Supplementary material for LBM: Latent Bridge Matching for Fast Image-to-Image Translation

Clément Chadebec Onur Tasar Sanjeev Sreetharan Benjamin Aubin
Jasper Research
{name}.{surname}@jasper.ai

A. Training details

In this section, we provide any relevant parameters used to train our models.

A.1. Object-removal task

For the object-removal task, we trained our model for 20k iterations on 2 H100 GPUs. We set $\sigma = 0.05$ and used the timestep distribution we propose in the main paper *i.e.* $\pi(t) = \mathcal{U}(i/4)_{i\in\{0,1,2,3\}}$. We use a bucketing strategy as proposed in [6] allowing us to handle multiple aspect ratios and resolutions. This strategy consists of defining buckets with pre-defined aspect ratios and pixel budgets and filling them with the data flow. During each training iteration, a target pixel budget is sampled and then the upcoming images are assigned to the bucket with the closest aspect ratio and budget and are resized accordingly. We use the following bucket pixel budgets: $[256^2, 512^2, 768^2, 1024^2]$ sampled with probabilities $[0.1, 0.2, 0.2, 0.5]$. For each budget we consider aspect ratios ranging from 0.25 to 4. The batch sizes are respectively set to 32, 16, 8 and 4 for each budget. We trained the model with LPIPS pixel loss with weight $\lambda = 10$ and a learning rate of $3e^{-5}$ and we used the AdamW optimizer [5]. For data sources, we randomly sampled data from the RORD train set, our synthetic dataset or our in-the-wild dataset with probabilities $[0.3, 0.3, 0.4]$. For the latter, we used the random masking strategy proposed in [13] while for RORD and our synthetic dataset we used the provided semantic masks. The denoiser is initialized using the weights of the pre-trained text-to-image model SDXL [6].

A.2. Depth estimation

For depth estimation, we trained our model for 20k iterations on 2 H100 GPUs. We set $\sigma = 0.005$ and set $\lambda = 50$ for the pixel loss (LPIPS) scale. We used the following timestep distribution $\pi(t) = 0.025 \cdot \delta_{t=0.75} + 0.05 \cdot \delta_{t=0.5} + 0.025 \cdot \delta_{t=0.25} + 0.9 \cdot \delta_{t=0}$ to favor 1 step inference. We use a batch size of 4 and trained the model with a combination of *hypersim* [7] (40%), *virtual KITTI* [1] (10%) and replica [12] (50%) datasets. For *virtual KITTI*, as is common, we set the far plane to 80m. The learning rate is set to $4e^{-5}$ and we used the AdamW optimizer during training.

A.2.1. Normal estimation

For surface normal estimation, we trained an LBM model for 25k iterations on 2 H100 GPUs. We set $\sigma = 0.1$ and $\lambda = 50$ and used a pixel loss chosen as L1. We used the following timestep distribution $\pi(t) = 0.05 \cdot \delta_{t=0.75} + 0.1 \cdot \delta_{t=0.5} + 0.05 \cdot \delta_{t=0.25} + 0.8 \cdot \delta_{t=0}$ to favour 1 step inference. We used a batch size of 4 and trained the model with a combination of *hypersim* [7] (20%), *virtual KITTI* [1] (10%) and replica [12] (70%) datasets. The learning rate is set to $4e^{-5}$ and we used the AdamW optimizer during training.

A.2.2. Image relighting

In the case of image relighting, we trained a LBM model for 20k iterations on 2 H100 GPUs. We set $\sigma = 0.01$ and $\lambda = 10$ and used a LPIPS pixel loss. We used the same timestep distribution and the same data bucketing strategy as for the object-removal task with the same bucket pixel budgets and probabilities. The training data is composed of synthetic data created using the rendering engine (90%) and in-the-wild data (10%). We trained the model with a learning rate of $3e^{-5}$ together with the AdamW optimizer.

A.2.3. Controllable shadow generation and controllable image relighting

For these experiments, we trained a conditional LBM for 19k iterations using a pixel loss scale set to $\lambda = 2.5$ with LPIPS loss. We used a timestep distribution $\pi(t)$ similar to the one used for the object-removal task. We used a batch size of 4 and trained the model with a learning rate set to $5e^{-5}$ together with AdamW optimizer. The light map conditioning is injected by concatenating it in the latent space along the channels axis. In these cases, we only trained with the synthetic data created using the rendering engine.

B. Additional object-removal results

In this appendix, we provide additional results for the object removal task. In this case, instead of considering the

coarse semantic masks from RORD validation set, we consider the fine semantic masks precisely indicating the object to remove from the source image. We provide in Tab. 1, the same metrics as in the main paper. Similar to what was observed in the previous experiment, the proposed model is again able to reach the best results.

Method (NFE)	FID ↓	Local FID ↓	fMSE ↓	PSNR ↑	SSIM ↑
LAMA (1)	30.43	36.69	2450.60	19.74	56.45
SDXL inp. (50)	42.55	45.35	3976.95	20.06	69.49
PowerPaint (50)	40.61	40.35	3673.91	20.71	66.85
AE (50)	<u>18.43</u>	<u>22.24</u>	<u>1772.99</u>	<u>22.81</u>	<u>70.79</u>
Ours (1)	**15.50**	**15.62**	**1024.67**	**24.28**	**73.10**

Table 1. Metrics for object-removal task with models fine-tuned on RORD train set and evaluated on RORD validation set (52k images) using the fine semantic masks. Our method uses a single NFE. Best results are highlighted in bold, second best are underlined.

For the sake of completeness, we also fine-tune LAMA, SDXL-inpaint., PowerPaint and our LBM checkpoint (Attentive Eraser is training-free) only on RORD train set such that all the models see approx. 400k samples, which was enough to reach convergence. For the sake of completeness, we also train a LBM model from scratch only on the RORD train set with the same number of iterations. We share the results in Tab. 2. As shown in the table, while this fine-tuning step improves competitors' results, in particular for fine masks, our method still outperforms competitors for most metrics. Also note that our initial model is 047 trained on 2 H100 for ≈18h vs. 240h on 8 V100 for LAMA.

Method	FID↓		Local FID↓		fMSE↓		PSNR↑		SSIM↑		Inf.
	Coa.	Fin.	Coa.	Fin.	Coa.	Fin.	Coa.	Fin.	Coa.	Fin.	time (s)
LAMA	30.3	21.4	38.0	28.2	<u>1592.2</u>	<u>1350.3</u>	19.7	20.6	55.9	57.1	**0.1**
SDXL-inp.	<u>27.2</u>	18.5	**27.3**	<u>18.0</u>	2297.3	2213.1	19.8	21.4	64.9	69.0	7.2
PowerPaint	29.9	27.0	<u>30.0</u>	23.7	2871.2	2679.7	18.5	19.9	58.3	63.4	4.2
AE	29.7	<u>18.4</u>	33.2	22.2	2029.0	1773.0	<u>20.9</u>	<u>22.8</u>	<u>65.7</u>	<u>70.8</u>	8.0
Ours	**26.9**	**15.7**	30.5	**15.6**	**1306.6**	**997.4**	**22.5**	**24.5**	**69.2**	**73.2**	<u>0.3</u>
Ours (scratch)	27.9	16.7	30.7	16.9	1329.5	1032.2	22.4	24.4	69.0	72.9	0.3

Table 2. Metrics for object-removal task computed on RORD validation set using the coarse (Coa.) and fine (Fin.) masks. Our method and LAMA use a single neural function evaluation (NFE), others use 50 NFEs. Inference time is averaged over 50 images and computed on a single H100 GPU.

C. Results for depth estimation

As mentioned in the main paper, we also consider the monocular depth estimation task which consists of estimating a depth map from a two dimensional image. We provide in Tab. 3 the zero-shot results of our method compared to the state-of-the-art methods on commonly used evaluation datasets such as NYUv2 [11], KITTI [3], ETH3D [10], Scannet [2] and DIODE [14]. As shown in the table , the proposed method is able to outperform or be competitive with the state-of-the-art methods and achieves the best average ranking across all metrics and datasets.

Figure 1. Failure cases for object-removal. In the first row the model is not able to remove completely the shadow underneath the object. In the second row the model is not able to remove the reflection on the glass.

D. Failure cases

In this section, we present some identified failure cases of our model for the different tasks considered.

D.1. Object-removal

For object-removal, we noticed that our method can remove shadows more efficiently than all the existing methods as shown in the main paper, but there still exists some cases where it is not able to remove the shadow perfectly. Moreover, sometimes the model is not able to remove complex reflections of the object in the environment. These two failure cases are illustrated in Fig. 1. On the top row, the shadow underneath the object to remove is still visible in the output image. On the bottom row, the model successfully removed the person and associated shadow but failed to remove the reflection on the glass door.

D.2. Image relighting

While the proposed method is able to handle most cases, we noticed that it can sometimes fail to remove existing reflections on the foreground image, induce a color shift or add a *plastic* effect to the output image due to the use of synthetic data for training. We believe that these three failure cases can be addressed with a more careful training data curation and through more realistic renderings of the synthetic data.

E. Memory footprint and inference time

Our intuition to use a latent model is motivated by the key observations made in [8] where the authors scale image generation from diffusion models. Nevertheless, we quantitatively report in Tab. 4 the memory/latency comparison between a pixel model and a latent model both for training and inference. Note that the VAE compresses the source image

Method	NYUv2			KITTI			ETH3D			ScanNet			DIODE			Avg
	AbsRel↓	$\delta 1$ ↑	$\delta 2$ ↑	AbsRel↓	$\delta 1$ ↑	$\delta 2$ ↑	AbsRel↓	$\delta 1$ ↑	$\delta 2$ ↑	AbsRel↓	$\delta 1$ ↑	$\delta 2$ ↑	AbsRel↓	$\delta 1$ ↑	$\delta 2$ ↑	Rank
DiverseDepth	11.7	87.5	-	19.0	70.4	-	22.8	69.4	-	10.9	88.2	-	37.6	63.1	-	16.6
MiDaS	11.1	88.5	-	23.6	63.0	-	18.4	75.2	-	12.1	84.6	-	33.2	71.5	-	16.1
LeRes	9.0	91.6	-	14.9	78.4	-	17.1	77.7	-	9.1	91.7	-	27.1	76.6	-	13.2
Omnidata	7.4	94.5	-	14.9	83.5	-	16.6	77.8	-	7.5	93.6	-	33.9	74.2	-	13.2
DPT	9.8	90.3	-	10.0	90.1	-	7.8	94.6	-	8.2	93.4	-	**18.2**	75.8	-	10.8
HDN	6.9	94.8	-	11.5	86.7	-	12.1	83.3	-	8.0	93.9	-	24.6	78.0	-	10.2
DepthFM	6.0	95.5	-	9.1	90.2	-	6.5	95.4	-	6.6	94.9	-	22.4	**78.5**	-	7.2
GenPercept	5.6	96.0	99.2	13.0	84.2	97.2	7.0	95.6	98.8	6.2	96.1	99.1	35.7	75.6	86.6	8.3
Diff.-E2E-FT	5.4	96.5	99.1	9.6	92.1	98.0	6.4	95.9	98.7	5.8	96.5	98.8	30.3	77.6	87.9	5.6
DepthAnything V2	4.5	97.9	99.3	**7.4**	94.6	98.6	13.1	86.5	97.5	**4.2**	97.8	99.3	26.5	73.4	87.1	5.4
DepthAnything	**4.3**	**98.1**	**99.6**	7.6	**94.7**	**99.2**	12.7	88.2	98.3	4.3	**98.1**	**99.6**	26.0	75.9	87.5	4.1
GeoWizard	5.6	96.3	99.1	14.4	82.0	96.6	6.6	95.8	98.4	6.4	95.0	98.4	33.5	72.3	86.5	9.6
Marigold (LCM)	6.1	95.8	99.0	9.8	91.8	98.7	6.8	95.6	99.0	6.9	94.6	98.6	30.7	77.5	**89.3**	7.7
Marigold	5.5	96.4	99.1	9.9	91.6	98.7	6.5	95.9	99.0	6.4	95.2	98.8	30.8	77.3	88.7	6.4
Lotus-D	5.1	97.2	99.2	8.1	93.1	98.7	6.1	**97.0**	99.1	5.5	96.5	99.0	22.8	73.8	86.2	4.0
Lotus-G	5.4	96.8	99.2	8.5	92.2	98.4	**5.9**	**97.0**	99.2	5.9	95.7	98.8	22.9	72.9	86.0	5.3
Ours	5.6	97.2	99.2	9.4	93.0	98.9	6.3	96.5	**99.3**	5.7	97.0	99.2	30.3	77.5	**89.3**	**3.7**

Table 3. Metrics for depth estimation task. Our method uses a single NFE. Competitors results are taken from [4]. Best results are highlighted in bold, second best are underlined.

Figure 2. Failure cases for image relighting. On the left, the model is not able to remove the reflection in the subject glasses. On the right, the model changes the color of the person's jacket and create a *plastic* effect on the face.

by a factor of 8 and is frozen during training drastically reducing the memory footprint of the model as shown in the table.

Mode (Resolution)	Metric	Pixel Model	Latent Model	Gain
Inference (256 / 1024)	Latency (s) Peak Memory (Gb)	0.19 / 20.11 5.63 / 15.71	0.14 / 0.27 5.29 / 7.70	26.3% / 98.7% 6.0% / 51.0%
Training (256 / 1024)	Latency (s) Peak Memory (Gb)	0.71 / - 43.19 / OoM	0.43 / 0.58 24.85/ 25.35	39.4% / - 41.3% / -

Table 4. Training and inference memory usage and per-iteration latency for a *pixel* and a *latent* bridge model. The metrics are averaged over 10 images using a batch size of 1 with AdamW for training and 1 NFE for inference on a single H100 80Gb GPU.

F. Additional samples

Finally, we provide additional samples for object-removal in Fig. 3 and for image relighting in Figs. 4 to 8. For object-removal, our model remains the only one capable of removing the target object as well as the associated shadows. For image relighting, the proposed approach can create strong illumination effects on the foreground object and can handle complex lighting conditions. To further stress the method's versatility, we also consider an image restoration task and provide qualitative samples in Figs. 9 and 10. For this task, π_0 corresponds to the distribution of the latents of the degraded images while π_1 is the distribution of the latents of the clean images. We artificially create degraded images using the method proposed in [15]. In line with the performance observed for the tasks considered in the paper, the proposed method is able to create realistic outputs from degraded images.

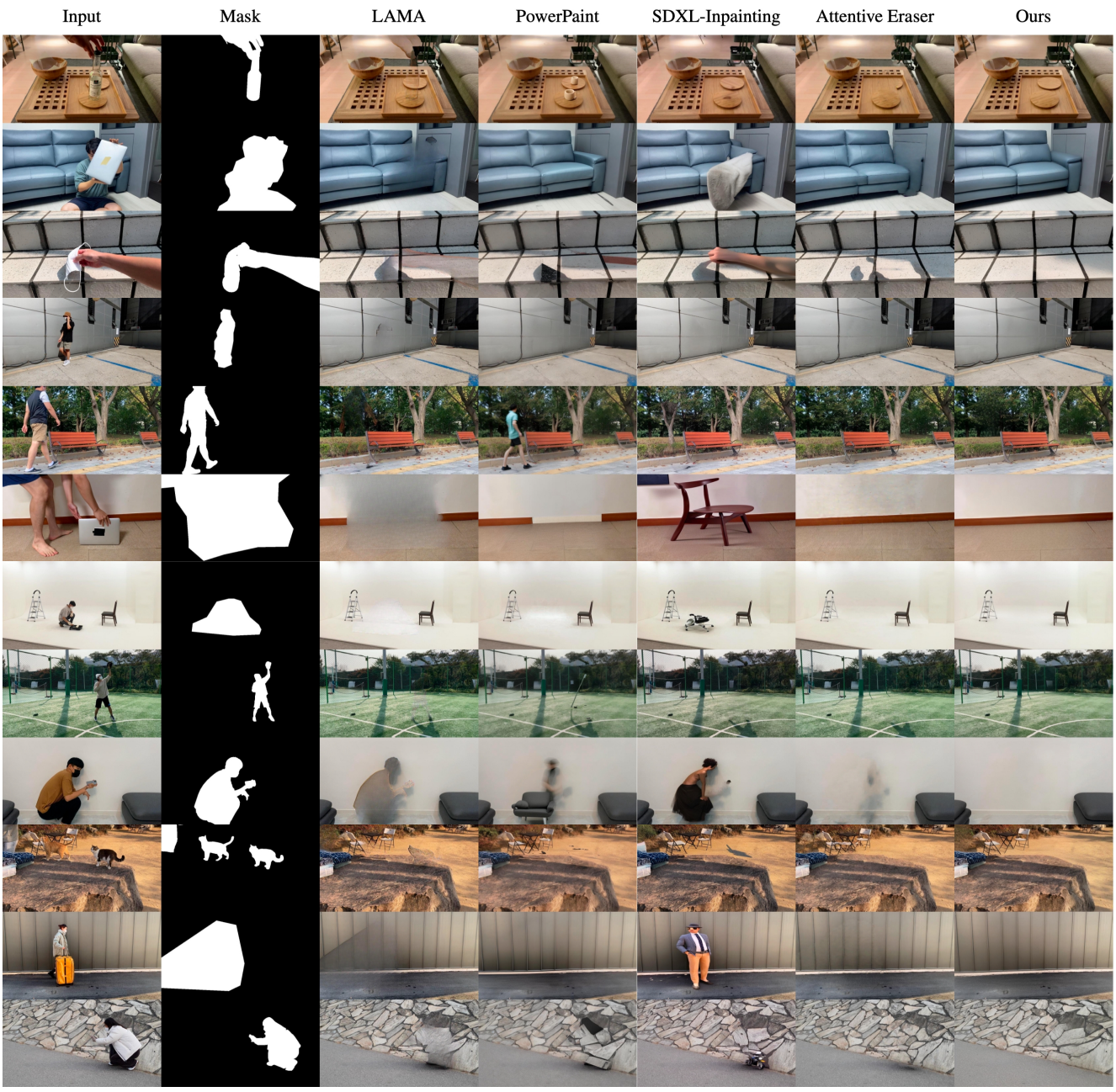

Figure 3. Qualitative results for object-removal on RORD validation dataset [9]. Best viewed zoomed in. Our model uses a single NFE and is able to successfully remove not only the object but also its shadow.

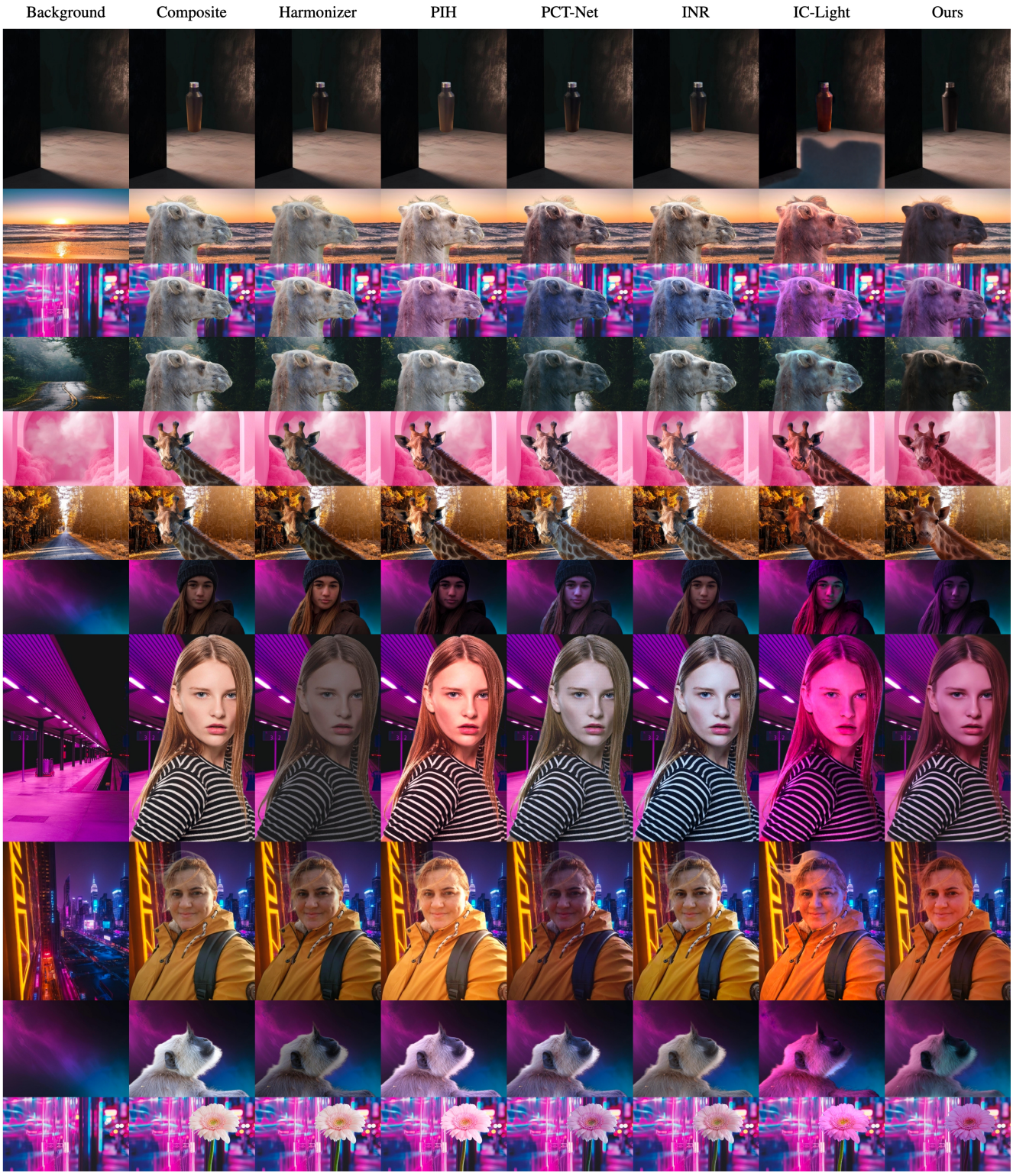

Figure 4. Qualitative results for object relighting. The model is able to relight the object according to the provided background and also remove existing shadows and reflections.

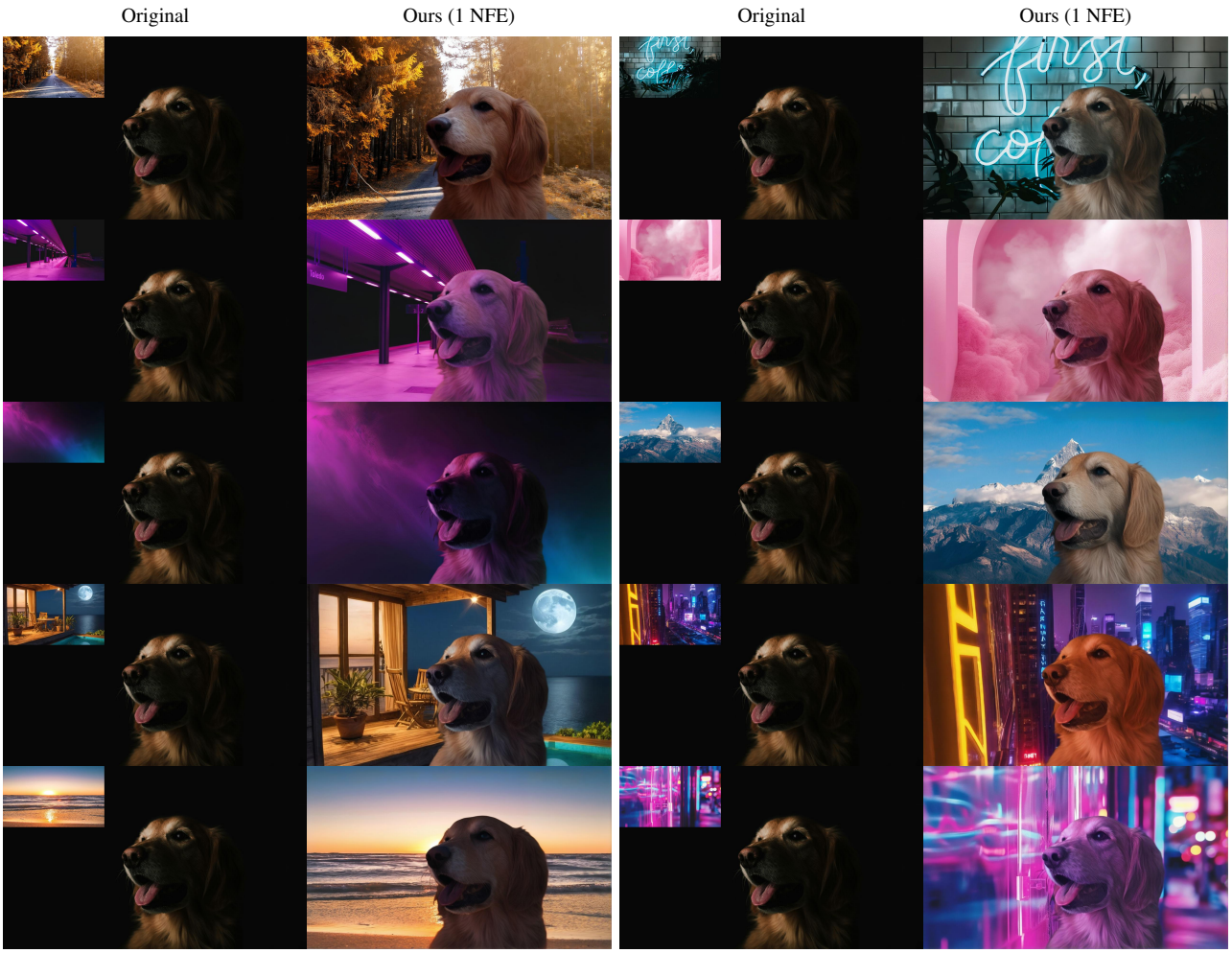

Figure 5. Qualitative results for object relighting. The model is able to relight the object according to the provided background and also remove existing shadows and reflections.

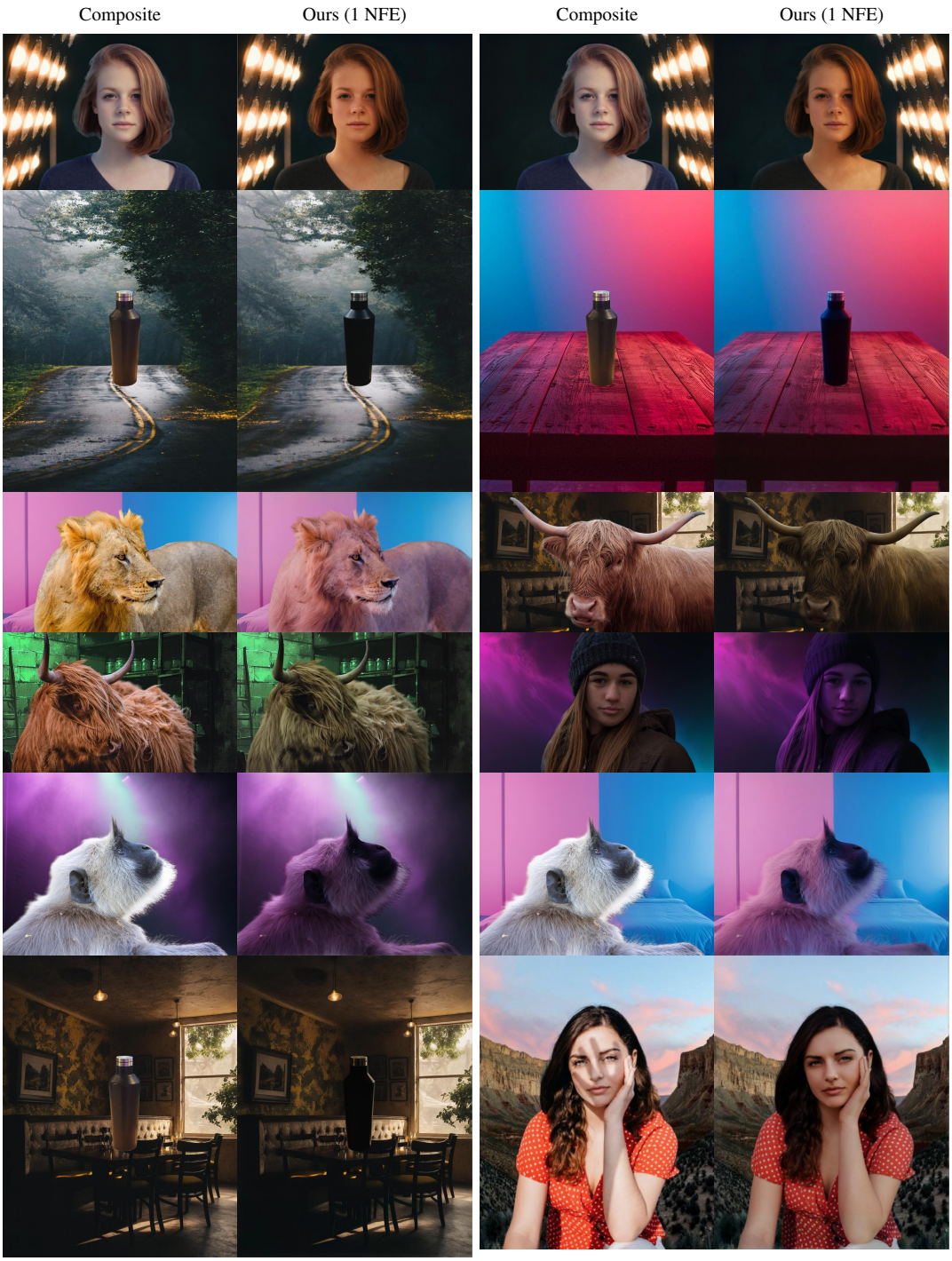

Figure 6. Qualitative results for object relighting. The model is able to relight the object according to the provided background and also remove existing shadows and reflections.

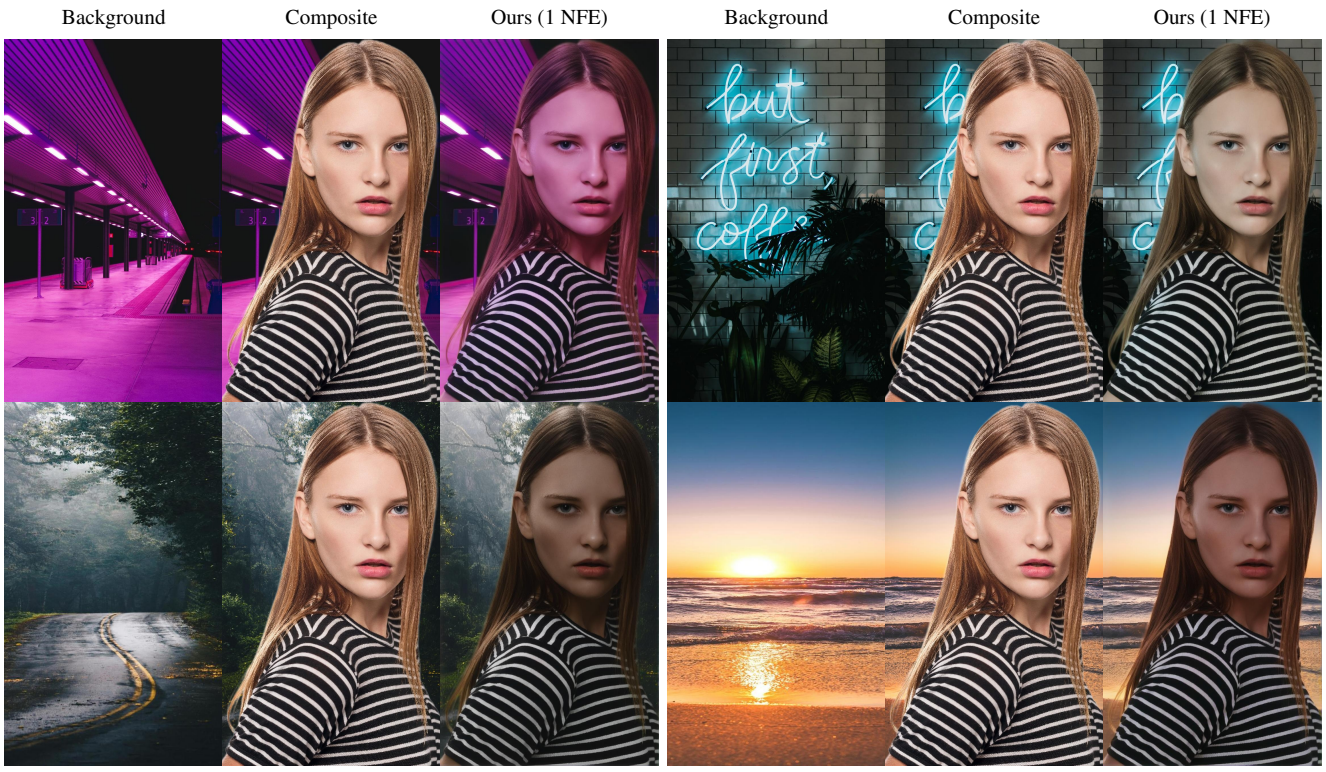

Figure 7. Qualitative results for object relighting. The model is able to relight the object according to the provided background and also remove existing shadows and reflections.

Figure 8. Qualitative results for controllable image relighting.

Figure 9. Qualitative results for object image restoration.

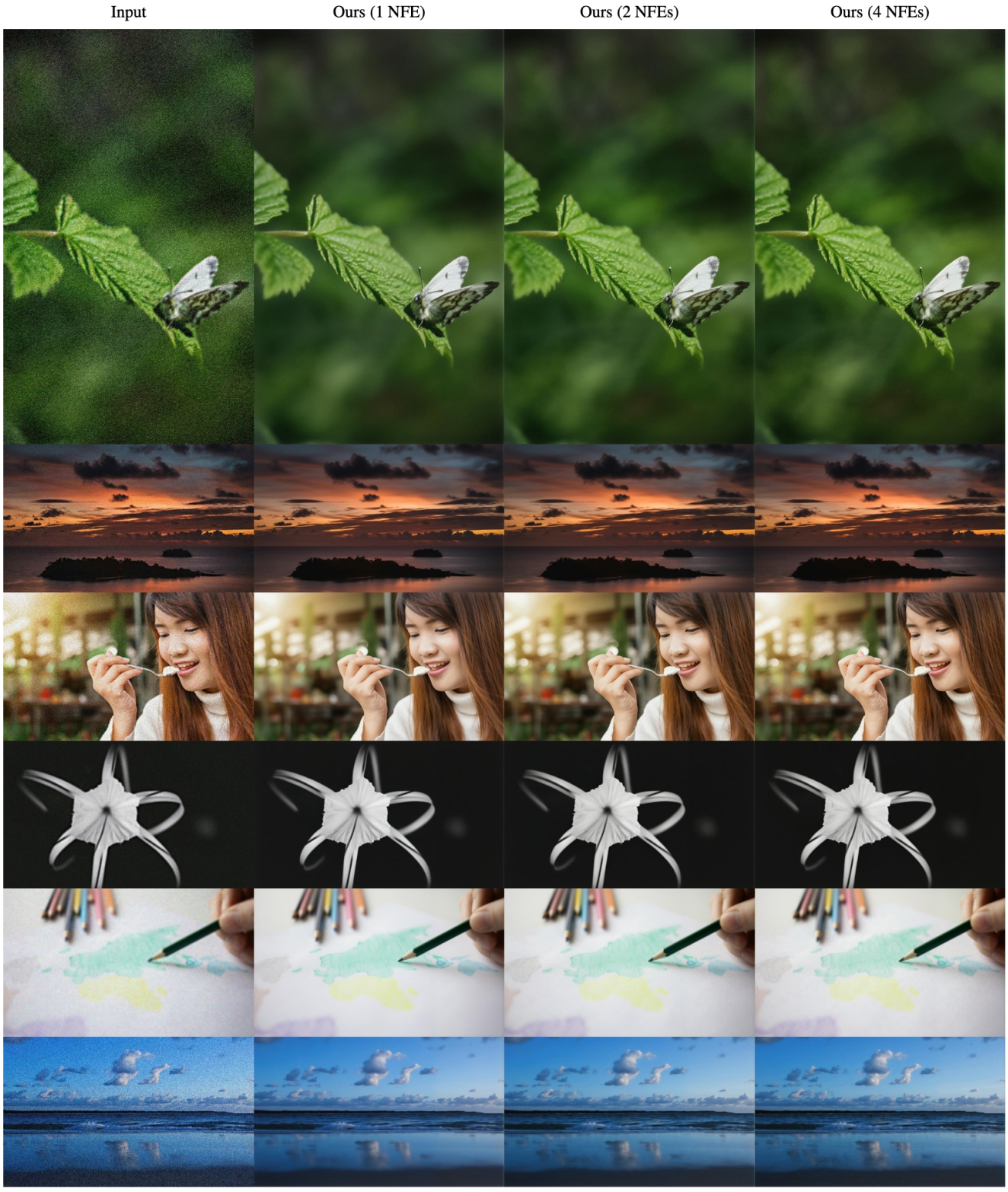

Figure 10. Qualitative results for object image restoration.

References

[1] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv e-prints*, pages arXiv–2001, 2020. 1

[2] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 2

[3] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2

[4] Jing He, Haodong Li, Wei Yin, Yixun Liang, Leheng Li, Kaiqiang Zhou, Hongbo Zhang, Bingbing Liu, and Ying-Cong Chen. Lotus: Diffusion-based visual foundation model for high-quality dense prediction. *arXiv preprint arXiv:2409.18124*, 2024. 3

[5] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 1

[6] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2023. 1

[7] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10912–10922, 2021. 1

[8] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

[9] Min-Cheol Sagong, Yoon-Jae Yeo, Seung-Won Jung, and Sung-Jea Ko. Rord: A real-world object removal dataset. In *BMVC*, page 542, 2022. 4

[10] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3260–3269, 2017. 2

[11] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, pages 746–760. Springer, 2012. 2

[12] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 1

[13] 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 *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022. 1

[14] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z Dai, Andrea F Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R Walter, et al. Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019. 2

[15] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1905–1914, 2021. 3