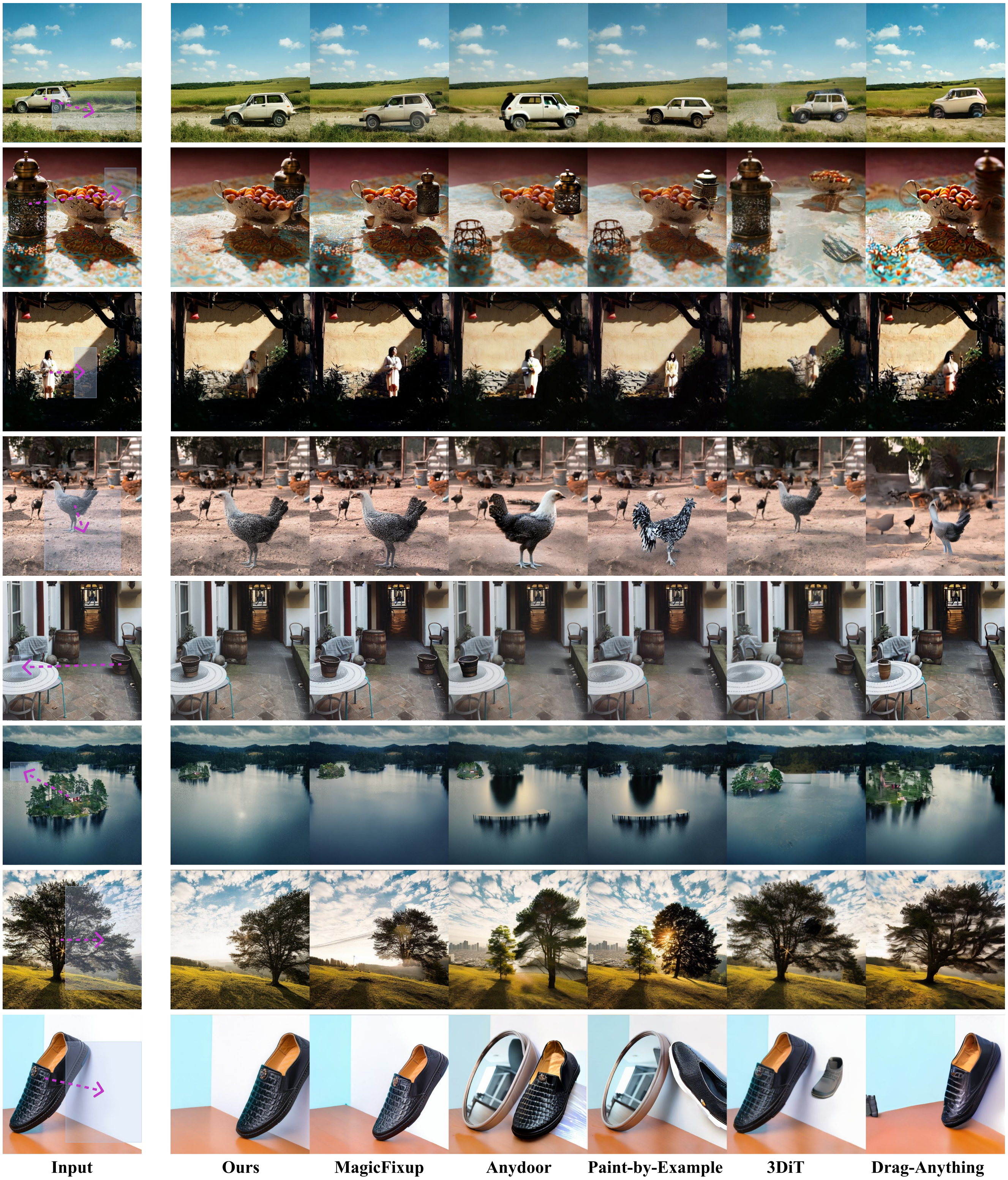

Figure 10. **Qualitative comparisons on object movement.** Our method consistently outperforms state-of-the-art methods.

[3] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023. 5

[4] Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He,

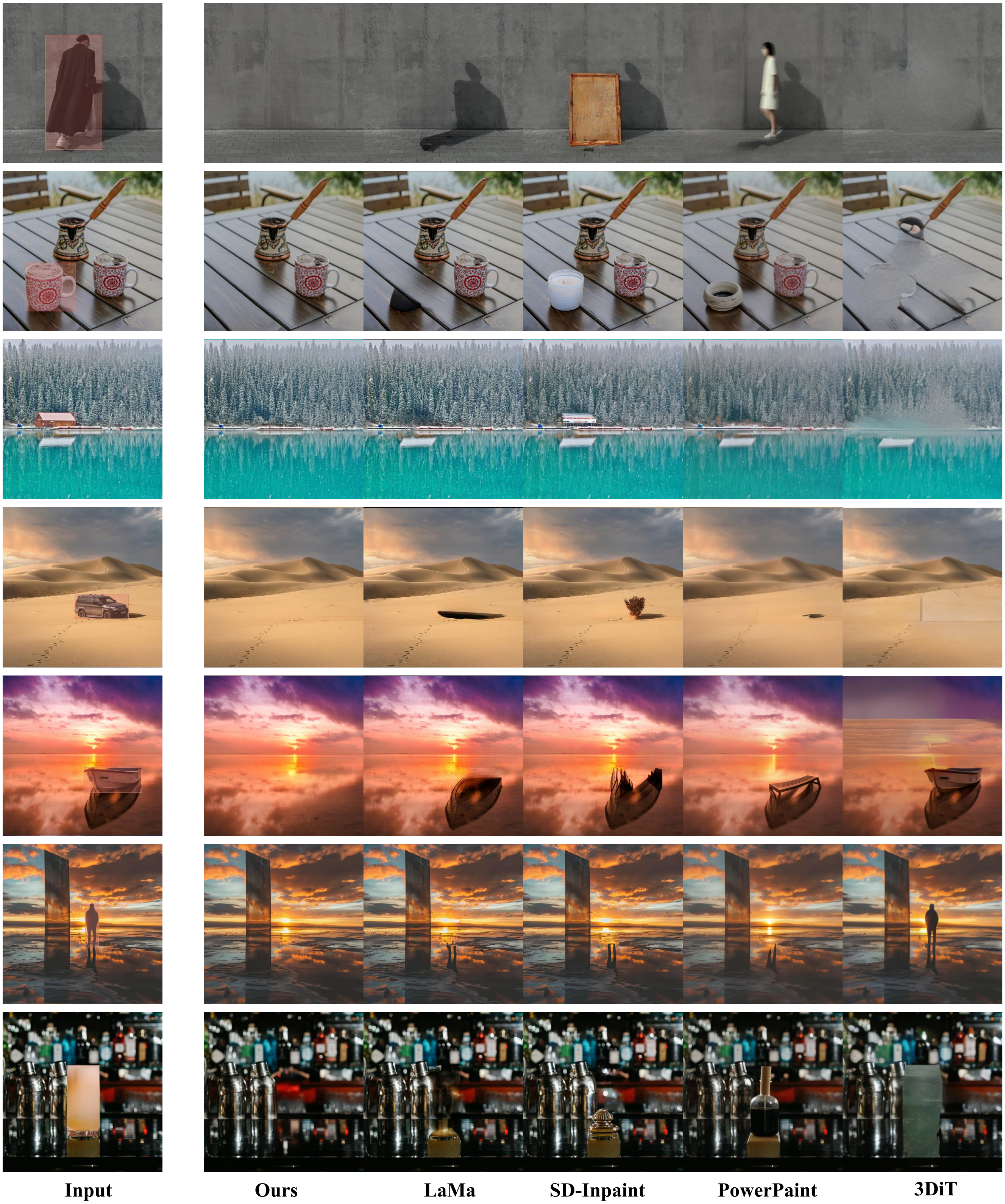

Figure 11. **Qualitative comparisons on object removal.** Our method consistently outperforms state-of-the-art methods.

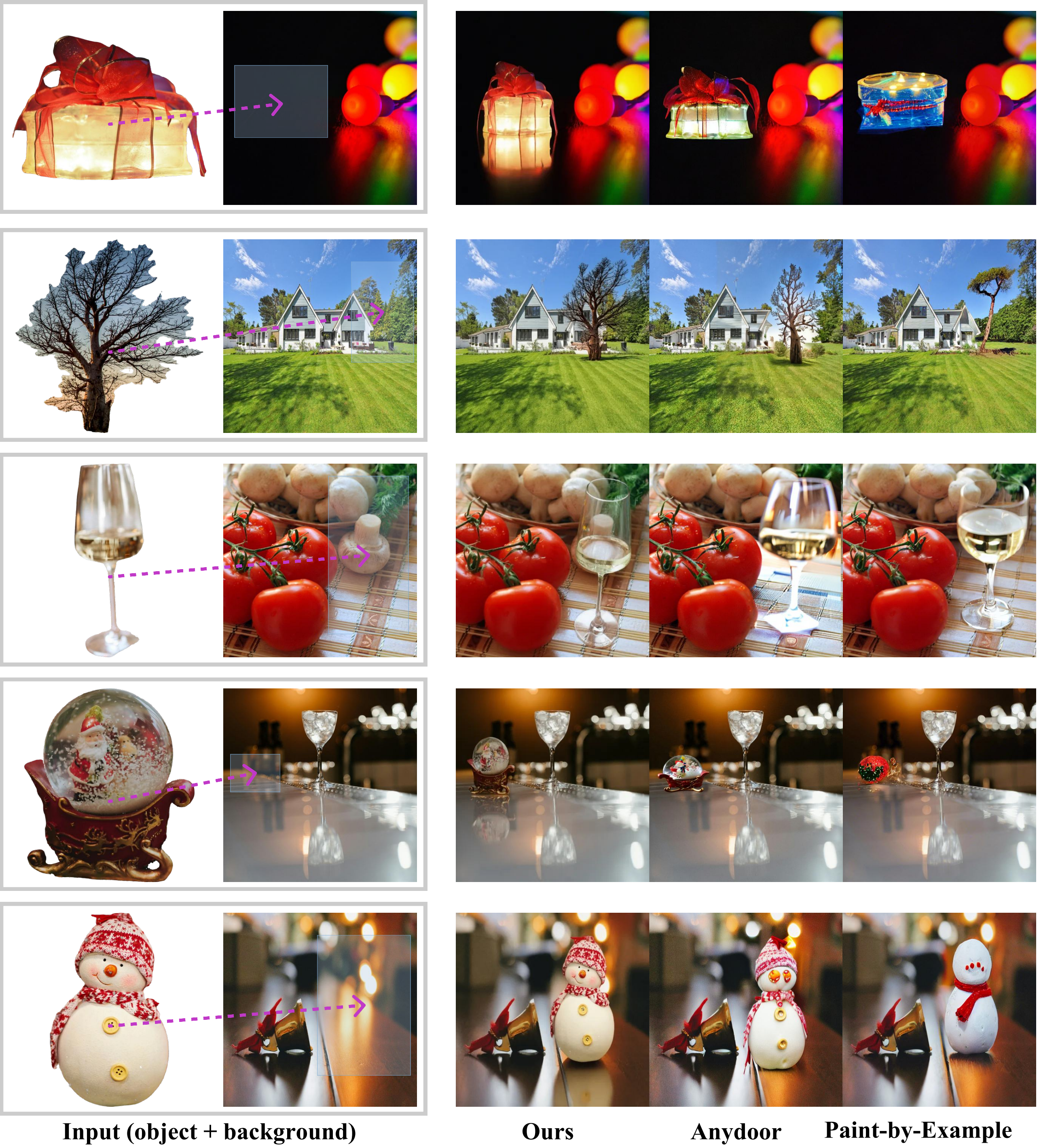

Figure 12. **Qualitative comparisons on object insertion.** Our method consistently outperforms state-of-the-art methods.

David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. Draganything: Motion control for anything using entity representation. In *European Conference on Computer Vision*, pages 331–348. Springer, 2025. 1

[5] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *arXiv preprint arXiv:2405.14867*, 2024. 5

[6] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision*

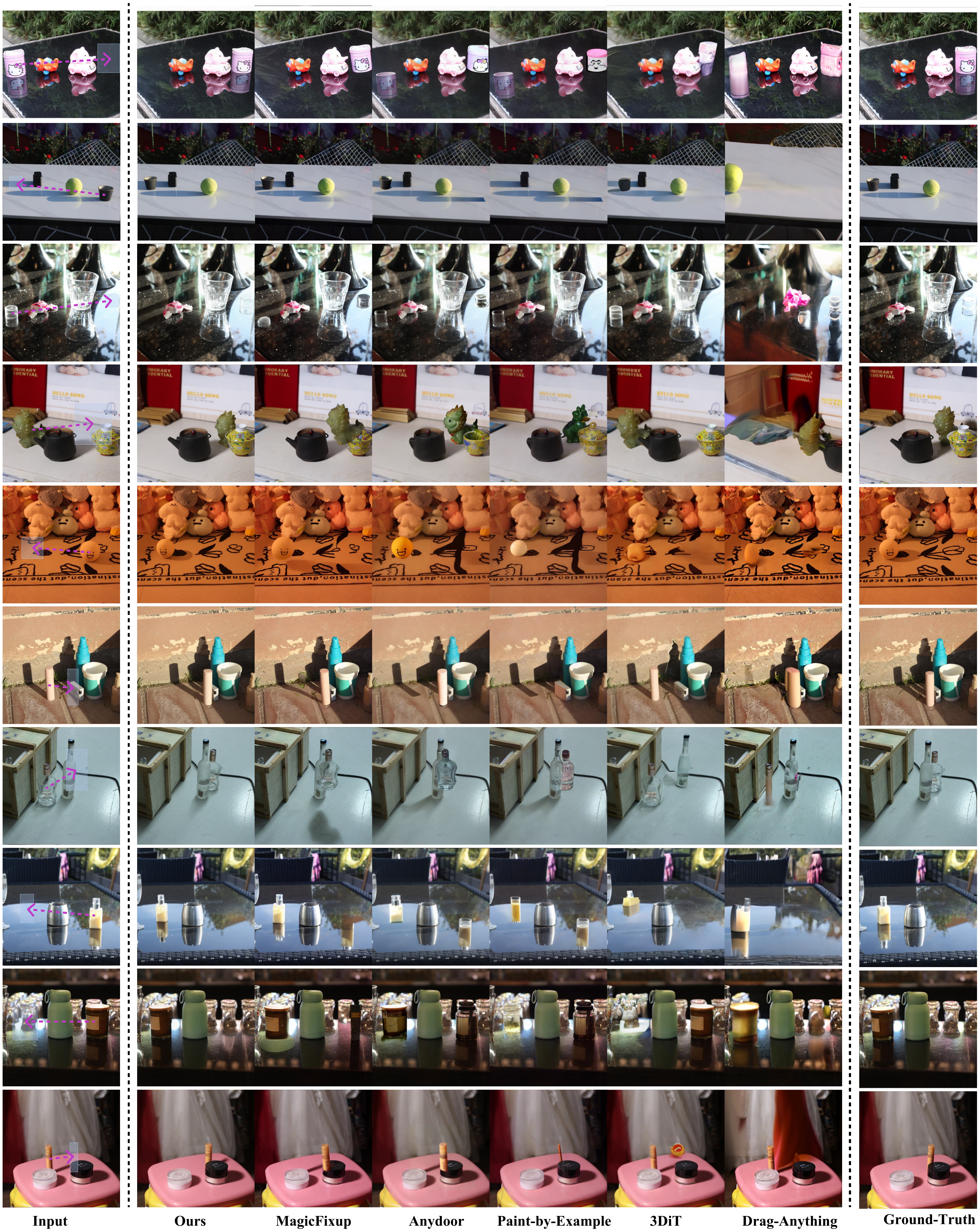

Figure 13. **Qualitative comparisons on object movement.** Our method consistently outperforms state-of-the-art methods.

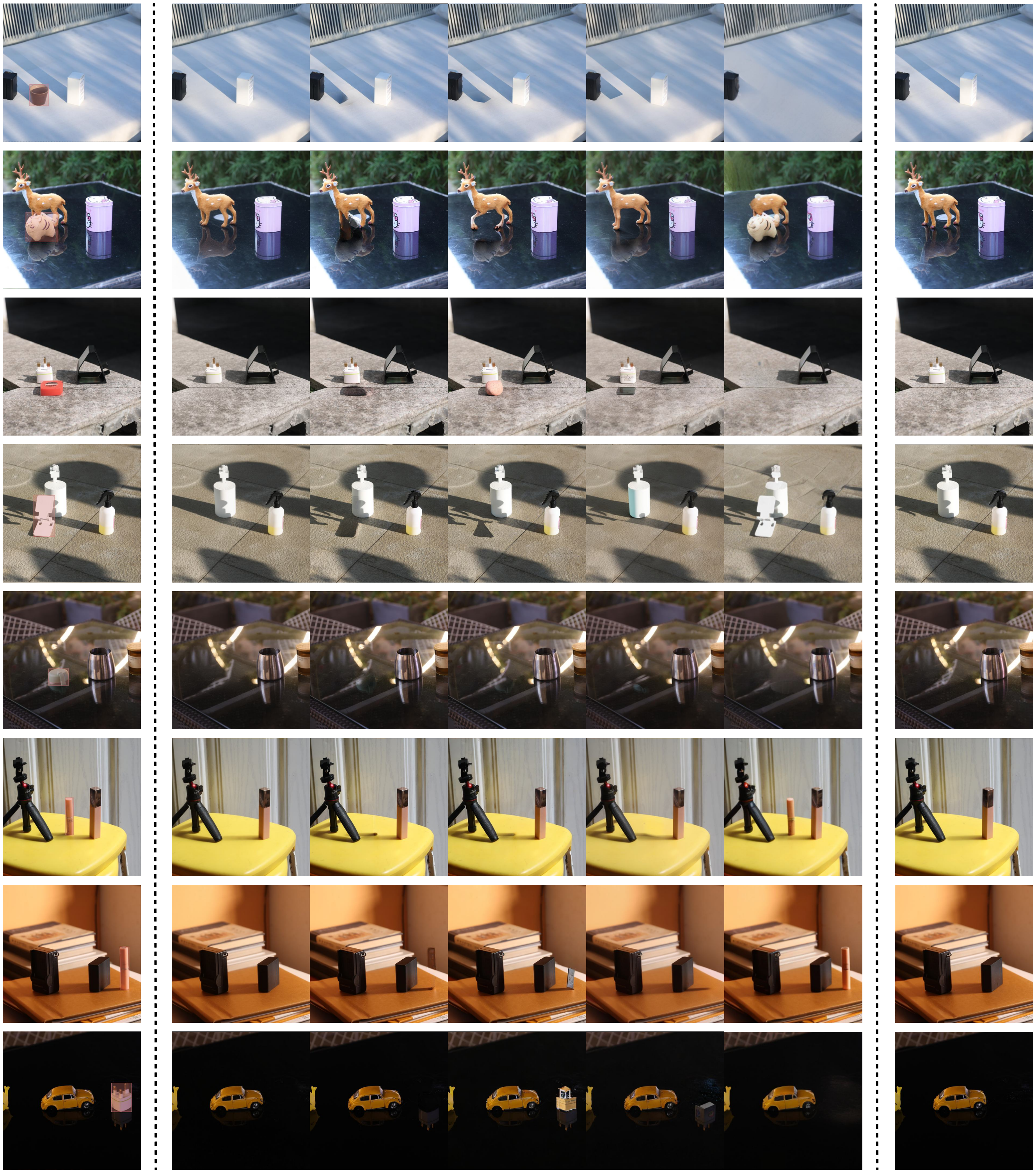

Figure 14. **Qualitative comparisons on object removal.** Our method consistently outperforms state-of-the-art methods.

and Pattern Recognition, pages 6613–6623, 2024. 5

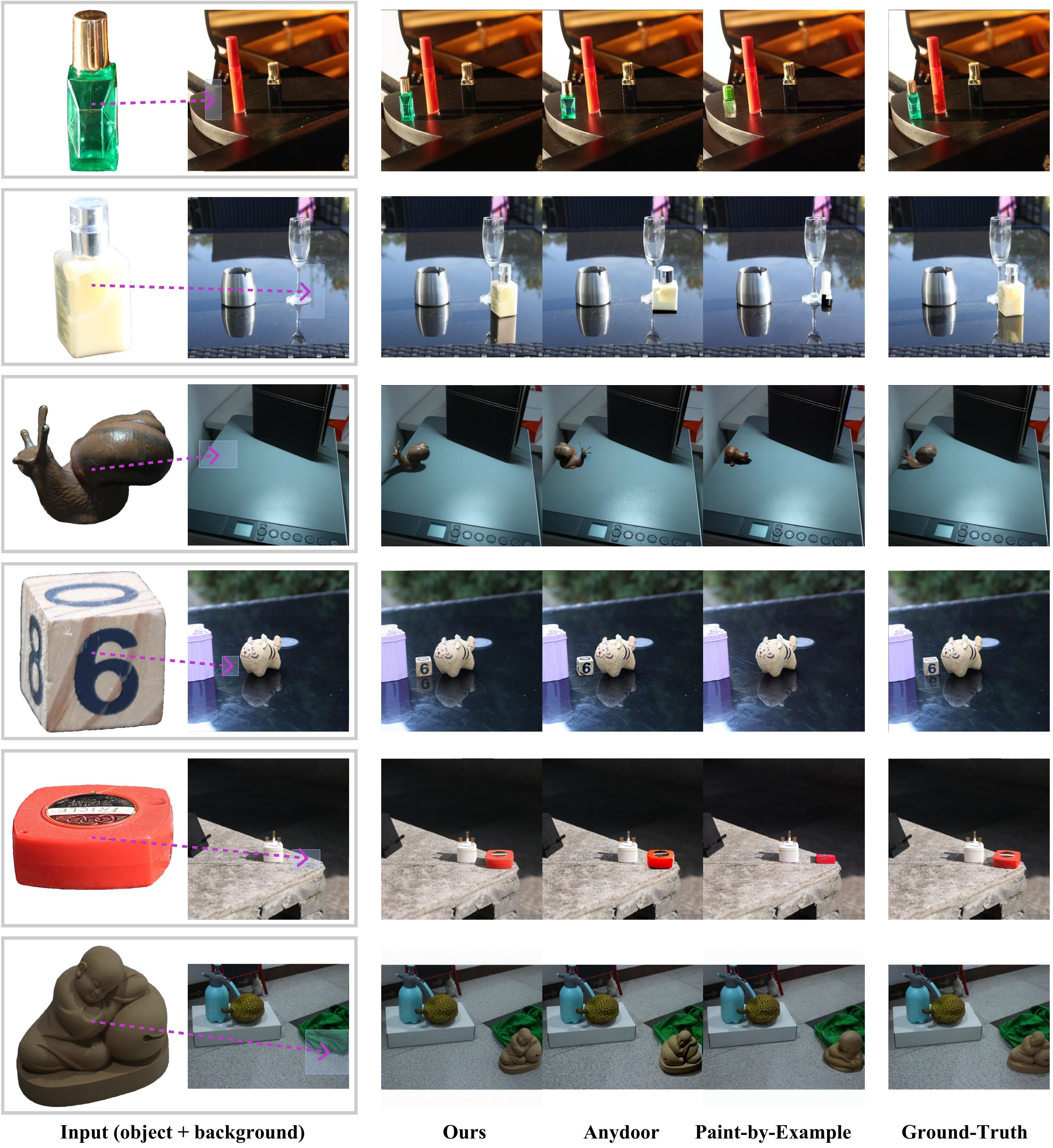

Figure 15. **Qualitative comparisons on object insertion.** Our method consistently outperforms state-of-the-art methods.