Supplementary – "OmniPaint: Mastering Object-Oriented Editing via Disentangled Insertion-Removal Inpainting"

Yongsheng Yu[1 *], Ziyun Zeng[1 *], Haitian Zheng[2], Jiebo Luo[1]
[1]University of Rochester, [2]Adobe Research
{yyu90,zzeng24}@ur.rochester.edu, hazheng@adobe.com, jluo@cs.rochester.edu

A. Implementation Details

A.1. Training Details

OmniPaint is built upon FLUX.1-dev [6] and trained across multiple stages with batch size 8 at a resolution of 1024^2. Training images, including target images, masked images, and reference object images, are first resized so that the shortest edge is 1024. This is followed by random cropping while ensuring that the target object, as defined by the input mask, remains intact.

With the FLUX backbone frozen, all training stages optimize both LoRA parameters θ and ϕ using the Prodigy optimizer [7], with safeguard warmup, bias correction enabled, and a weight decay of 0.01.

Pretext Training. We use 200K images from LAION [8] and apply the random mask generator from LaMa [11]. Both θ and ϕ are trained for 25K iterations.

Warmup Training. We train on 3,000 real-world paired samples collected across diverse indoor and outdoor environments, capturing various physical effects such as shadows, reflections, and occlusions. A subset of 300 samples is reserved for testing. Both removal and insertion parameters are trained for 25K iterations.

CycleFlow Post-Training. For unpaired training in object insertion, we use a curated set of 105,561 object segmentation samples from COCO-Stuff [1] and HQSeg [5], prioritizing objects with physical effects. Additionally, we filter out objects occupying more than 35% or less than 5% of the image area to exclude impractical cases. Training runs for 60K iterations with a cycle loss weight of $\gamma = 1.5$.

LoRA Configuration. LoRA parameters θ and ϕ use a default rank of 4, with LoRA scaling set to 0 for non-condition tokens to preserve the model's inherent capabilities while allowing flexibility [12].

A.2. Mask Augmentation

Figure A illustrates the randomized mask augmentations applied during object removal training, categorized into four distinct types. To better simulate slightly imprecise mask inputs and enhance adaptability to real-world scenarios, we employ the following probabilities for augmentation selection: *original* (20%), *dilate* (10%), *erode* (20%), *add shapes* (10%), and *remove shapes* (40%).

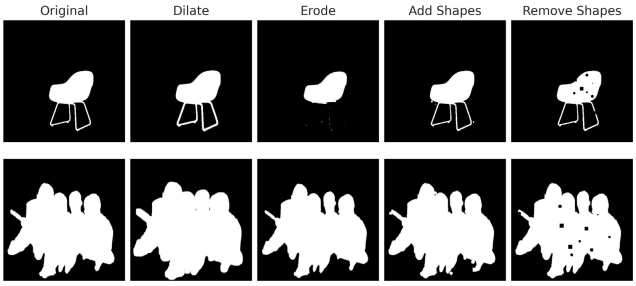

Figure A. Mask augmentation examples.

B. Ablation Study

Training Strategies. We conduct an ablation study to evaluate the impact of different training strategies on object removal and insertion. As shown in Table A, pretext training establishes basic inpainting capabilities and benefits from subsequent warmup. Leveraging our paired training data, warmup further improves removal and insertion quality, reducing CFD and DreamSim while enhancing PSNR and CLIP-I. For object insertion, incorporating CycleFlow yields the best performance, demonstrating that OmniPaint effectively learns under unpaired data with the proposed cycle loss. The last row of Table A omits removal metrics, as CycleFlow is designed solely to enhance insertion. These results highlight the importance of our progressive training pipeline. Warmup establishes foundational object removal and insertion capabilities, while CycleFlow leverages large-scale unpaired data to enhance object identity preservation. This is further illustrated in Fig. 7(b) of the main text, where the model trained with warmup alone successfully inserts objects with physical effects but exhibits lower object identity consistency. With CycleFlow ($\gamma = 1.5$), OmniPaint not only generates realistic physical effects but also maintains high-fidelity object identity.

Table A. Ablation study on training strategies. Each variant is trained for 25K iterations with the same setup, except for the specified strategies. The first row omits insertion metrics as pretext training enables only inpainting. The last row excludes removal metrics, as CycleFlow is used solely for insertion in our design.

Training Strategy			*Object Removal*		*Object Insertion*	
Pretext	**Warmup**	**CycleFlow**	**CFD ↓**	**PSNR ↑**	**DreamSim ↓**	**CLIP-I ↑**
✓			0.4042	19.4864	-	-
	✓		0.2794	23.1935	0.2332	89.0768
✓	✓		0.2789	23.4757	0.1758	89.2344
✓	✓	✓	-	-	0.1557	92.2693

Mask Augmentation. To assess the impact of mask augmentation on object removal, we compare models trained with and without this technique. As shown in Fig. B, models trained without augmentation struggle in challenging scenarios, particularly with complex reflections, incomplete masks, or masks that do not conform to object shapes. Introducing mask augmentation enhances robustness. The model adapts to a wider range of mask shapes, enabling more consistent object removal while effectively eliminating specular reflections.

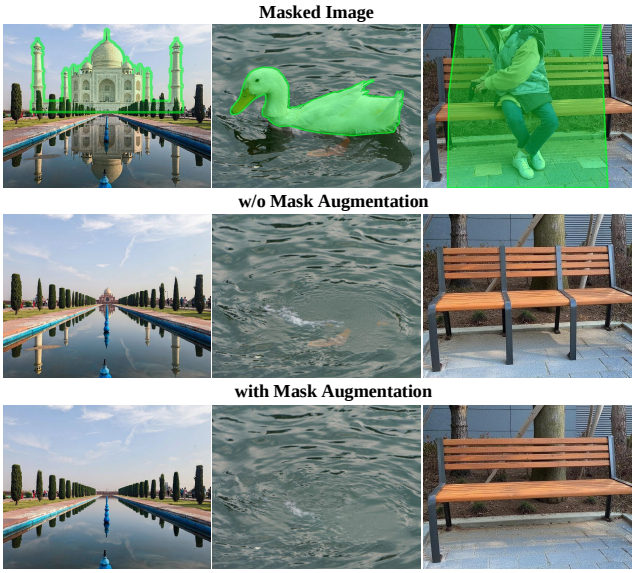

Figure B. Impact of mask augmentation on object removal training. The top row shows masked inputs across three challenging scenarios: complex reflections, imprecise masks, and non-object-shaped masks. The middle row depicts training with only high-precision, human-crafted object masks.

C. Limitations

Despite its strong performance, our approach has limitations. As shown in Fig. C, our insertion results struggle to accurately reproduce fine text due to inevitable information loss from VAE encoding of the reference object, highlighting the need for fine-grained feature preservation. Addressing this challenge is essential for future improvements to enhance applicability in more challenging scenarios.

Figure C. Limitations of OmniPaint. The reference object (left) is inserted into the scene (middle), with zoomed-in comparisons on the right. The top-right close-up shows the reference, while the bottom-right shows the insertion result, revealing text discrepancy.

D. Dataset Samples

D.1. Unpaired Data

High quality image segmentation datasets, such as COCO-Stuff [1] and HQSeg [5], serve as sources for unpaired samples. Each unpaired sample contains an original image and a corresponding mask of the foreground object. Since the object's shadow and reflection remain in the unmask region, directly training object insertion on this data leads to copy-paste look, making it difficult for the model to learn how to generate realistic physical effects. Examples of unpaired samples are shown in Fig.D

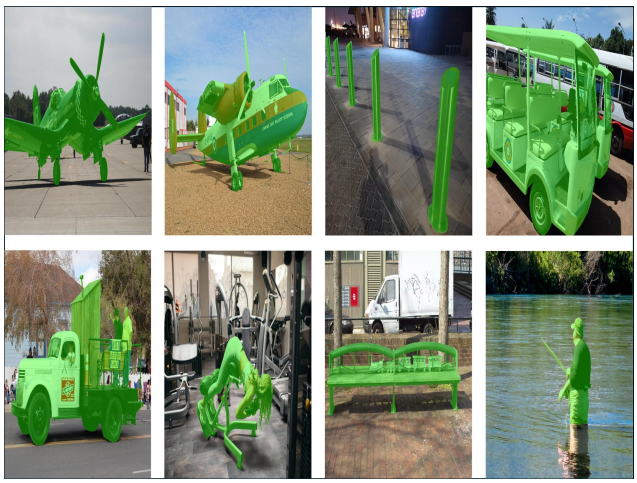

Figure D. Overview of the unpaired samples. In real-world scenarios, the areas covered by the green mask in the image represents the object mask.

D.2. Paired Data

We manually collected 3,300 paired samples for real-world object insertion and removal training and validation. A stable tripod ensured a fixed viewpoint. First, we captured

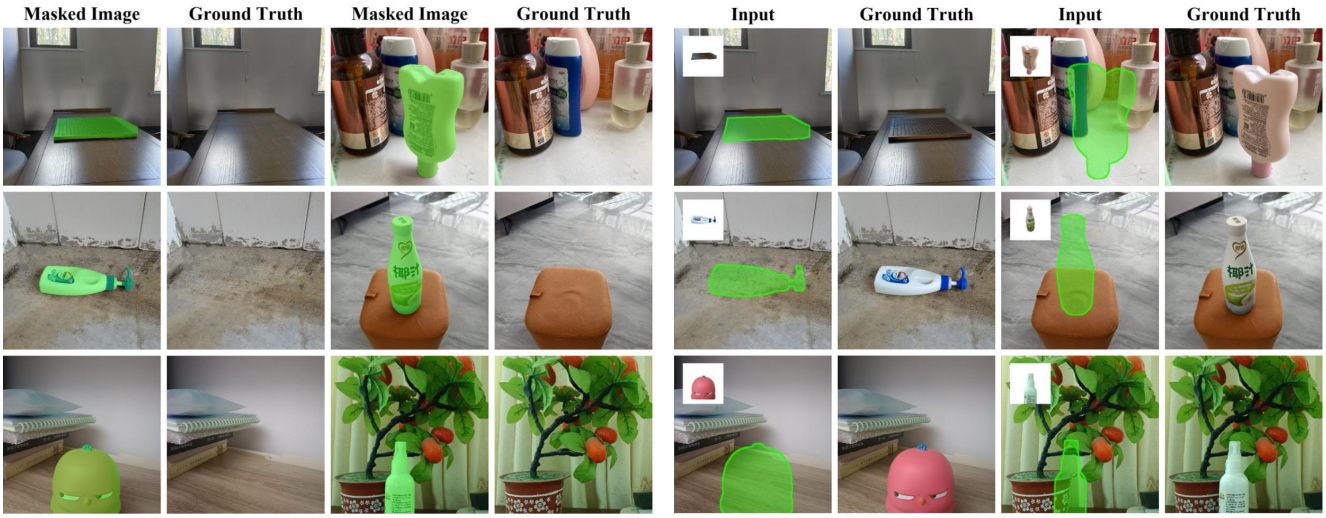

Figure E. Examples of our real-world dataset for object removal (left) and insertion (right) training.

an image with a specific set of objects, then physically removed them and took another image as the ground truth for object removal. After obtaining paired images, we manually annotated the removal object masks for each pair. Figure E displays paired samples for both object removal and insertion training.

E. More Visual Comparisons

Figures F–J present 65 qualitative comparisons for object removal in challenging scenarios, while Fig. K showcases 7 additional cases for object insertion.

References

[1] Holger Caesar, Jasper R. R. Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, 2018. 1, 2

[2] Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. In *CVPR*, 2024. 9

[3] Zhekai Chen, Wen Wang, Zhen Yang, Zeqing Yuan, Hao Chen, and Chunhua Shen. Freecompose: Generic zero-shot image composition with diffusion prior. In *ECCV*, 2024. 4, 5, 6, 7, 8

[4] Yiğit Ekin, Ahmet Burak Yildirim, Erdem Eren Caglar, Aykut Erdem, Erkut Erdem, and Aysegul Dundar. CLIP-Away: Harmonizing focused embeddings for removing objects via diffusion models. In *NeurIPS*, 2024. 4, 5, 6, 7, 8

[5] Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. In *NeurIPS*, 2023. 1, 2

[6] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 1

[7] Konstantin Mishchenko and Aaron Defazio. Prodigy: An expeditiously adaptive parameter-free learner. In *ICML*, 2024. 1

[8] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: an open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022. 1

[9] Yizhi Song, Zhifei Zhang, Zhe Lin, Scott Cohen, Brian L. Price, Jianming Zhang, Soo Ye Kim, and Daniel G. Aliaga. Objectstitch: Object compositing with diffusion model. In *CVPR*, 2023. 9

[10] Yizhi Song, Zhifei Zhang, Zhe Lin, Scott Cohen, Brian L. Price, Jianming Zhang, Soo Ye Kim, He Zhang, Wei Xiong, and Daniel G. Aliaga. IMPRINT: generative object compositing by learning identity-preserving representation. In *CVPR*, 2024. 9

[11] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *WACV*, 2022. 1

[12] Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. Ominicontrol: Minimal and universal control for diffusion transformer. *arXiv preprint arXiv:2411.15098*, 2024. 1

[13] Alimama Creative Team. Flux-controlnet-inpainting. https://github.com/alimama-creative/FLUX-Controlnet-Inpainting, 2024. 4, 5, 6, 7, 8

[14] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *CVPR*, 2023. 9

[15] Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting. In *ECCV*, 2024. 4, 5, 6, 7, 8

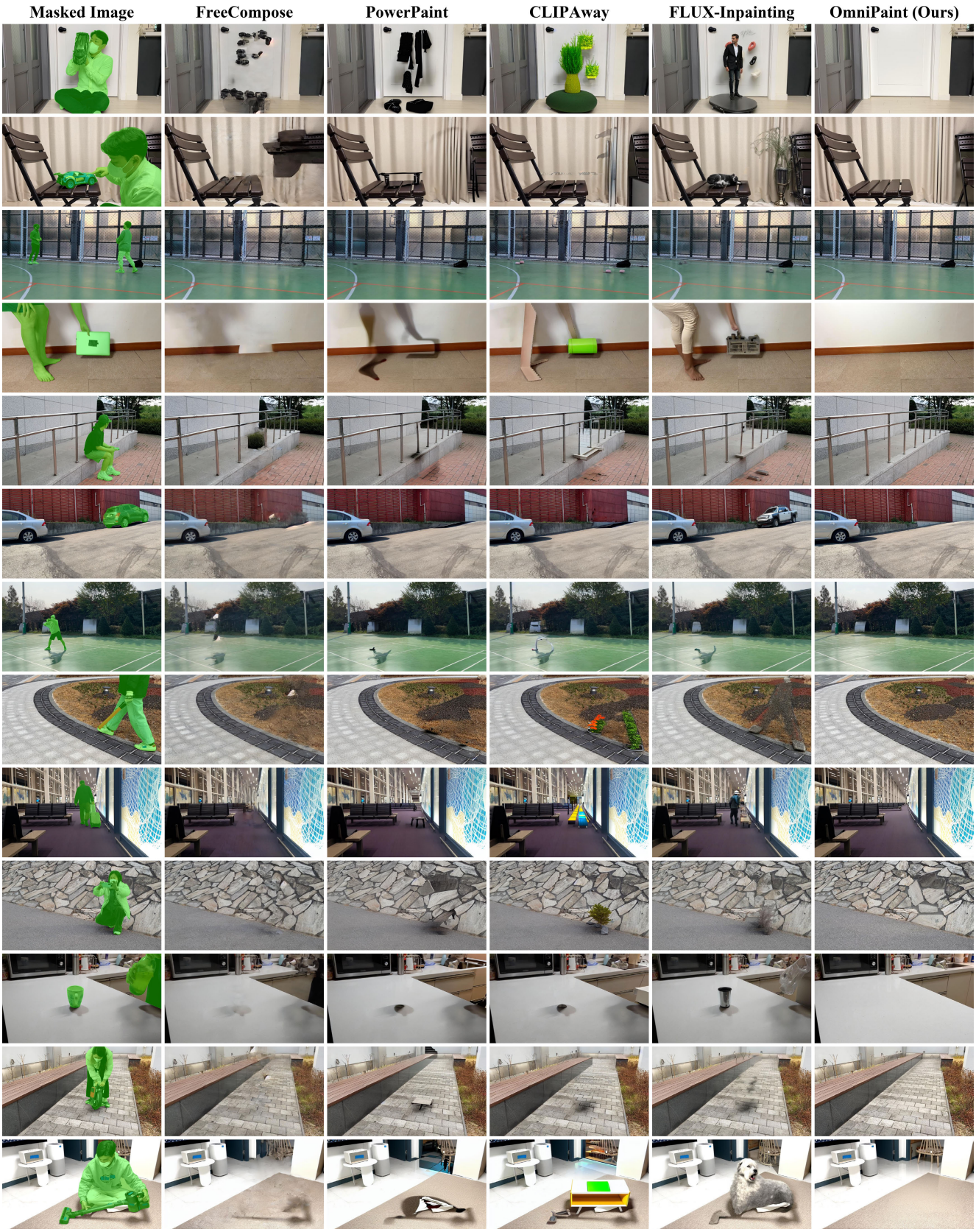

Figure F. More Qualitative comparison of object removal in challenging scenarios (Part 1). The compared methods include FreeCompose [3], PowerPaint [15], CLIPAway [4], and FLUX-Inpainting [13].

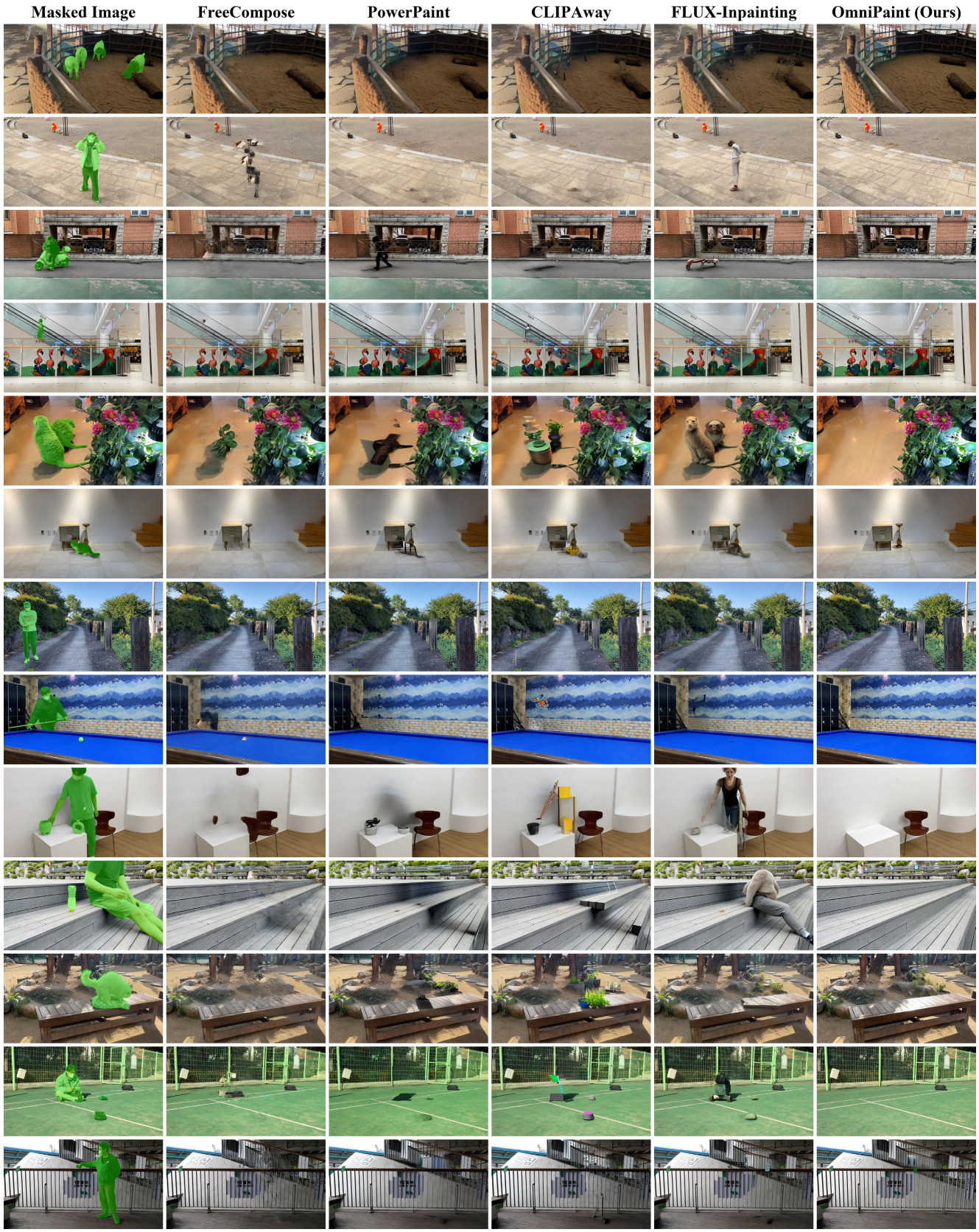

Figure G. More Qualitative comparison of object removal in challenging scenarios (Part 2). The compared methods include FreeCompose [3], PowerPaint [15], CLIPAway [4], and FLUX-Inpainting [13].

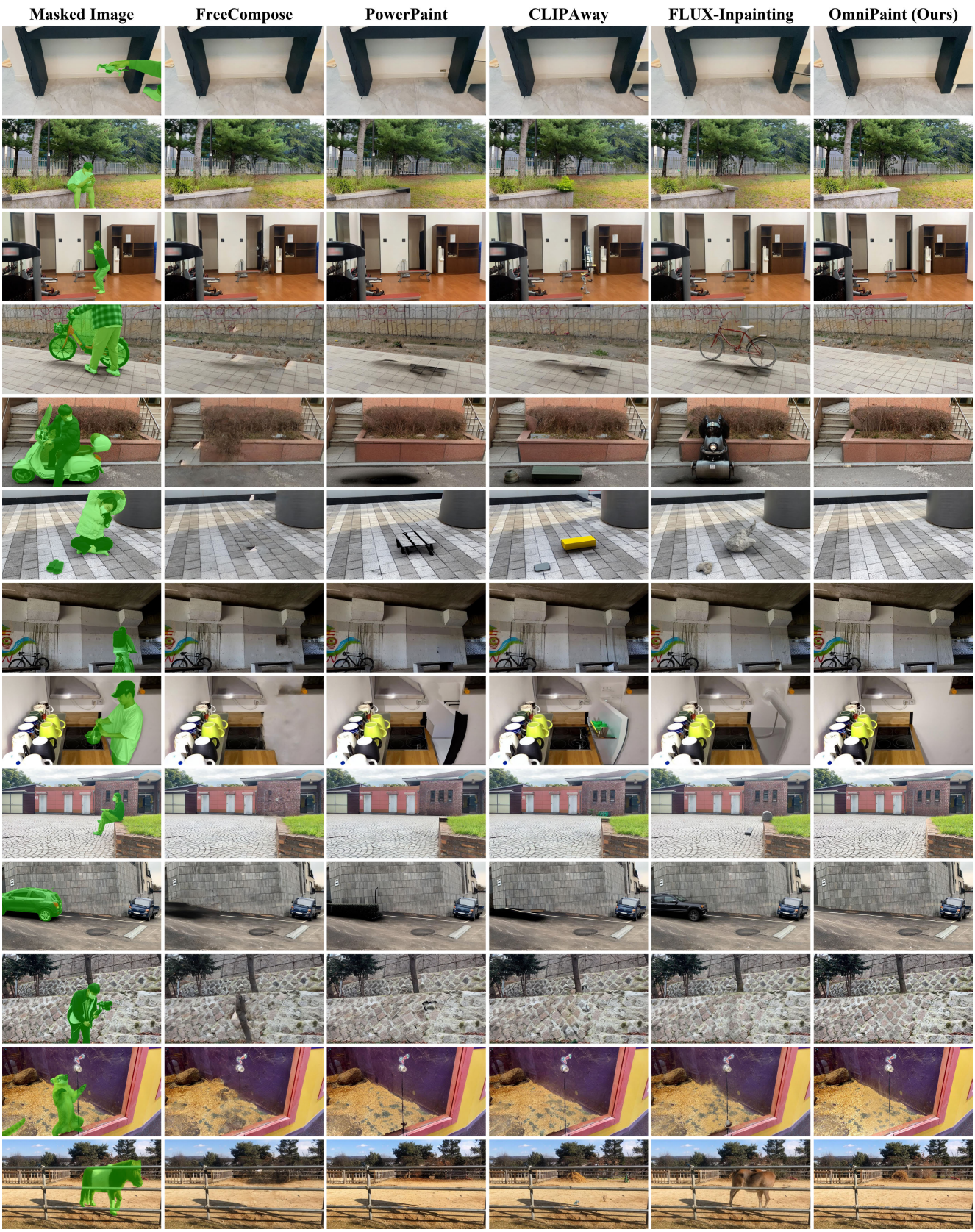

Figure H. More Qualitative comparison of object removal in challenging scenarios (Part 3). The compared methods include FreeCompose [3], PowerPaint [15], CLIPAway [4], and FLUX-Inpainting [13].

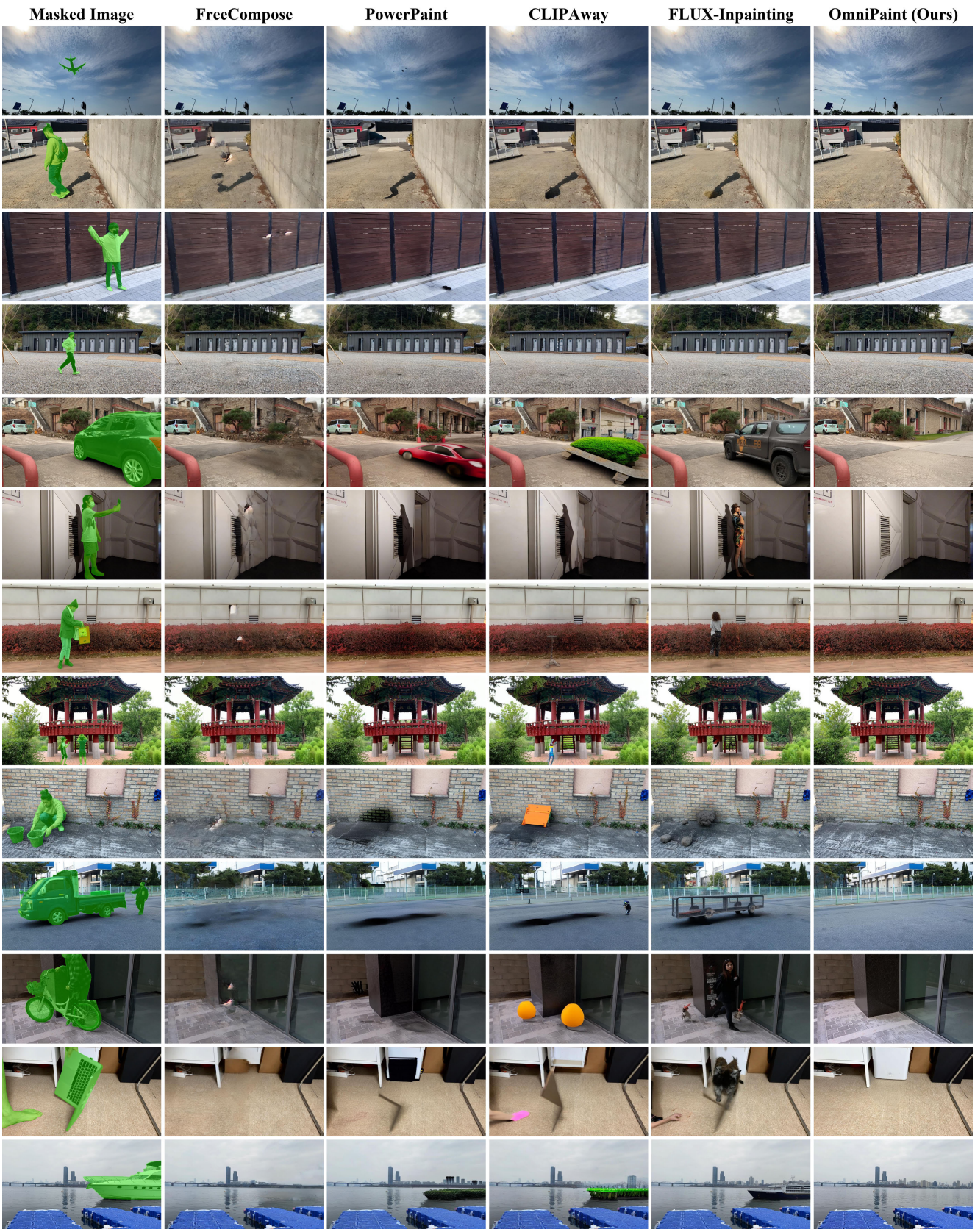

Figure I. More Qualitative comparison of object removal in challenging scenarios (Part 4). The compared methods include FreeCompose [3], PowerPaint [15], CLIPAway [4], and FLUX-Inpainting [13].

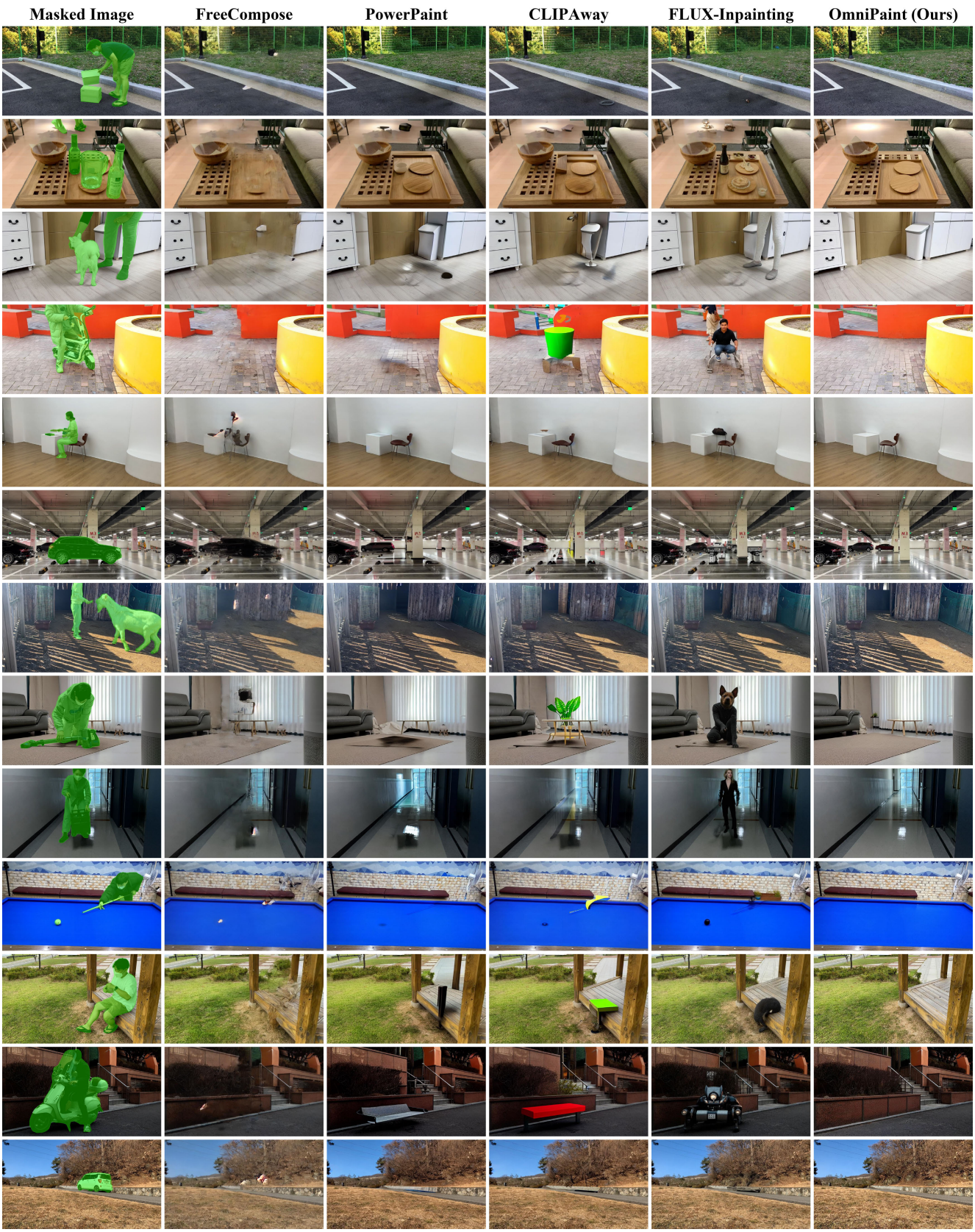

Figure J. More Qualitative comparison of object removal in challenging scenarios (Part 5). The compared methods include FreeCompose [3], PowerPaint [15], CLIPAway [4], and FLUX-Inpainting [13].

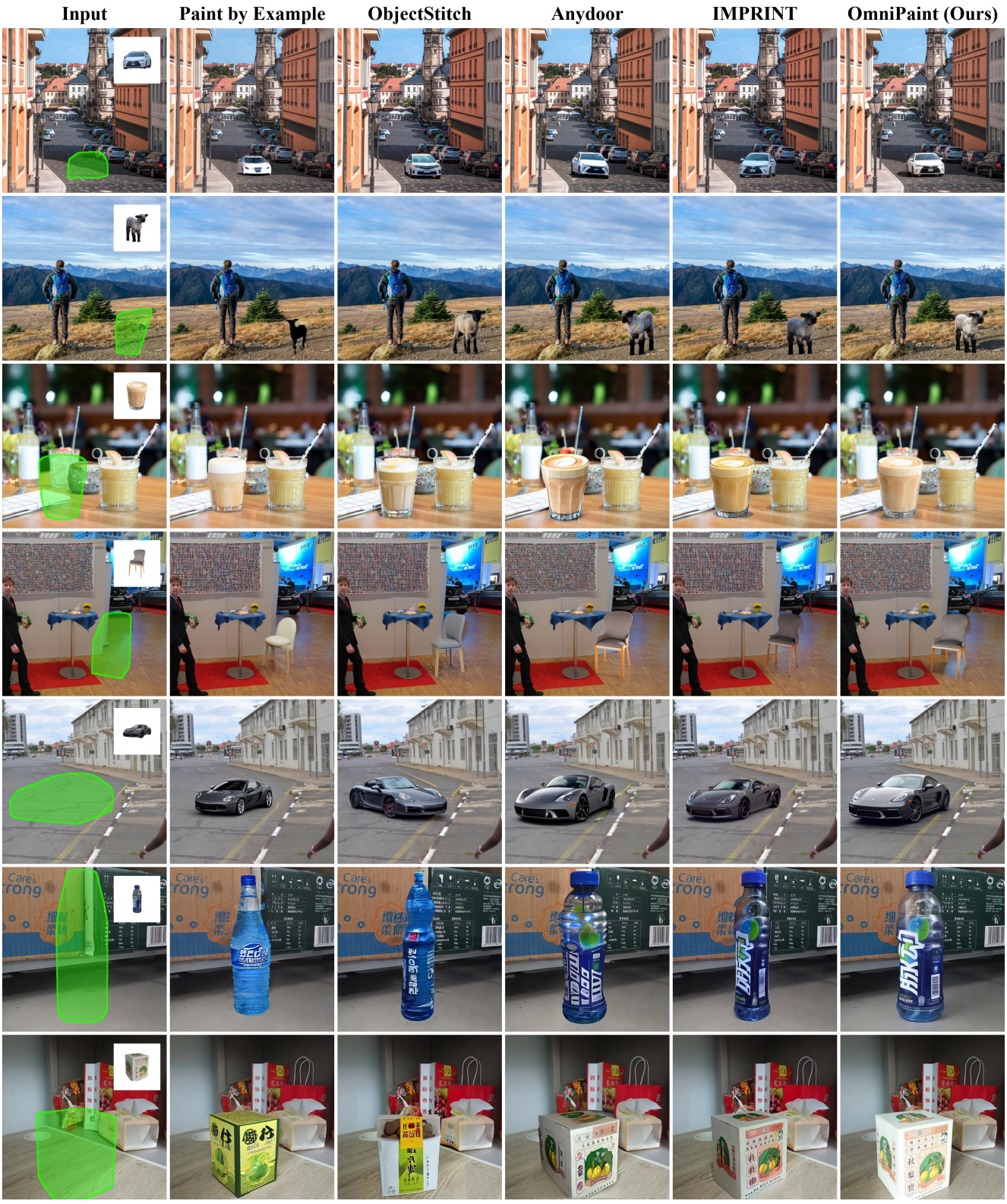

Figure K. More Qualitative comparison of object insertion in challenging scenarios. The first column is the input of object insertion setting, containing a masked image and a reference object image. The compared methods include Paint-by-Example [14], ObjectStitch [9], Anydoor [2], and IMPRINT [10].