Supplementary Material for "Learning a Sketch Tensor Space for Image Inpainting of Man-made Scenes"

Chenjie Cao, Yanwei Fu
School of Data Science, Fudan University
{20110980001,yanweifu}@fudan.edu.cn

1. Network Architectures

The network architectures are illustrated as follows.

Partially Gated Convolution (GC) Block. Both encoder and decoder contain two groups of partially GC blocks for the input (3 blocks) and output (2 blocks) features, and the GC is consist of GateConv2D [10] $\rightarrow$ InstanceNorm $\rightarrow$ ReLU. We also add the spectral normalization [6] to the GC used in the generator.

Dilated Residual Block. The dilated residual blocks are the same as the ones used in EdgeConnect [7] with Conv2D $\rightarrow$ InstanceNorm $\rightarrow$ ReLU $\rightarrow$ Conv2D $\rightarrow$ InstanceNorm, where the first convolution is based on $\text{dilation} = 2$.

Efficient Attention. The input feature of the Efficient Attention is 256 channels, and we use the multi-head attention with $n_{head} = 4$. So, the dimension d' of each group is $256/4 = 64$.

2. Details of the Experiments

2.1. Preprocessing

For our method, the input images are resized to 256×256 at first. Then they are normalized to $[-1, 1]$, and values of the masked regions are set to 1. For the line detection, we set the thresholds of LSM-HAWP for unmasked and masked with (0.95, 0.925) in ShanghaiTech [3], and (0.85, 0.8) in Places2 [12] and York Urban [1].

2.2. Places2 Dataset Selection

Places2 dataset [12] is consisted of 10 million+ images from 365 various scenes. Since the proposed method is devoted to reconstructing the images with man-made structures, we need to select some representative scenes whose images contain enough structure information for man-made Places2 (P2M). Therefore, we leverage the retrained LSM-HAWP to predict the number of extracted line segmentations with scores larger than 0.925 in the validation of Places2. The scenes with top20 average line segment numbers are shown in Fig. 1. For the generalization, we randomly select 10 of them which is consist of 'balcony-exterior', 'computer-room', 'embassy', 'galley', 'general-store-outdoor', 'home-office', 'kitchen', 'library-outdoor', 'parking-garage-outdoor', and 'shopfront'. And there are 47949 training images and 1000 test images at all, which

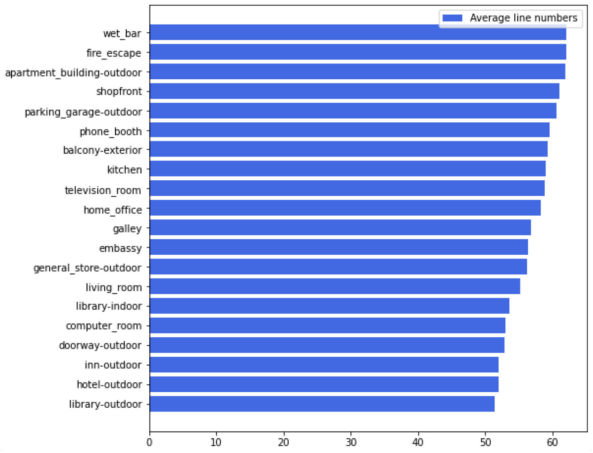


Figure 1. The bar chart of the scenes with top20 average line segment (confidence≥ 0.925) numbers of Places2.

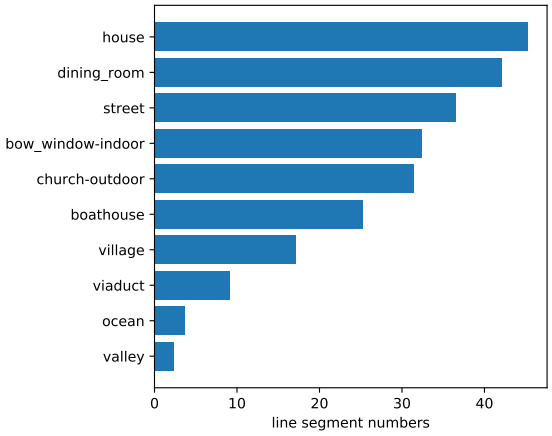


Figure 2. The bar chart of the line segment (confidence≥ 0.925) numbers of the comprehensive Places2 (P2C).

include buildings, indoor scenes, and outdoor scenes with various structures.

Besides, we also provide the results of comprehensive Places2 (P2C) in the main paper, which is consist of 10 randomly selected scenes: 'valley', 'church-outdoor', 'village', 'house', 'dining-room', 'street', 'ocean', 'bow-window-indoor', 'boathouse', and 'viaduct'. And the related line segments are shown in Fig. 2.

2.3. Implements of Compared Methods

All of the compared methods are based on the official implements with the new mask strategy mentioned in the paper. The hyper-parameter settings and learning rate settings refer to the official settings too. Besides, other settings are listed as follows. Note that code addresses listed in the footnote are the official implements of other compared methods.

Gated Convolution (GC) [10][1] GC is trained with 1500 epochs (about 468k steps) for ShanghaiTech, and 350 epochs (1049k steps) for Places2 with batch size 16.

Edge Connect (EC) [7][2] EC is trained with 400k (150k for the edge refinement network) steps for ShanghaiTech, and 1000k (300k steps for the edge refinement network) for Places2 with batch size 16.

Recurrent Feature Reasoning (RFR) [4][3] RFR is trained with 400k (150k for finetune) steps for ShanghaiTech, and 1200k (200k for finetune) for Places2 with batch size 6.

Mutual Encoder Decoder with Feature Equalizations (MED) [2][4] Note that the official implement of MED only supports training with batch=1 due to its special CSA attention design [5]. So, the training is very inefficient and it takes us more than three weeks to train the MED on Places2. However, MED still works badly on Places2, which dues to the instable training process with batch=1 in our opinions. As the result, MED is trained with 200 epochs (1000k steps) for ShanghaiTech, and 60 epochs (about 2877k steps) for Places2 with batch size 1.

3. Supplementary Experimental Results

3.1. Visualization of Gated Convolutions

We show the visualization results of the origin GC in Fig. 3, which are collected by averaging the $\sigma(\mathbf{G})$ outputs from different GateConvs. $\sigma(\mathbf{G})$ for the masked regions are active in Fig. 3(b) and Fig. 3(e), which are the encoder of the coarse network and the decoder of the refinement network.

[1] https://github.com/JiahuiYu/generative_inpainting

[2] https://github.com/knazeri/edge-connect

[3] https://github.com/jingyuanli001/RFR-Inpainting

[4] https://github.com/KumapowerLIU/Rethinking-Inpainting-MEDFE

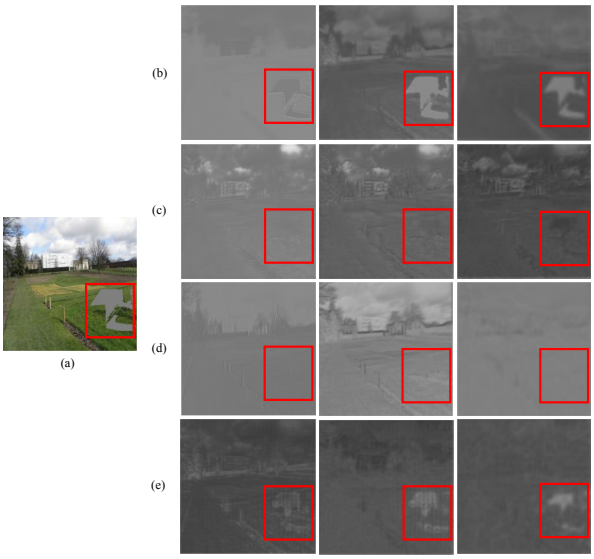


Figure 3. Visualization of sigmoid weight outputs $\sigma(\mathbf{G})$ in GC: (a) is the masked input, and (b), (c), (d), (e) are $\sigma(\mathbf{G})$ with different scales from the coarse network encoder, coarse network decoder, refinement network encoder, and refinement network decoder respectively.

Attention	Batchsize	Image/sec
CA	16	19.42
EA	16	36.37
CA	32	20.70
EA	32	40.81

Table 1. Comparison of the training speed of our model based on Contextual Attention (CA) [9] and Efficient Attention (EA) [8] with different batch sizes.

And they can also be seen as the encoder and decoder of the whole GC pipeline model.

3.2. Efficiency Comparisons of Efficient Attention

In this section, we pay attention to compare the efficiency of Contextual Attention (CA) [9] and Efficient Attention (EA) [8]. All comparisons are based on our structure refinement network, and only one attention layer is added to the middle of the residual blocks with the resolution 64×64. Then, we train the model with two attention strategies and batch sizes on the ShanghaiTech dataset for one epoch. The results are shown in Tab. 1. Since CA can only be implemented without parallel computing for the batch, we just compare the training speed with images per second between CA and EA. From Tab. 1, EA can be trained more efficiently than CA. Moreover, benefited from the parallelism, EA enjoys the speedup by enlarging the training batch size.

(a) input (b) PSS (edge) (c) PSS (edge+line)

Figure 4. Qualitative results w. and w./o. lines in ShanghaiTech.

	PSNR↑	SSIM↑	FID↓
PSS (edge)	26.78	0.875	21.40
PSS (edge+line)	**26.90**	**0.876**	**21.16**

Table 2. Quantitative results w. and w./o. lines in ShanghaiTech.

Figure 5. Qualitative results compared with ProFill in flickr.

3.3. Ablation Study with Only Edge and PSS

To further improve the effectiveness of lines extracted from the LSM-HAWP. Here we provide the qualitative and quantitative results in Fig. 4 and Tab. 2: only canny edges with multi-scale training of PSS (b), V.S. our full model (c). Some lines on the roof are not recovered in (b) of Fig. 4.

3.4. Comparisons with ProFill

ProFill [11] is one of SOTA inpainting methods, but no open source codes/models of ProFill are available now. To ensure the fairness, we compare qualitative results from flickr by the API of ProFill in Fig. 5, which are independent from the training set of both our method and ProFill.

3.5. Additional Visual Results

In this section we provide more experiment results for image inpainting and object removal on different datasets. Furthermore, we provided more comparisons in P2C compared with EC, which show the superior performance of the proposed sketch tensor space reconstructing.

3.6. Object Removal Video

We also provide a video that displays the object removal in ShanghaiTech, P2M, and P2C. The video is played in a triple-speed to compress the storage size. This video shows the good generalization and the fast interactive experience of the proposed method in the GPU environment.

References

[1] Patrick Denis, James H Elder, and Francisco J Estrada. Efficient edge-based methods for estimating manhattan frames in urban imagery. In *European conference on computer vision*, pages 197–210. Springer, 2008.

[2] Yibing Song Wei Huang Hongyu Liu, Bin Jiang and Chao Yang. Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In *Proceedings of the European Conference on Computer Vision*, 2020.

[3] Kun Huang, Yifan Wang, Zihan Zhou, Tianjiao Ding, Shenghua Gao, and Yi Ma. Learning to parse wireframes in images of man-made environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 626–635, 2018.

[4] Jingyuan Li, Ning Wang, Lefei Zhang, Bo Du, and Dacheng Tao. Recurrent feature reasoning for image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7760–7768, 2020.

[5] Hongyu Liu, Bin Jiang, Yi Xiao, and Chao Yang. Coherent semantic attention for image inpainting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4170–4179, 2019.

[6] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[7] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

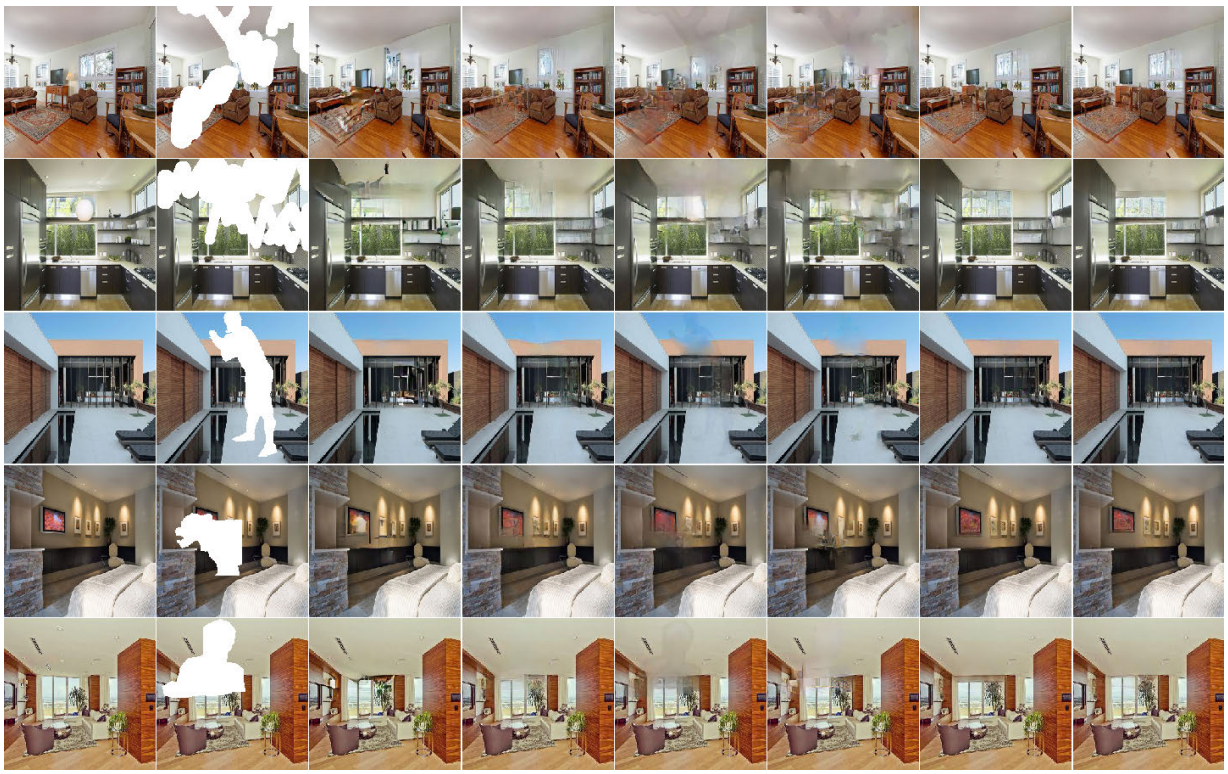
[8] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. *arXiv preprint arXiv:1812.01243*, 2018.

[9] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018.

[10] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4471–4480, 2019.

[11] Yu Zeng, Zhe Lin, Jimei Yang, Jianming Zhang, Eli Shechtman, and Huchuan Lu. High-resolution image inpainting with iterative confidence feedback and guided upsampling. *arXiv preprint arXiv:2005.11742*, 2020.

[12] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.



Ground Truth Input GC EC RFR MED Ours Ours*

Figure 6. Inpainting comparisons on ShanghaiTech, where * means that our method works in the object removal mode.

Ground Truth Input GC EC RFR MED Ours Ours*

Figure 7. Inpainting comparisons on P2M, where * means that our method works in the object removal mode.

Figure 8. Inpainting results in P2C compared with EC (part1). From left to right: origin image, input masked image, refined edges from EC, inpainted results of EC, our refined sketch tensors, our inpainted results (generated lines and edges are blue).

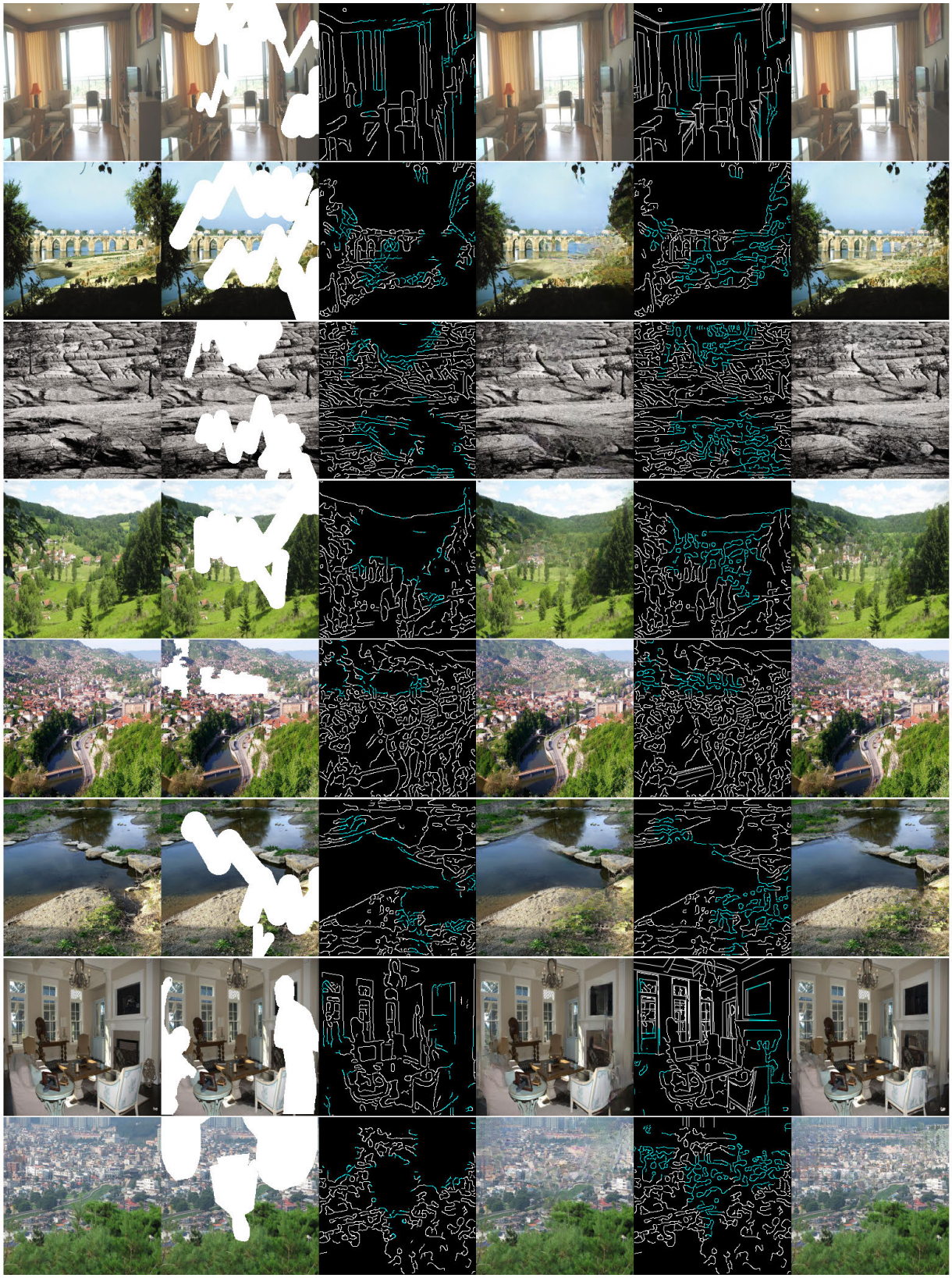


Figure 9. Inpainting results in P2C compared with EC (part2). From left to right: origin image, input masked image, refined edges from EC, inpainted results of EC, our refined sketch tensors, our inpainted results (generated lines and edges are blue).

Figure 10. Inpainting results of our method on ShanghaiTech. For each of the two columns, from left to right: origin image, input masked image, refined structures (generated lines and edges are blue), inpainted results.

Figure 11. Inpainting results of our method on P2M. For each of the two columns, from left to right: origin image, input masked image, refined structures (generated lines and edges are blue), inpainted results.

Figure 12. Inpainting results of our method on York Urban. For each of the two columns, from left to right: origin image, input masked image, refined structures (generated lines and edges are blue), inpainted result.

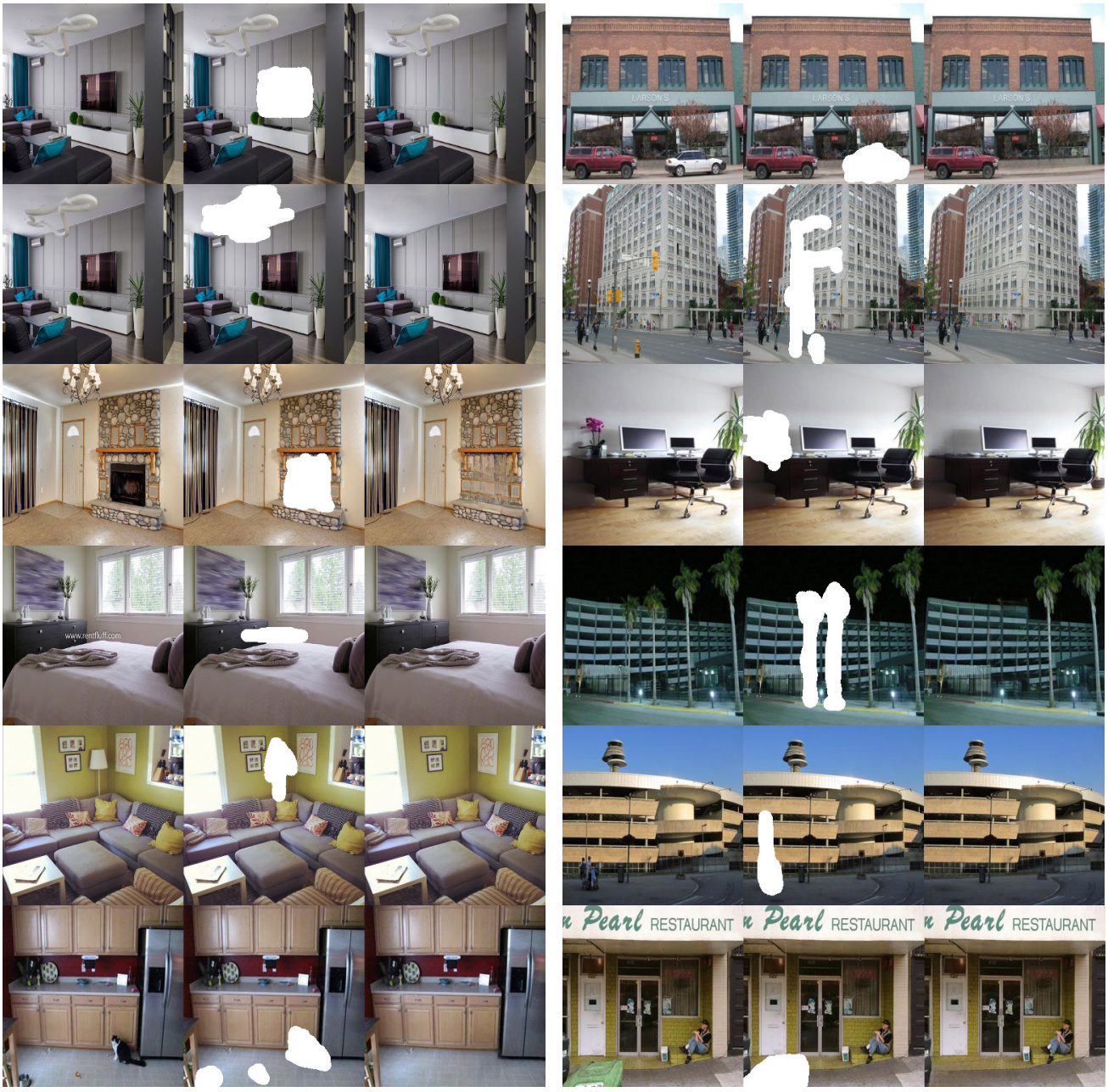


Figure 13. Object removal results from ShanghaiTech, Places2, and York Urban. For each of the two columns, from left to right: origin image, input masked image, generated image.