Supplementary material:

MetaShadow: Object-Centered Shadow Detection, Removal, and Synthesis

There are ten parts in this supplementary material.

Part 1 presents details on our MOS dataset.

Part 2 presents additional details about our Shadow Analyzer.

Part 3 presents additional details and ablation study about our Shadow Synthesizer.

Part 4 presents details on our shadow-specific data augmentations.

Part 5 presents the pipeline of object relocation.

Part 6 presents general shadow removal ability of MetaShadow.

Part 7 presents additional comparisons on object-centered shadow detection and removal, as well as additional results from our Shadow Analyzer.

Part 8 presents additional comparisons on object-centered shadow synthesis.

Part 9 presents additional video results on object-centered video shadow synthesis.

Part 10 discusses a possible solution to address the limitation of our Shadow Synthesizer.

Part 1: Details on Moving Objects in the Shadow (MOS) Dataset

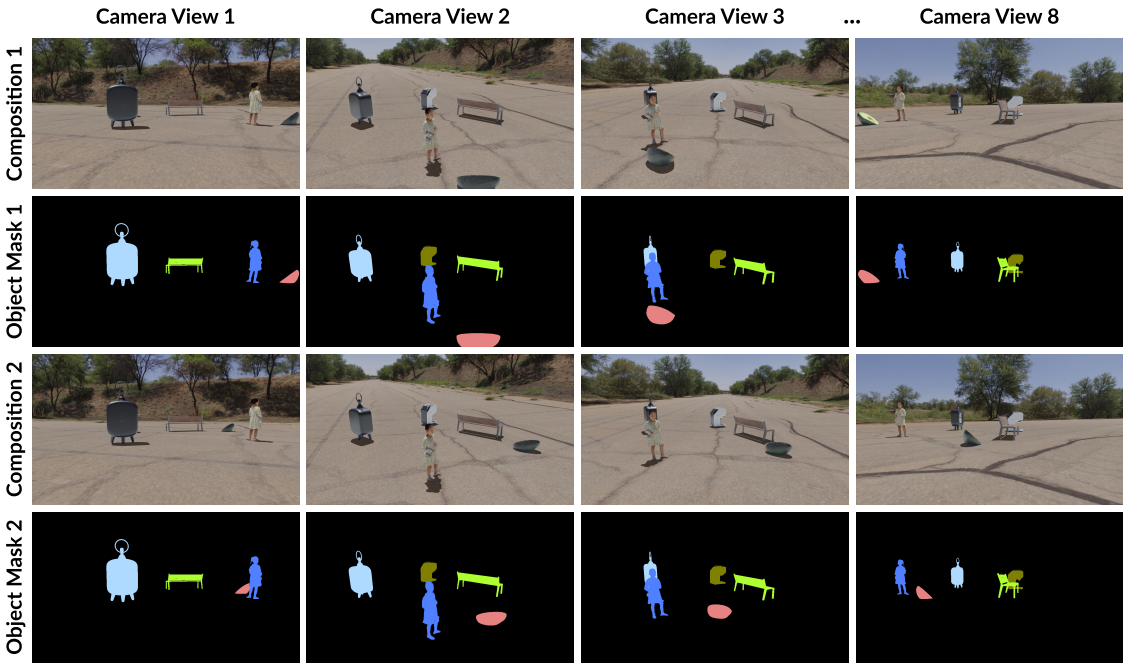


Figure 1. An example of our MOS dataset, the object mask is additional output from the render.

We compiled 1,200 free 3D assets from Sketchfab and Polyhaven with an additional 500 scanned human models to support the common application scenarios [1]. We use 200 sunny outdoor HDRIs from Polyhaven for environment maps and backgrounds. We set up a camera ring within the virtual environment and randomly place two to five objects, while ensuring no occlusion between them. We render five scenes for each set of objects, randomly picking an object from that set and placing it at a new location. We produced 8,000 images with automatically-generated object mask annotations and images with/without each object in the scene to construct the ground truths for training. We show an example in Fig 1.

By combining the MOS dataset during object-centered shadow-removal training, we observe improvements in visual quality and a reduction in grid artifacts, as shown in Fig. 2.

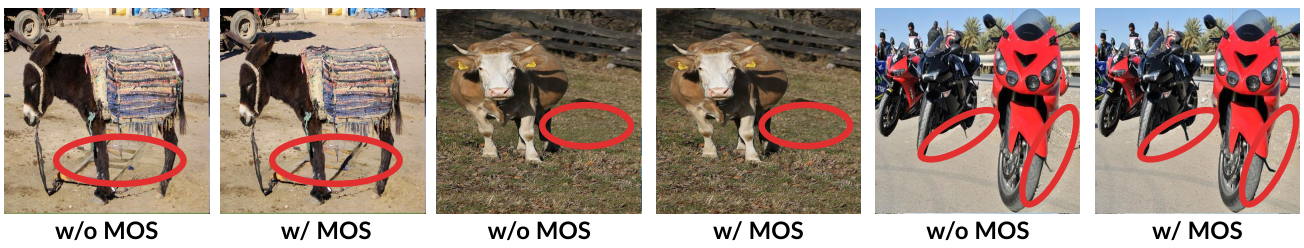


Figure 2. The MOS dataset improves our Shadow Analyzer by enhancing visual quality and reducing grid artifacts.

Part 2: Additional Details on Shadow Analyzer

Table 1. Architecture of the Proposed Shadow Detector

Stage	**Operation**	Kernel size	**Output Size**
Input	-	-	$64 \times 64 \times (384 * 4)$
Conv1	Conv2d-BN-GELU	1×1	$64 \times 64 \times 128$
Conv2	Conv2d-BN-GELU	3×3	$64 \times 64 \times 128$
Up1	ConvTranspose2d-BN-GELU	4×4	$128 \times 128 \times 128$
Conv3	Conv2d-BN-GELU	3×3	$128 \times 128 \times 128$
Conv4	Conv2d-BN-GELU	3×3	$128 \times 128 \times 32$
Up2	ConvTranspose2d-BN-GELU	4×4	$256 \times 256 \times 32$
Conv5	Conv2d-BN-GELU	3×3	$256 \times 256 \times 32$
Final	Conv2d-Sigmoid	1×1	$256 \times 256 \times 1$

Architecture of Shadow Detector. Tab. 1 shows the architecture of our proposed shadow detector, starting with a 1×1 convolution to compress input features F_s^i (1536 to 128) from GAN, followed by layers of 3×3 2D convolutions, batch normalization, and GELU activations for feature refinement. Upscaling is achieved through transposed convolutional layers. The architecture concludes with a 2D convolution with sigmoid function, resulting in a shadow mask M_s.

Architecture of Discriminator. Tab. 2 shows the architecture of our proposed discriminator based on the discriminator of StyleGAN2 [**?**]. The difference is that we add the ground truth shadow mask along with the RGB image into the discriminator to provide contextual awareness, enabling more precise evaluations of shadow removal effectiveness and guiding the generator towards more targeted improvements in shadow regions.

Table 2. Architecture of Discriminator.

Layer	**Operation**	**Kernel Size**	**Output Size**
Input	-	-	$512 \times 512 \times 4$
FromRGB	Conv2d-Leaky ReLU	1×1	$512 \times 512 \times 64$
Conv1	Conv2d-Leaky ReLU	3×3	$512 \times 512 \times 64$
Conv2	Conv2d-Leaky ReLU	3×3	$256 \times 256 \times 128$
Skip1	Conv2d-Leaky ReLU	1×1	$256 \times 256 \times 128$
Conv3	Conv2d-Leaky ReLU	3×3	$256 \times 256 \times 128$
Conv4	Conv2d-Leaky ReLU	3×3	$128 \times 128 \times 256$
Skip2	Conv2d-Leaky ReLU	1×1	$128 \times 128 \times 256$
Conv5	Conv2d-Leaky ReLU	3×3	$128 \times 128 \times 256$
Conv6	Conv2d-Leaky ReLU	3×3	$64 \times 64 \times 512$
Skip3	Conv2d-Leaky ReLU	1×1	$64 \times 64 \times 512$
Conv7	Conv2d-Leaky ReLU	3×3	$64 \times 64 \times 512$
Conv8	Conv2d-Leaky ReLU	3×3	$32 \times 32 \times 512$
Skip4	Conv2d-Leaky ReLU	1×1	$32 \times 32 \times 512$
Conv9	Conv2d-Leaky ReLU	3×3	$32 \times 32 \times 512$
Conv10	Conv2d-Leaky ReLU	3×3	$16 \times 16 \times 512$
Skip5	Conv2d-Leaky ReLU	1×1	$16 \times 16 \times 512$
Conv11	Conv2d-Leaky ReLU	3×3	$16 \times 16 \times 512$
Conv12	Conv2d-Leaky ReLU	3×3	$8 \times 8 \times 512$
Skip6	Conv2d-Leaky ReLU	1×1	$8 \times 8 \times 512$
Conv13	Conv2d-Leaky ReLU	3×3	$8 \times 8 \times 512$
Conv14	Conv2d-Leaky ReLU	3×3	$4 \times 4 \times 512$
Skip7	Conv2d-Leaky ReLU	1×1	$4 \times 4 \times 512$
MinibatchStd	MinibatchStd	-	$4 \times 4 \times 512$
Conv15	Conv2d-Leaky ReLU	3×3	$4 \times 4 \times 512$
FC	Fully Connected	-	512
Output	Fully Connected	-	1

Part 3: Additional Details on Shadow Synthesizer

Our Shadow Synthesizer in MetaShadow takes an RGB image, an object mask, and an optional shadow mask as input. The shadow mask is initially empty because when an object is moved to a new location, its shadow may take on a different shape due to perspective and changes in geometry compared to its original position. Additionally, in scenarios where objects are inserted, the shape of the shadow may not be known. Sometimes, the plane is flat and the detected shadow mask from the original location is suitable for guiding the Shadow Synthesizer to synthesize shadows. Therefore, we combined these two situations for training by using a mask type $\mathbf{m} = 0$ or 1, similar to [12], to indicate the condition for the model. Tab. 3 shows the performance gain when we take the ground-truth shadow mask as guidance. Also, we include full ablation study in this table where we did not update in the main paper. For Table 4 in the main paper, we upsampled our results to 256×256 using the Lanczos interpolation method to ensure a fair comparison. For Tab. 3, we used the original 128×128 resolution to compute all the metrics.

Method	Global RMSE ↓	Bbox PSNR↑	Bbox SSIM↑
Baseline 1: SSDM-Text [3, 10]	3.36	29.80	92.21
Baseline 2: SSDM-CLIP [3, 9]	4.51	29.72	93.17
Ours without shadow mask	2.93	30.73	93.49
Ours with shadow mask	**2.87**	**31.17**	**94.05**

Table 3. Ablation study on without or with shadow mask for Shadow Synthesizer.

More results on controllable shadow synthesis with ground-truth shadow mask are shown in 3. The shape of the synthesized shadow can follow the given ground-truth shadow mask. This indicates that Shadow Synthesizer can obtain shadow color and intensity information from the shadow knowledge of Shadow Analyzer and the accurate shadow shape information from the input shadow mask.

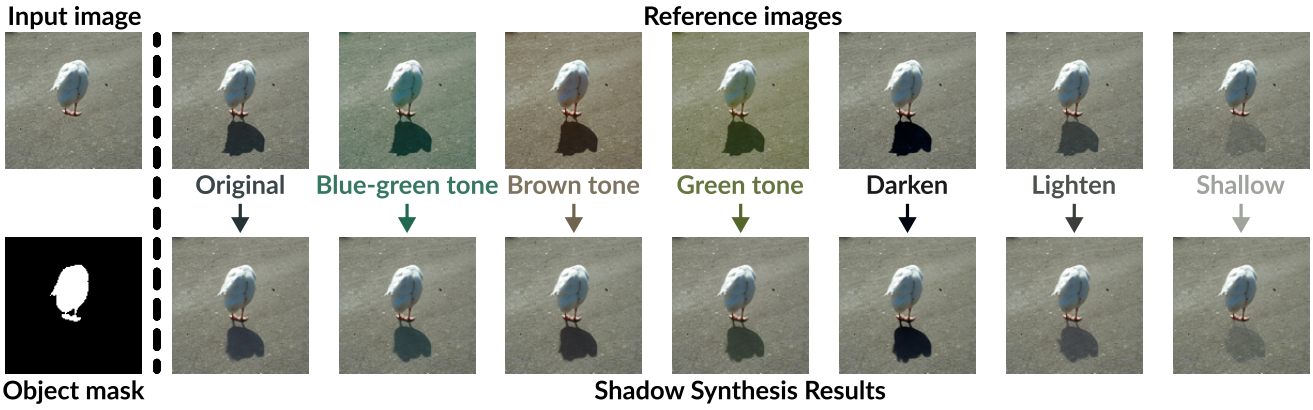


Figure 3. By editing the reference images with color or intensity, the synthesized shadow will follow the modification.

Part 4: Shadow-specific Data Augmentations

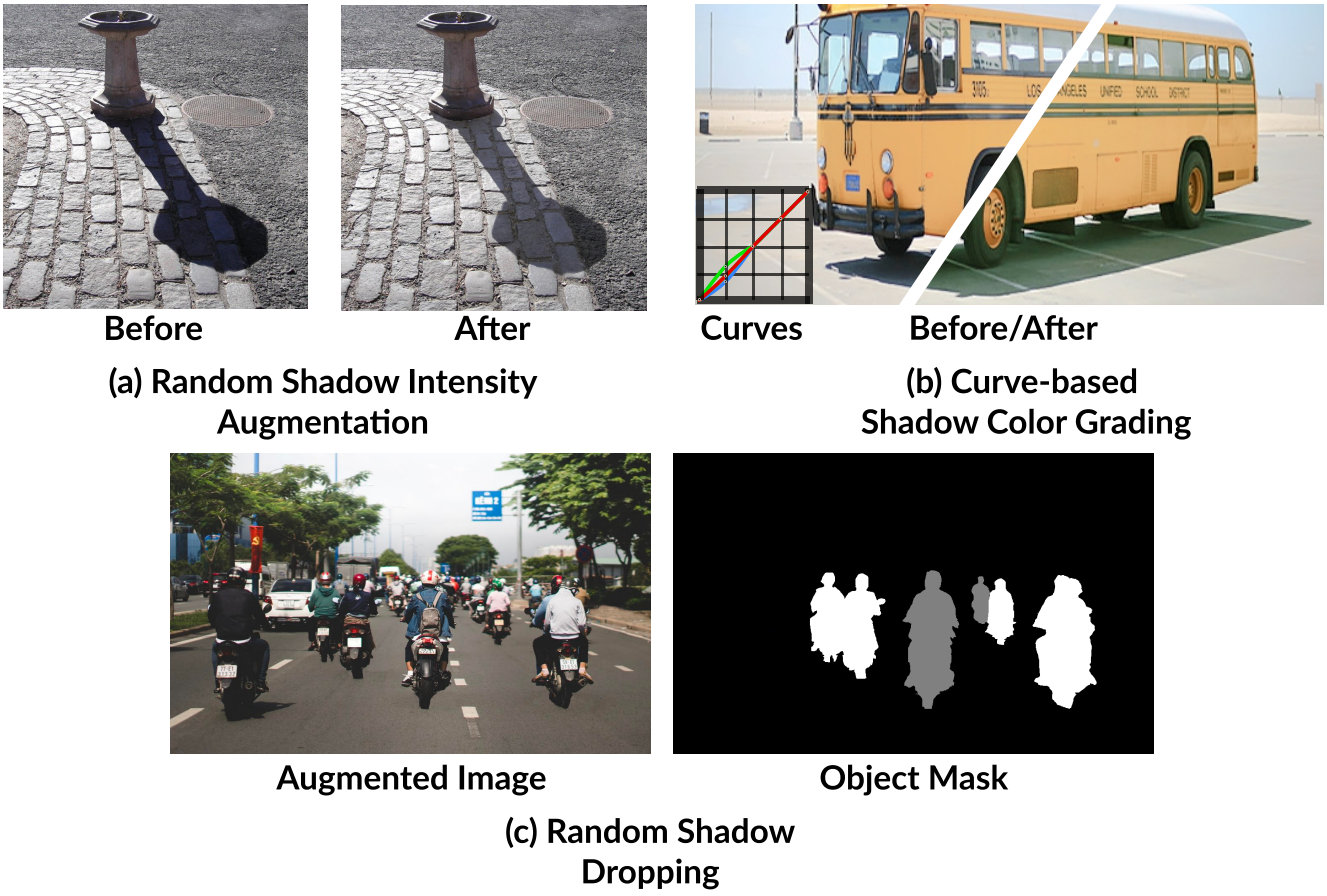


Figure 4. Shadow-specific data augmentations. In (c), the gray object mask indicates that the shadows of these objects have been removed in the augmented image and they are not included in the final object mask.

We perform three shadow-specific data augmentations to improve the model's generalizability and controllability: (i) Random shadow intensity augmentation, (ii) Curve-based shadow color grading, and (iii) Random shadow dropping.

(i) Random shadow intensity augmentation: We control the shadow intensity in images to enhance dataset diversity and model robustness, by computing the difference $\mathbf{D}$ between the shadow image and the shadow-free image and multiplying a shadow mask with a random scale factor $\mathbf{S}$, where $\mathbf{S} \in [0.7, 1.3]$

$$\begin{aligned} \mathbf{D} &= (I_s - I_{sf}) \cdot M_s, \\ \widetilde{I}_s &= I_{sf} - \mathbf{D} \times \mathbf{S}, \end{aligned} \tag{1}$$

(ii) Curve-based shadow color grading: This helps simulate varying lighting conditions and address photos with color adjustments. Inspired by color grading tools, we apply a 3D curve to randomly adjust the color of shadow regions. Specifically, as shown in Fig. 4 (b), we set five control points (ranging from 0 to 255) uniformly in each of the three color channels. We randomly add a small value (-10,10) to the second control point (64) and retain the others at original positions to keep the higher intensity values consistent.

This data augmentation also helps our MetaShadow in removing colored shadows, as shown in Fig. 5 (a-c). In other words, it enables the shadow analyzer to extract more accurate shadow information and contribute to the shadow synthesis tasks. Note that the controllable shadow synthesis ability of our MetaShadow, as shown in Fig. 7 in the main paper and Fig. 3, benefits from these two data augmentations, *i.e.*, (i) and (ii).

Figure 5. Ablation study on (ii) curve-based shadow color grading and (iii) random shadow dropping. We can observe that (a) has a green-blur tone inside the shadow region. The previous model (b) fails to eliminate this tone in the shadow-removed area. However, by employing this data augmentation, we can achieve better color consistency between the shadow-removed and non-shadow regions (c). From another set of images, we can observe that, without random shadow dropping, the Shadow Synthesizer will synthesize shadow for objects that do not have shadows and ignore the object mask. With random shadow dropping, the Shadow Synthesizer now only synthesizes shadow for the object given in the object mask.

(iii) Random shadow dropping: In training the Shadow Synthesizer, we found that it may ignore the object mask and synthesize shadows for objects without shadows. This is caused by the fact that the input images are always without shadows, while the ground truths always have shadows. To address this, we randomly drop the shadows of some objects and exclude these objects in the object mask to encourage the model to be aware of the information in the object mask. Fig. 5 (d-f) illustrates the effectiveness of this data augmentation.

Part 5: The Pipeline of Object Relocation

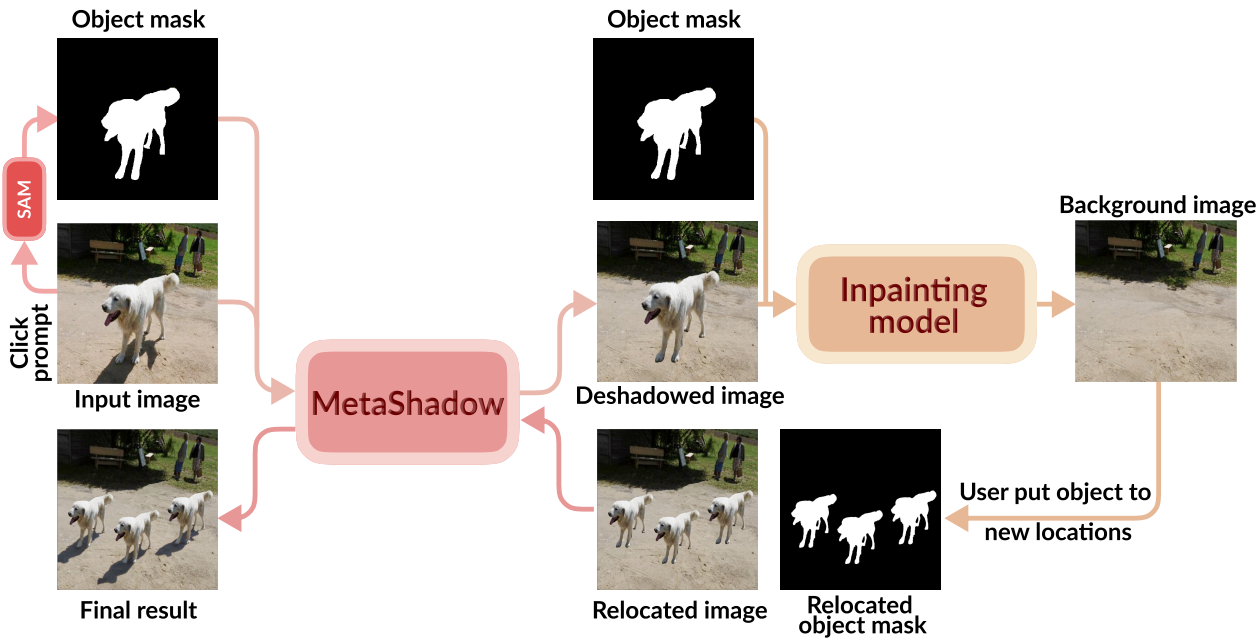


Figure 6. The pipeline of object relocation.

Fig. 6 presents the pipeline of our object relocation application. We initially feed the input image with the user's click prompt to the Segment Anything Model (SAM) [6] to generate an object segmentation mask. Then, our MetaShadow framework takes the input image and object mask to perform object-centered shadow removal. Next, we remove the object by using CMGAN [14], an image inpainting model, to get the background image. Based on the segmentation mask of the object, we can let the user relocate the object in the background image and also produce the relocated object mask. Finally, the MetaShadow takes the relocated image and relocated object mask to synthesize shadows for the object inside the object mask.

We use this pipeline to generate the result of Fig. 1 in the main paper and also use the right part of this pipeline to produce our Moving DESOBA dataset.

Part 6: General Shadow Removal Ability of MetaShadow

As shown in Fig. 20, our Shadow Analyzer can detect and remove general cast shadows when the object mask is empty, with each pixel of the mask being zero. This ability is contributed by our unique training scheme as introduced in the main paper. By utilizing this capability, our MetaShadow can synthesize shadows by using the cast shadow in the image, as shown in Fig. 7.

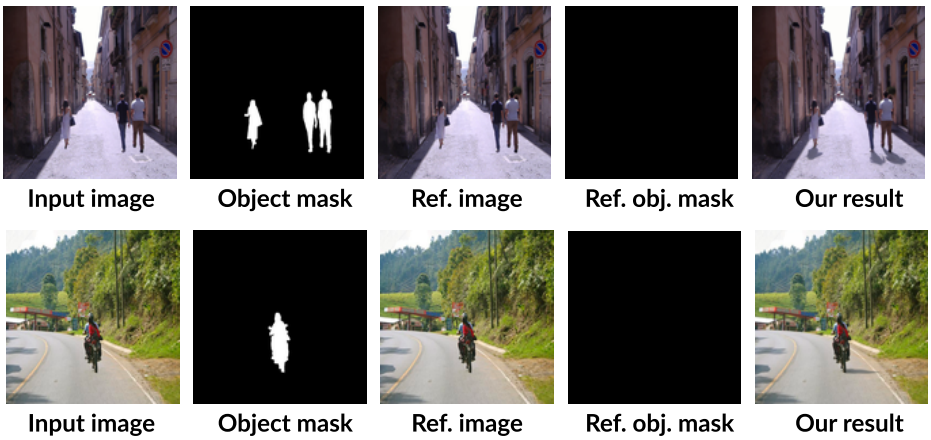


Figure 7. Synthesizing shadow by using the information of cast shadows.

Part 7: Additional Comparison on Object-Centered Shadow Removal

We present comparisons on the DESOBA Dataset [4]. Some of our results even surpass the reference ground truth, whose shadows were removed by experts using Photoshop [4]. ShadowDiffusion [2], even when fine-tuned with our settings (datasets and data augmentations), fails to eliminate the shadows. Fig. 14 shows our Shadow Analyzer's capability to remove soft shadows.

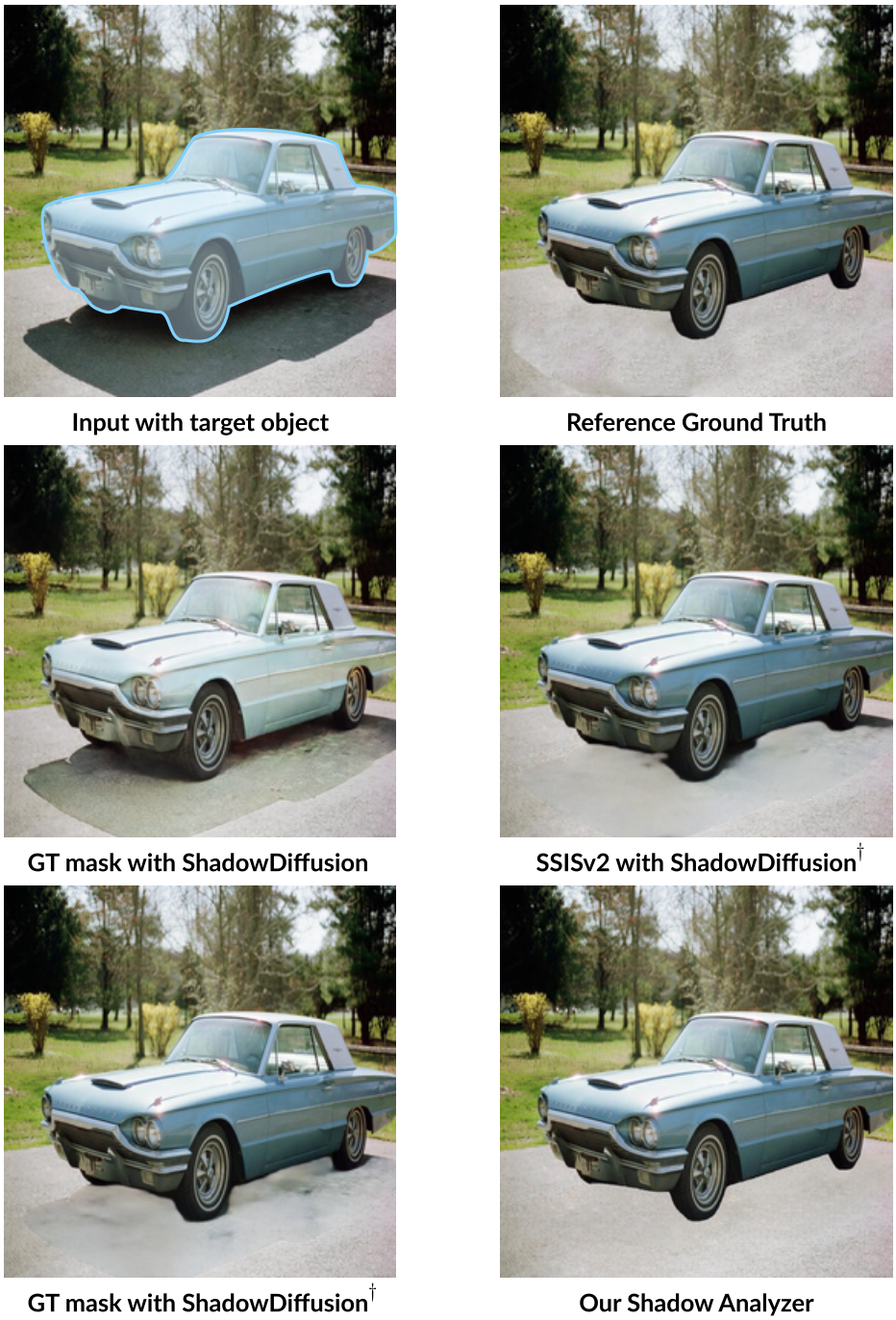


Figure 8. Additional comparison #1 on object-centered shadow removal.

Input with target object

Reference Ground Truth

GT mask with ShadowDiffusion

SSISv2 with ShadowDiffusion†

GT mask with ShadowDiffusion†

Our Shadow Analyzer

Figure 9. Additional comparison #2 on object-centered shadow removal.

Input with target object

Reference Ground Truth

GT mask with ShadowDiffusion

SSISv2 with ShadowDiffusion†

GT mask with ShadowDiffusion†

Our Shadow Analyzer

Figure 10. Additional comparison #3 on object-centered shadow removal.

Input with target object

Reference Ground Truth

GT mask with ShadowDiffusion

SSISv2 with ShadowDiffusion†

GT mask with ShadowDiffusion†

Our Shadow Analyzer

Figure 11. Additional comparison #4 on object-centered shadow removal.

Input with target object

Reference Ground Truth

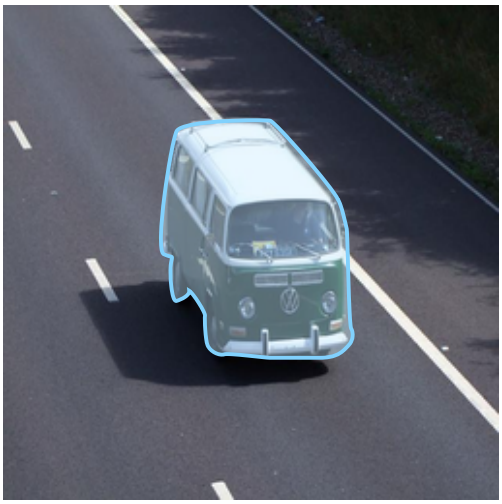
GT mask with ShadowDiffusion

SSISv2 with ShadowDiffusion†

GT mask with ShadowDiffusion†

Our Shadow Analyzer

Figure 12. Additional comparison #5 on object-centered shadow removal.



Input with target object

Reference Ground Truth

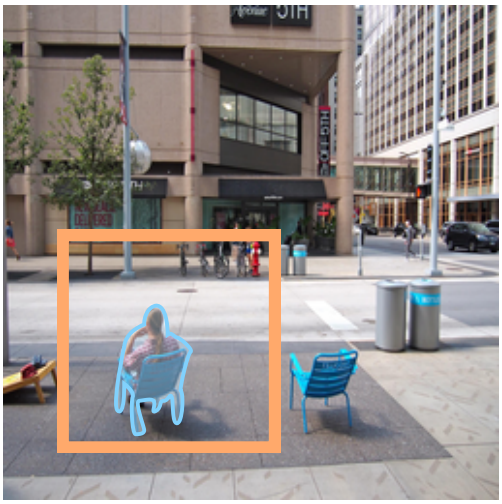
GT mask with ShadowDiffusion

SSISv2 with ShadowDiffusion†

GT mask with ShadowDiffusion†

Our Shadow Analyzer

Figure 13. Additional comparison #6 on object-centered shadow removal.



Input with target object

Reference Ground Truth

GT mask with ShadowDiffusion

SSISv2 with ShadowDiffusion†

GT mask with ShadowDiffusion†

Our Shadow Analyzer

Figure 14. Additional comparison #7 on object-centered shadow removal.

Part 7: Additional Results from Shadow Analyzer

Here we show additional object-centered shadow detection and removal results on real-world images from the Web. The scenarios include soft shadows (Fig. 15(1), Fig. 16(3), Fig. 17(1), Fig. 19(3)), complex textures (Fig. 15(2), Fig. 16(1), Fig. 17(3), Fig. 18(1)), complex geometry and overlap (Fig. 15(2), Fig. 17(2), Fig. 18(3)), and colored shadows (Fig. 18(2), Fig. 19(1,2)). The number in () indicates the column number.

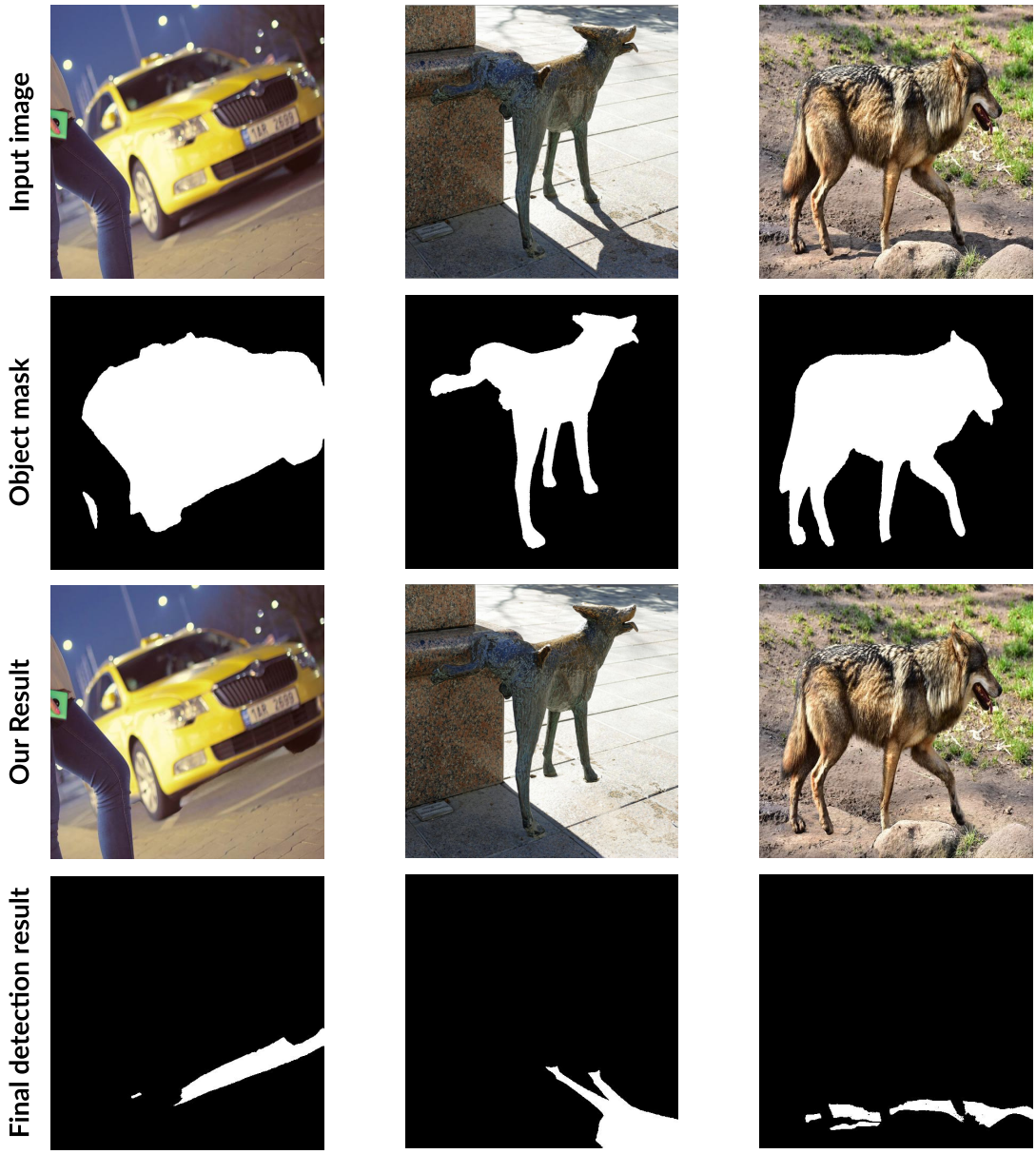


Figure 15. Additional result #8 on object-centered shadow detection and removal from our Shadow Analyzer.

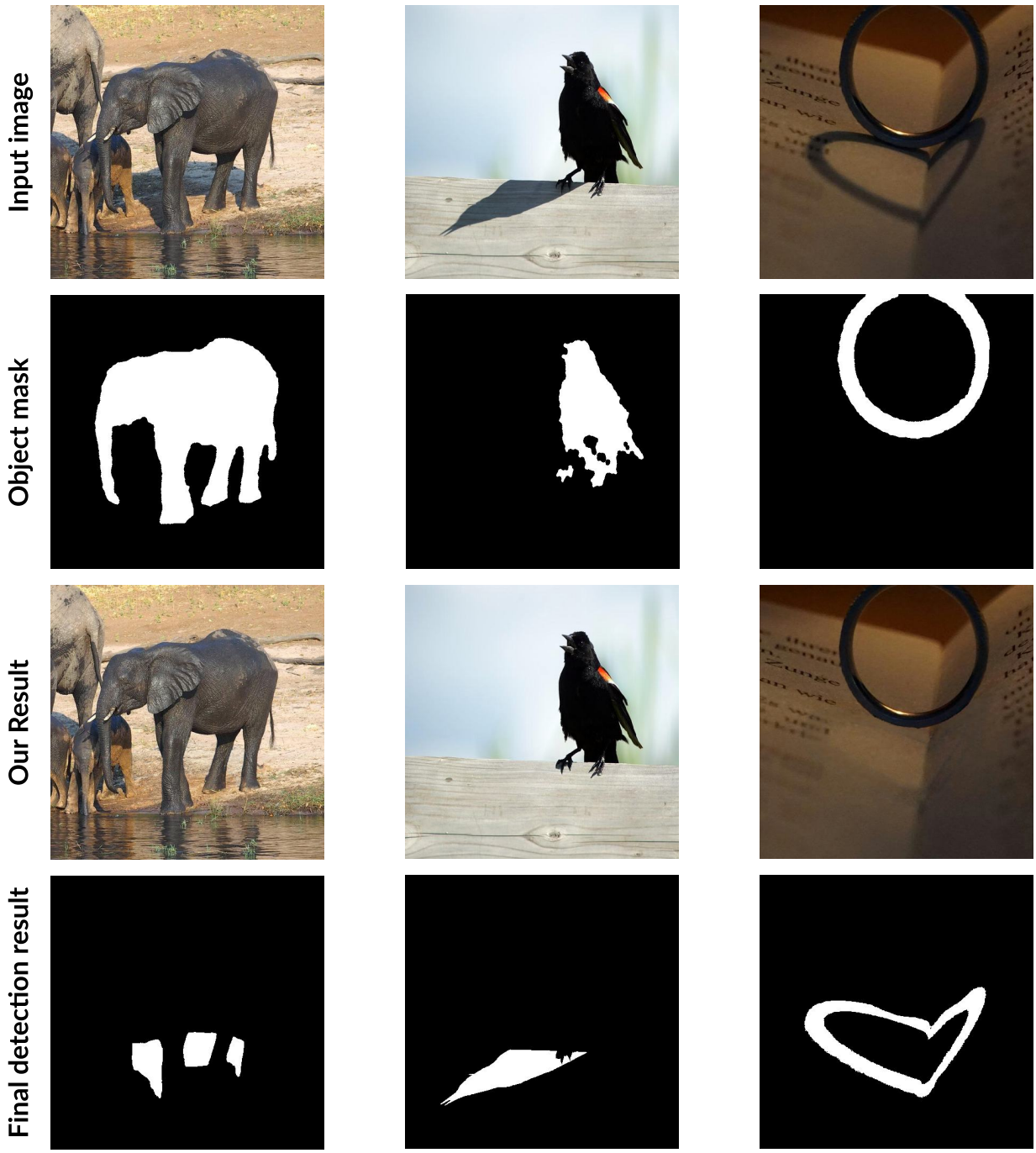


Figure 16. Additional result #9 on object-centered shadow detection and removal from our Shadow Analyzer.

Figure 17. Additional result #10 on object-centered shadow detection and removal from our Shadow Analyzer.

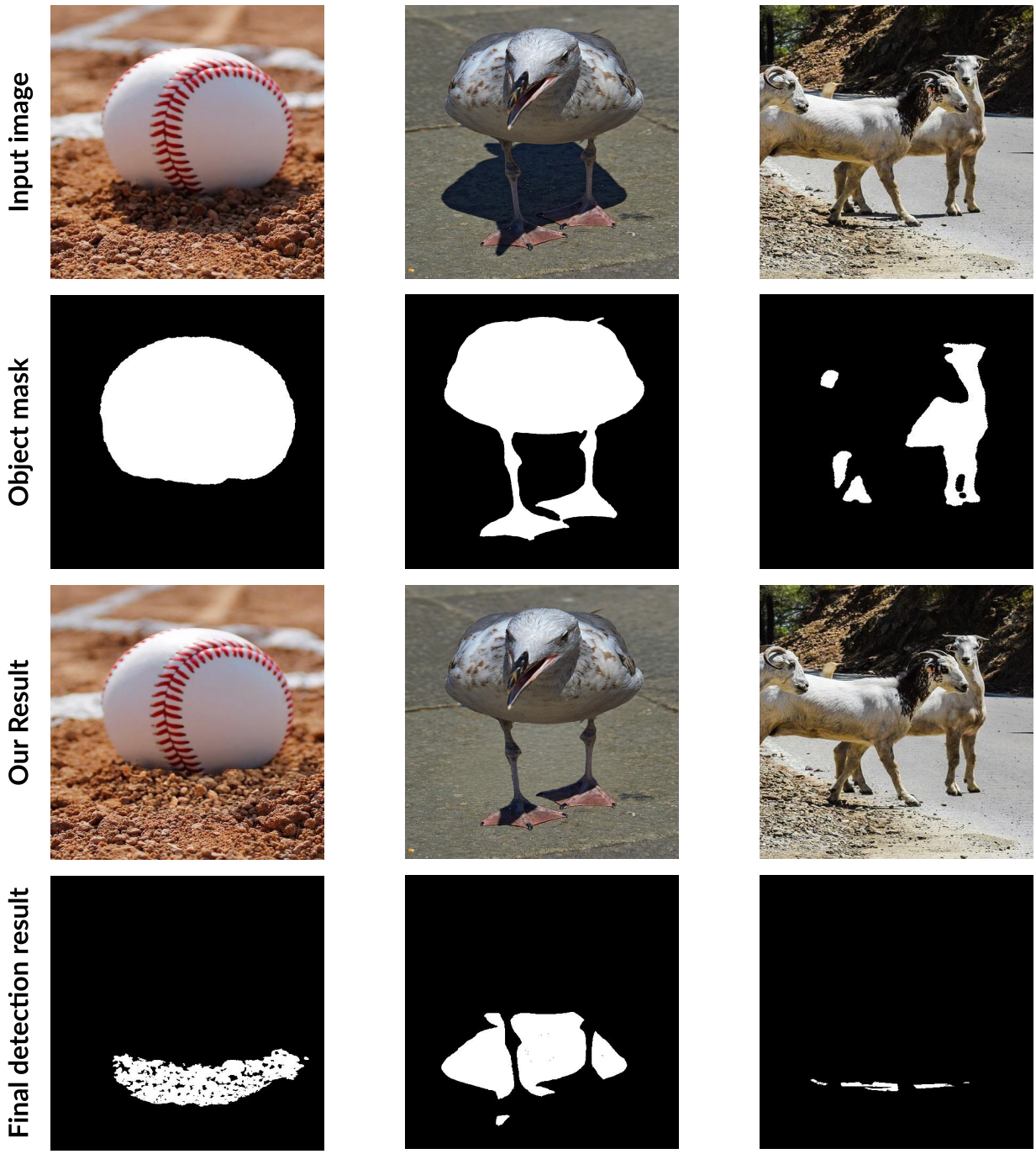


Figure 18. Additional result #11 on object-centered shadow detection and removal from our Shadow Analyzer.

Figure 19. Additional result #12 on object-centered shadow detection and removal from our Shadow Analyzer.

Figure 20. Additional result #13 on object-centered shadow removal from our Shadow Analyzer. In the last row, we also provide the results of general shadow removal when the input object mask is empty.

Part 8: Additional Results on Object-Centered Shadow Synthesis

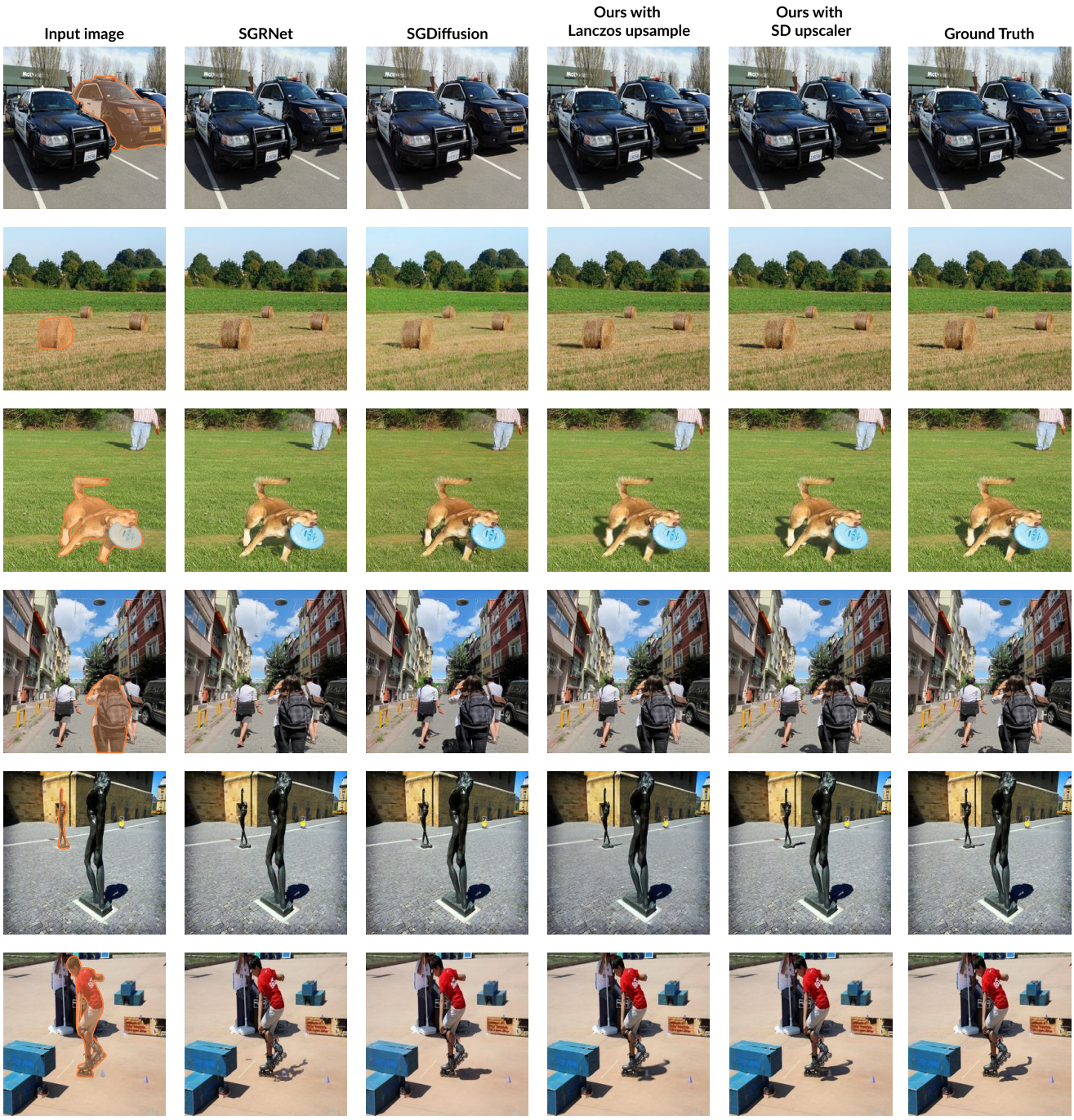


Figure 21. Additional result #1 on object-centered shadow synthesizing from our MetaShadow.

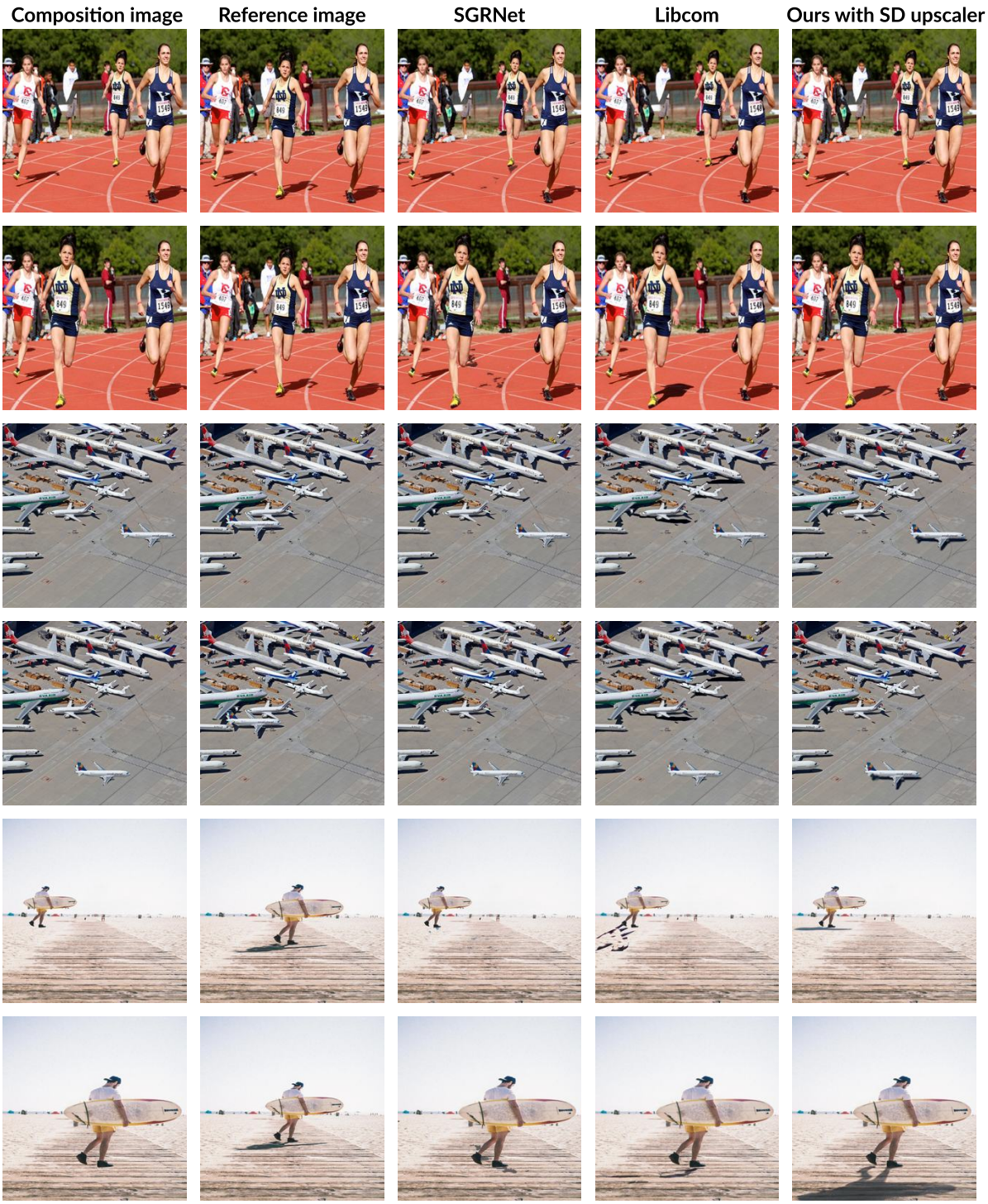


Figure 22. Additional result #2 on object-centered shadow synthesizing from our MetaShadow.

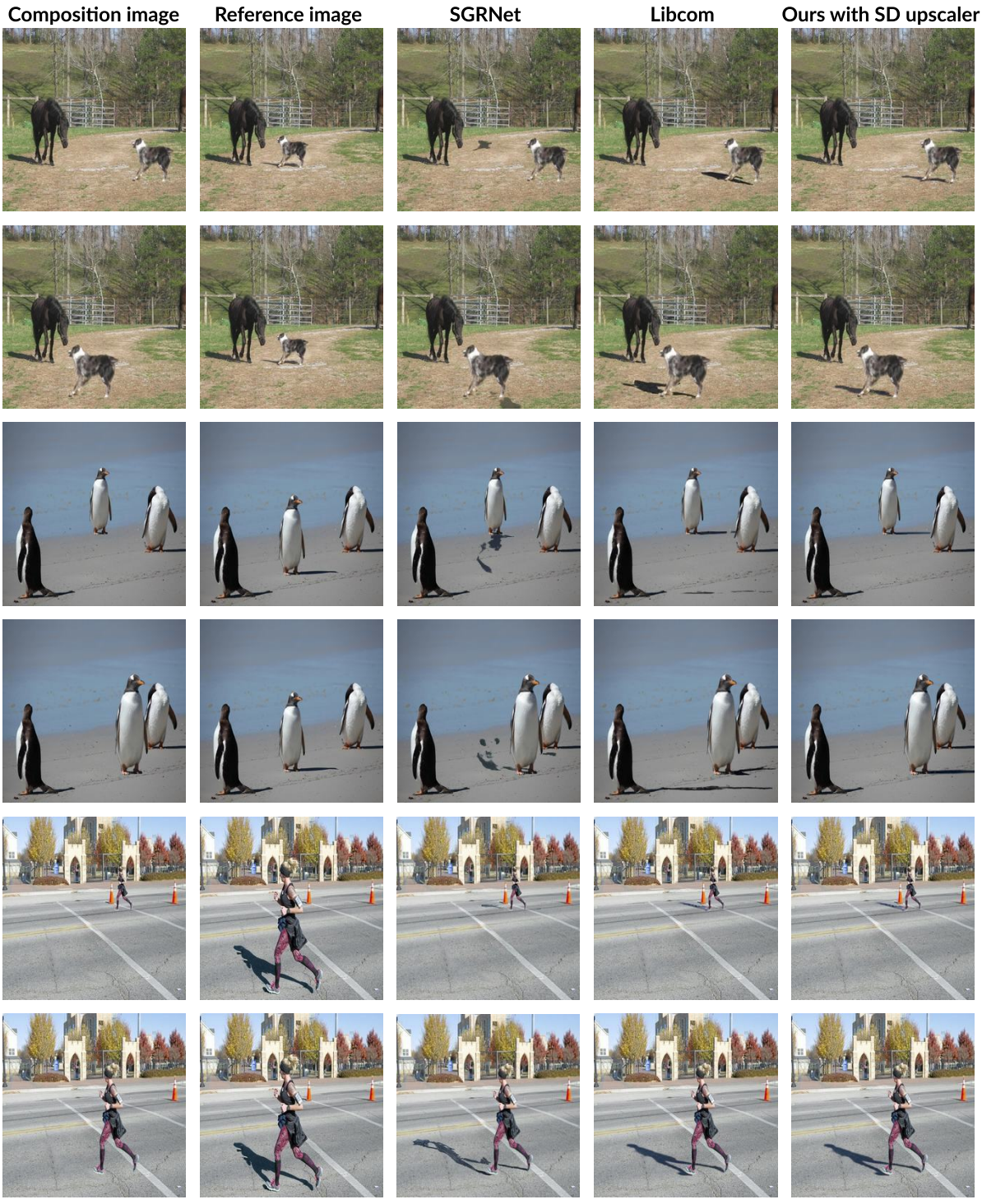


Figure 23. Additional result #3 on object-centered shadow synthesizing from our MetaShadow.

Part 9: Additional Video Results On Object-Centered Video Shadow Synthesis

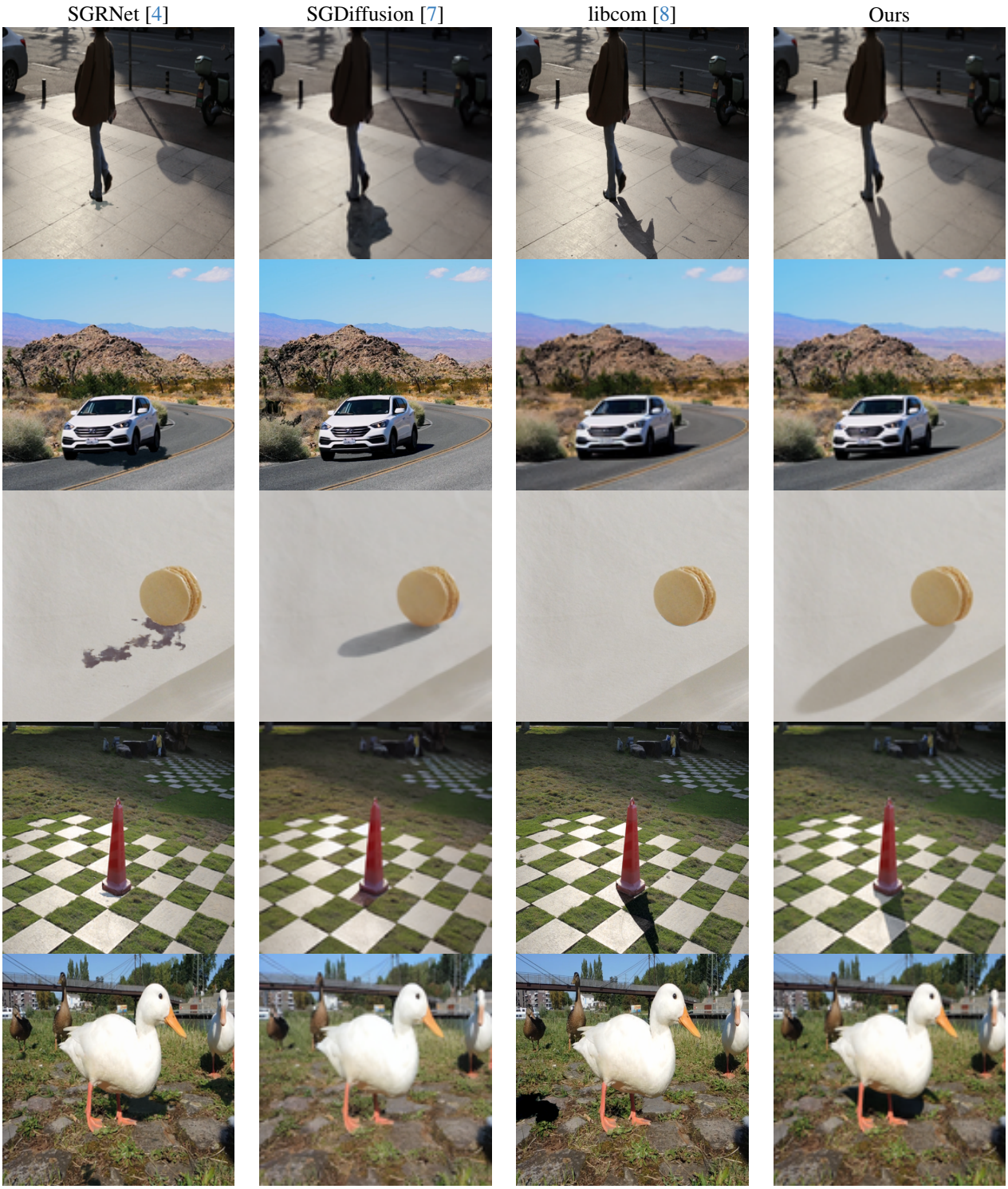


Table 4. Additional video results tested on the Video DESOBA dataset. We take the first frame as the reference and synthesize the shadow for the remaining frames. Please use Adobe Acrobat to see the GIFs or check the GIF folder provided as part of the supplementary material.

Part 10: Limitation, Possible Solution, And Future Works

Since MetaShadow is the first attempt to join these different shadow tasks, our framework has several limitations. The resolution of the current Shadow Synthesizer is limited to 128×128. A possible solution is to upsample the shadow region and then replace the region in the high-resolution input image with an upsampled shadow region. However, even for the Lanczos algorithm, the resulting shadow may become blurry, as shown in Fig. 24 (b). Another way is to use a deep upscaler, like the Stable Diffusion upscaler [11]. Upsampling the entire image directly may lead to numerous artifacts compared to the original image. Therefore, we utilize our predicted shadow mask solely for upsampling the shadow regions, as shown in Fig. 24 (a). This is the default solution that we employed to upsample our results to 256×256 for a visual comparison with SGRNet [4], which has an output resolution of 256×256. There are other options like leveraging advanced techniques such as latent diffusion models [11] or exploring the efficacy of GAN-based upscaling methods like GigaGAN [5] or diffusion-based upscaling methods like SUPIR [13]. This is left as future work.

Another notable limitation is accurately handling scenarios with multiple light sources. This issue primarily stems from the limited diversity in our current dataset. To improve this, our future work will not only incorporate sophisticated rendering techniques for indoor scenes but also extend our dataset by including real-world videos.

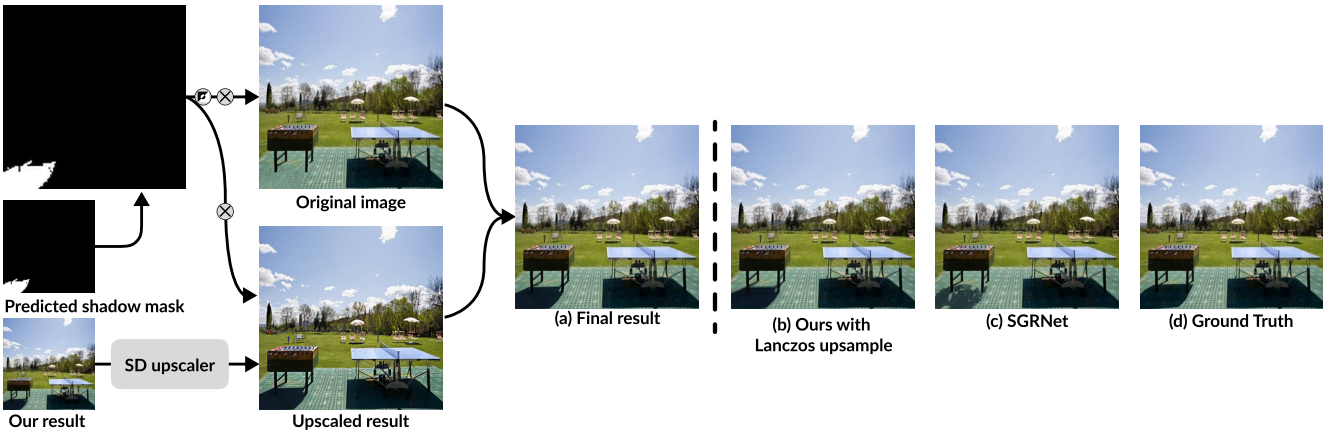


Figure 24. A possible solution to address the limitation of our Shadow Synthesizer.

References

[1] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3D objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 3

[2] Lanqing Guo, Chong Wang, Wenhan Yang, Siyu Huang, Yufei Wang, Hanspeter Pfister, and Bihan Wen. ShadowDiffusion: When degradation prior meets diffusion model for shadow removal. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 11

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020. 6

[4] Yan Hong, Li Niu, and Jianfu Zhang. Shadow generation for composite image in real-world scenes. In *AAAI Conference on Artificial Intelligence*, 2022. 11, 27, 28

[5] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 28

[6] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *IEEE International Conference on Computer Vision*, pages 4015–4026, 2023. 9

[7] Qingyang Liu, Junqi You, Jianting Wang, Xinhao Tao, Bo Zhang, and Li Niu. Shadow generation for composite image using diffusion model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 27

[8] Li Niu, Wenyan Cong, Liu Liu, Yan Hong, Bo Zhang, Jing Liang, and Liqing Zhang. Making images real again: A comprehensive survey on deep image composition. *arXiv preprint arXiv:2106.14490*, 2021. 27

[9] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 6

[10] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 2020. 6

[11] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 28

[12] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. SmartBrush: Text and shape guided object inpainting with diffusion model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 6

[13] Fanghua Yu, Jinjin Gu, Zheyuan Li, Jinfan Hu, Xiangtao Kong, Xintao Wang, Jingwen He, Yu Qiao, and Chao Dong. Scaling up to excellence: Practicing model scaling for photo-realistic image restoration in the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 28

[14] Haitian Zheng, Zhe Lin, Jingwan Lu, Scott Cohen, Eli Shechtman, Connelly Barnes, Jianming Zhang, Ning Xu, Sohrab Amirghodsi, and Jiebo Luo. CM-GAN: Image inpainting with cascaded modulation GAN and object-aware training. In *European Conference on Computer Vision*, 2022. 9