Supplementary Materials of Paper: Image Inpainting with Learnable Bidirectional Attention Maps

The following items are included in the supplementary materials.

- Visual comparison of several LBAM variants on Paris StreetView dataset.
- The architectures of our learnable bidirectional attention map and the discriminator.
- More comparison with state-of-the-art methods (*e.g.*, Partial Convolution [4]) on Paris StreetView [2] and Places [6] datasets .
- Object removal on real world images.

1. Visual comparison of several LBAM variants on Paris StreetView

We implement our bidirectional attention maps by employing an asymmetric Gaussian shaped form (Eqn. (9)) for activation the attention map and the modified activation function (Eqn. (8)) for updating the mask. In this material, we give visual comparison of several variants of our LBAM model, *i.e.*, (i) Ours(full): the full LBAM model, (ii) Ours(unlearned): the LBAM model where all the elements in mask convolution filters are set as $\frac{1}{16}$ because the filter size is 4×4, and we adopt the activation functions defined in Eqn. (4) and Eqn. (5), (iii) Ours(forward): the LBAM model without reverse attention map, (iv) Ours(w/o $\mathcal{L}_{adv}$): the LBAM model without (w/o) adversarial loss, (v) Ours(Sigmoid/LReLU/ReLU/3×3): the LBAM model using Sigmoid/LeakyReLU/ReLU as activation functions or 3×3 filter for mask updating.

Figure 1 shows qualitatively comparison over variants (i) to (iv). Ours (forward) model benefits from learnable attention map and helps reduce reduce the artifacts and noise of unlearned one, see Figure 1(a) and (b). But its decoder hallucinates both holes and known regions and produces some blurry effects compared to our full model with learnable reverse attention map Figure 1(d).

The qualitative comparison in ablation studies with the effect of GAN loss is shown in Figure 1(c) and (d). The inpainted results of our LBAM model without adversarial loss (Figure 1(c)), are much better than the unlearned model Figure 1(a), and somehow clearer in producing details than ours without reverse attention map which applied GAN loss. Our LBAM full model (Figure 1(d)) benefits from GAN loss, is superior in giving fine-detailed structures and capturing global semantics.

The visual comparison of different activation functions or 3×3 filter for mask updating are shown in Figure 2.

Failure cases. Figure 3 shows some failure cases of our LBAM model. Our model struggles to recover the high-frequency details while the damaged areas are too large and the background objects are too complex. In some cases, the mask covers a large portion of a specific object, like a car, it is still difficult for our LBAM model to recover the original shape.

2. Model Architectures

2.1. Architecture of Our Learnable Bidirectional Attention Map

The learnable bidirectional attention model takes the damaged image, the mask M^{in} and the reverse mask $1 - M^{in}$ as input. We adopt the basic U-Net structure with 14 layers, and both encoder and decoder consists of 7 layers. The features are normalized by the learnable bidirectional attention maps through element-wise product. We use convolution filters of size 4×4, stride $= 2$, padding $= 1$ for all layers including the bidirectional attention maps.

The forward attention map takes the mask M^{in} as input, it contains 7 layers, and the reverse attention map takes the reverse mask $1 - M^{in}$ as input, which consists of 6 layers. We adopt an asymmetric Gaussian-shaped form as activation function ($g_A(\cdot)$ of Eqn. (9)) for activating the attention map and a modified ReLU based activating function ($g_M(\cdot)$ of Eqn. (8)) for updating mask maps. In consideration of the skip connection of the U-Net structure, the symmetric forward and reverse attention maps are concatenated for normalizing the connected features of the corresponding layer in the decoder, under Eqn. (12). Besides, batch normalization and Leaky ReLU non-linearity are used to the features after attention re-normalization. The last layer of our LBAM model are directly de-convoluted with filters of size 4×4, stride $= 2$, padding $= 1$, followed by a tanh non-

Input (a) Ours(unlearned) (b) Ours(forward) (c) Ours(w/o $\mathcal{L}_{adv}$) (d) Ours(full)

Figure 1. Visual comparison of variants (i) to (iii) of our LBAM model. From left to right are: Input, (a) Ours with unlearned model, (b) Ours without reverse attention map, (c) Our without (w/o) adversarial loss, (d) our full LBAM model. All images are scaled to 256×256.

linear activation. More details about our model is given in Table 2. Note that each activation function $g_A(\cdot)$ and mask updating term $g_M(\cdot)$ are unique for each layer, and they do not share parameters among layers.

2.2. Architecture of the Discriminator

The discriminator is trained to produce adversarial loss for minimizing the distance between the generated images and the real data distributions. In our work, we use a two-column discriminator with one column takes the remained area of inpainted result or a ground-truth image, and another column takes the missing holes of inpainted result or a ground-truth image as input. The two-column discriminator consists of 7 layers, the two parallel features are emerged after 6_{th} layer at the resolution of 4×4. We specifically use convolution layer with filters size of 4×4, stride $= 2$ and padding $= 1$, except the last layer with stride $= 0$. We use sigmoid non-linear activation function at last layer, while the leaky ReLU with slope of 0.2 for other layers. Table 1 provides a more details of the discriminator.

3. More Comparisons on Paris StreetView and Places

More comparisons with PatchMatch (PM) [1], Global&Local (GL) [3], Context Attention (CA) [5], and Partial Convolution (Pconv) [4] are also conducted. Figure 4, 5 and 6 show the qualitative comparison on Paris StreetView dataset and Places dataset. For Paris StreetView [2] dataset, we use its original splits, $14,900$ images for training, and 100 images for testing.

For Places [6] dataset, 10 categories from the total 365 categories are choosed for training our LBAM model, they are: *apartment_building_outdoor*, *beach*, *house*, *ocean*, *sky*, *throne_room*, *tower*, *tundra*, *valley* and *wheat_field*. We gather all 5000 images of each category to form our training set of $50,000$ images. The validation set from each category of $1,000$ images into two equal non-overlapped sets of 500 images respectively for validation and testing. It can be seen that our model performs better in producing both global consistency and fine-detailed structures.

4. Object removal on real world images.

Finally, we apply our model trained on Places dataset for object removal on real world images. As shown in Figure 7, although these images contain different objects, background, context and shapes, even some of them have large portion masked regions, our model can handle them well, demonstrating the practicability and generalization ability of our LBAM model.

Input (a) (b) (c) (d) (e) Ours

Figure 2. Visual comparison of different activation functions or 3×3 filters on the bidirectional attention maps. From left to right are: Input, (a) Sigmoid as activation function, (b) Leaky ReLU with slope of 0.2 as activation function, (c) ReLU, (e) 3×3 filter for mask updating, and (e) our full LBAM model. All images are scaled to 256×256.

Input Ours Ground Truth Input Ours Ground Truth

Figure 3. Failure cases of our LBAM model. Each group is ordered as input image, our result and ground truth. All images are scaled to 256×256.

References

[1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (TOG)*, pages 24:1–24:11, 2009. 2, 5, 6, 7

[2] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei A Efros. What makes paris look like paris? *Communications of the ACM*, pages 103–110, 2015. 1, 2

[3] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, pages 107:1–107:14, 2017. 2, 5, 6, 7

[4] Guilin Liu, Fitsum A. Reda, Kevin Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *The European Conference on Computer Vision (ECCV)*, volume 11215, pages 89–105, 2018. 1, 2, 5, 6, 7

[5] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative image inpainting with contextual attention. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5505–5514, 2018. 2, 5, 6, 7

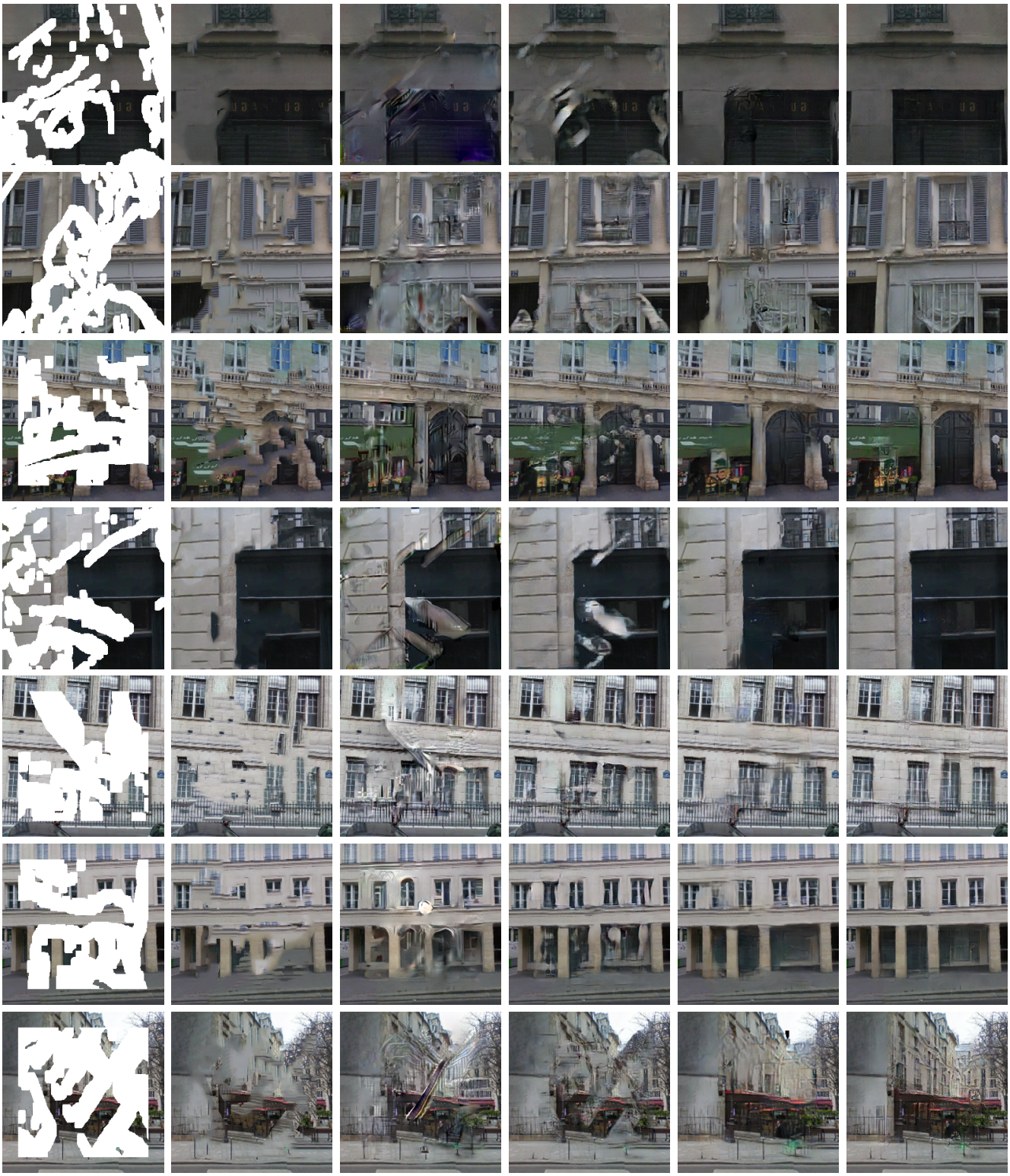
[6] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 1452–1464, 2017. 1, 2

Table 1. The architecture of the discriminator. BN represents BatchNorm, LReLU denotes leaky ReLU with slope of 0.2, and M represents mask with zeros denote the missing pixels and ones denote the remained pixels.

Input:	Image ($256 \times 256 \times 3$) $*M$	**Input**:	Image ($256 \times 256 \times 3$) $*(1 - M)$
[Layer 1-1]	Conv.$(4, 4, 64)$, stride = 2; LReLU;	[Layer 1-2]	Conv.$(4, 4, 64)$, stride = 2; LReLU;
[Layer 2-1]	Conv.$(4, 4, 128)$, stride = 2; BN; LReLU;	[Layer 2-2]	Conv.$(4, 4, 128)$, stride = 2; BN; LReLU;
[Layer 3-1]	Conv.$(4, 4, 256)$, stride = 2; BN; LReLU;	[Layer 3-2]	Conv.$(4, 4, 256)$, stride = 2; BN; LReLU;
[Layer 4-1]	Conv.$(4, 4, 512)$, stride = 2; BN; LReLU;	[Layer 4-2]	Conv.$(4, 4, 512)$, stride = 2; BN; LReLU;
[Layer 5-1]	Conv.$(4, 4, 512)$, stride = 2; BN; LReLU;	[Layer 5-2]	Conv.$(4, 4, 512)$, stride = 2; BN; LReLU;
[Layer 6-1]	Conv.$(4, 4, 512)$, stride = 2; BN; LReLU;	[Layer 6-2]	Conv.$(4, 4, 512)$, stride = 2; BN; LReLU;
Concatenate(Layer 6-1, Layer 6-2);			
[Layer 7]	Conv.$(4, 4, 1)$, stride = 0; Sigmoid;		
Output:	Real or Fake ($1 \times 1 \times 1$)		

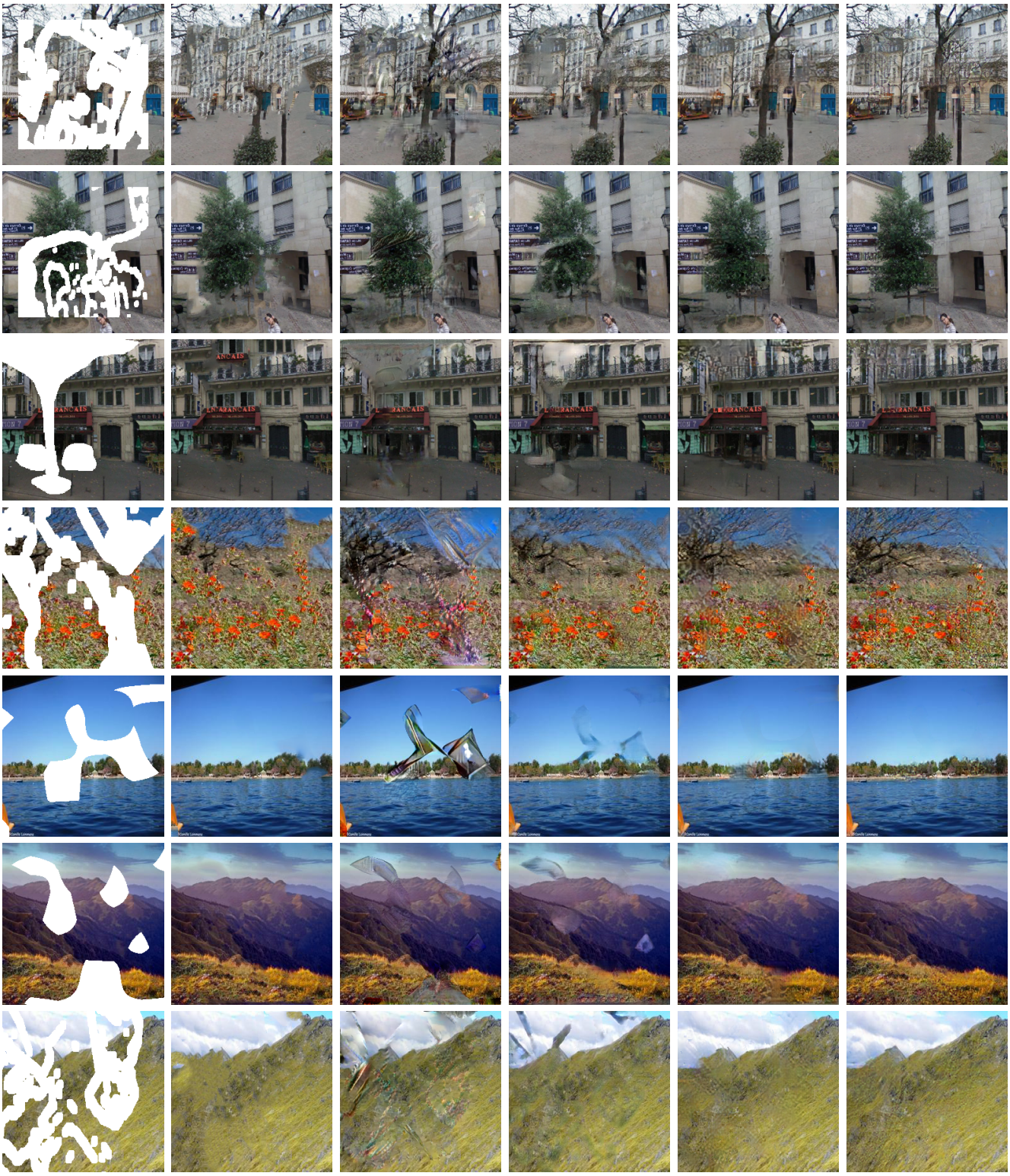
Table 2. The architecture of our LBAM model. Ewp() means element-wise product, Cat() represents feature concatenation operation, $g_A(\cdot)$ denotes asymmetric Gaussian-shaped form activation function of Eqn. (9), and $g_M(\cdot)$ denotes mask updating function of Eqn. (8), BN represents BatchNorm, LReLU denotes leaky ReLU with slope of 0.2, and M^{in} represents mask with zeros indicating the missing pixels and ones indicating the remained pixels. Note that $g_A(\cdot)$ and $g_M(\cdot)$ are unique among layers and do not share its parameters.

Our Modified U-Net		Learnable Bidirectional Attention Maps	
Input:	Image ($256 \times 256 \times 3$)	**Input**:	M^{in} ($256 \times 256 \times 3$)
[Layer 1-1]	Conv.$(4, 4, 64)$, stride = 2;	[Layer 1-2]	Conv.$(4, 4, 64)$, stride = 2;
Ewp(Layer 1-1, g_A(Layer 1-2)); LReLU;			
[Layer 2-1]	Conv.$(4, 4, 128)$, stride = 2;	[Layer 2-2]	$g_M(\cdot)$; Conv.$(4, 4, 128)$, stride = 2;
Ewp(Layer 2-1, g_A(Layer 2-2)); BN; LReLU;			
[Layer 3-1]	Conv.$(4, 4, 256)$, stride = 2;	[Layer 3-2]	$g_M(\cdot)$; Conv.$(4, 4, 256)$, stride = 2;
Ewp(Layer 3-1, g_A(Layer 3-2)); BN; LReLU;			
[Layer 4-1]	Conv.$(4, 4, 512)$, stride = 2;	[Layer 4-2]	$g_M(\cdot)$; Conv.$(4, 4, 512)$, stride = 2;
Ewp(Layer 4-1, g_A(Layer 4-2)); BN; LReLU;			
[Layer 5-1]	Conv.$(4, 4, 512)$, stride = 2;	[Layer 5-2]	$g_M(\cdot)$; Conv.$(4, 4, 512)$, stride = 2;
Ewp(Layer 5-1, g_A(Layer 5-2)); BN; LReLU;			
[Layer 6-1]	Conv.$(4, 4, 512)$, stride = 2;	[Layer 6-2]	$g_M(\cdot)$; Conv.$(4, 4, 512)$, stride = 2;
Ewp(Layer 6-1, g_A(Layer 6-2)); BN; LReLU;			
[Layer 7-1]	Conv.$(4, 4, 512)$, stride = 2;	[Layer 7-2]	$g_M(\cdot)$; Conv.$(4, 4, 512)$, stride = 2;
Ewp(Layer 7-1, g_A(Layer 7-2)); BN; LReLU;			
[Layer 8-1]	DeConv.$(4, 4, 512)$, stride = 2;	[Layer 6-3]	$g_M(\cdot)$; Conv.$(4, 4, 512)$, stride = 2;
Ewp(Cat(Layer 8-1, Layer 6-1), Cat(g_A(Layer 6-3), g_A(Layer 6-2)));BN; LReLU;			
[Layer 9-1]	DeConv.$(4, 4, 512)$, stride = 2;	[Layer 5-3]	$g_M(\cdot)$; Conv.$(4, 4, 512)$, stride = 2;
Ewp(Cat(Layer 9-1, Layer 5-1), Cat(g_A(Layer 5-3), g_A(Layer 5-2)));BN; LReLU;			
[Layer 10-1]	DeConv.$(4, 4, 512)$, stride = 2;	[Layer 4-3]	$g_M(\cdot)$; Conv.$(4, 4, 512)$, stride = 2;
Ewp(Cat(Layer 10-1, Layer 4-1), Cat(g_A(Layer 4-3), g_A(Layer 4-2)));BN; LReLU;			
[Layer 11-1]	DeConv.$(4, 4, 256)$, stride = 2;	[Layer 3-3]	$g_M(\cdot)$; Conv.$(4, 4, 256)$, stride = 2;
Ewp(Cat(Layer 11-1, Layer 3-1), Cat(g_A(Layer 3-3), g_A(Layer 3-2)));BN; LReLU;			
[Layer 12-1]	DeConv.$(4, 4, 128)$, stride = 2;	[Layer 2-3]	$g_M(\cdot)$; Conv.$(4, 4, 128)$, stride = 2;
Ewp(Cat(Layer 12-1, Layer 2-1), Cat(g_A(Layer 2-3), g_A(Layer 2-2)));BN; LReLU;			
[Layer 13-1]	DeConv.$(4, 4, 64)$, stride = 2;	[Layer 1-3]	Conv.$(4, 4, 64)$, stride = 2;
Ewp(Cat(Layer 13-1, Layer 1-1), Cat(g_A(Layer 1-3), g_A(Layer 1-2)));BN; LReLU;			
[Layer 14-1]	DeConv.$(4, 4, 3)$, stride = 2; tanh;	**Input**:	$1 - M^{in}$ ($256 \times 256 \times 3$)
Output:	Final result ($256 \times 256 \times 3$)	Reverse Attention Maps	



Input PM [1] GL [3] CA [5] PConv [4] Ours

Figure 4. Qualitative comparison on Paris StreetView dataset. Comparison with PatchMatch (PM) [1], Global&Local (GL) [3], Context Attention (CA) [5], and Partial Convolution (PConv) [4]. All images are scaled to 256×256.



Input PM [1] GL [3] CA [5] PConv [4] Ours

Figure 5. Qualitative comparison on Paris StreetView dataset. Comparison with PatchMatch (PM) [1], Global&Local (GL)GL [3], Context Attention (CA) [5], and Partial Convolution (PConv) [4]. First three rows are from Paris StreetView dataset and the last four rows are from Places dataset. All images are scaled to 256×256.

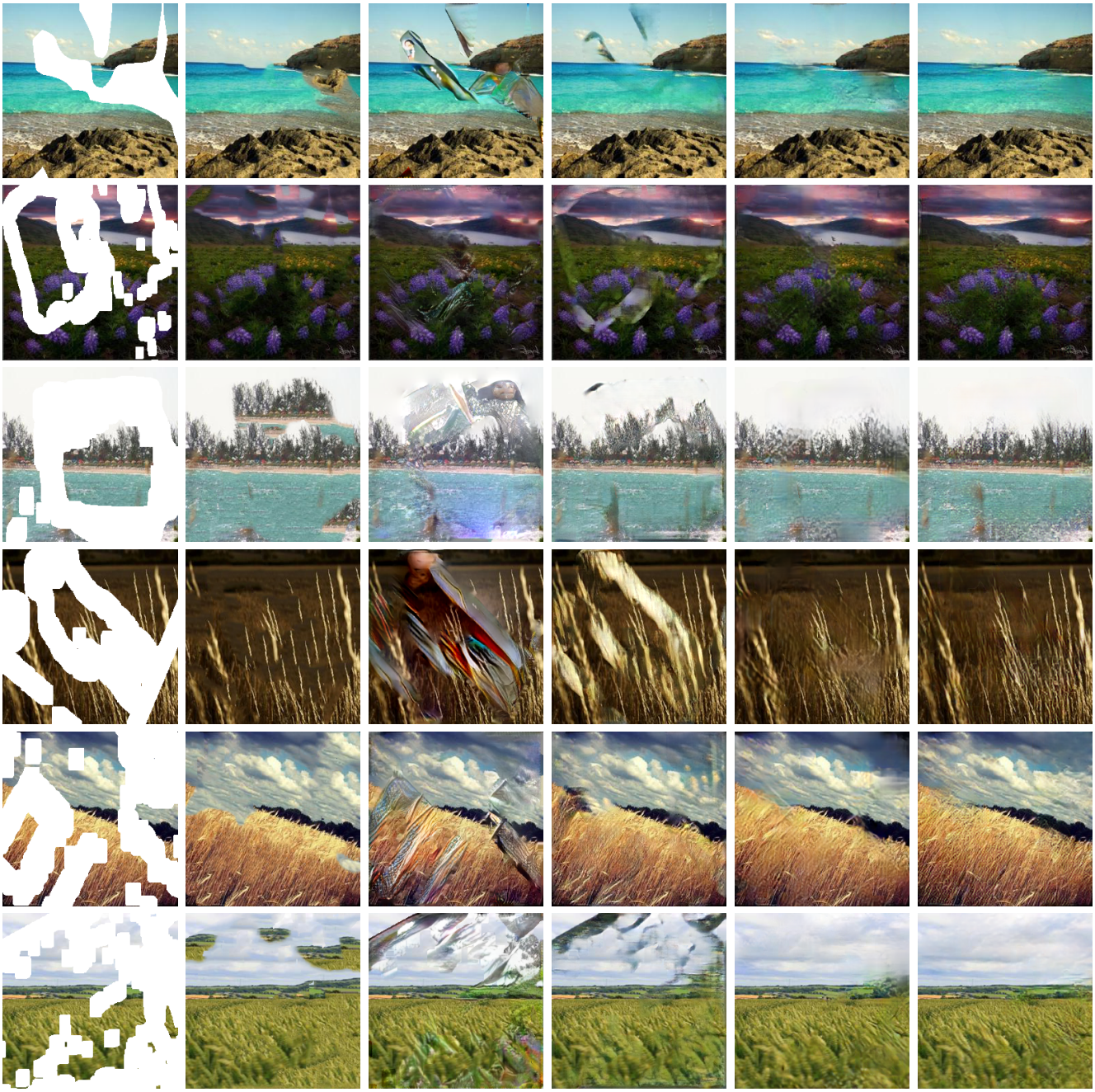


Figure 6. Qualitative comparison on Places dataset. Comparison with PatchMatch (PM) [1], Global&Local (GL) [3], Context Attention (CA) [5], and Partial Convolution (PConv) [4]. All images are scaled to 256×256.

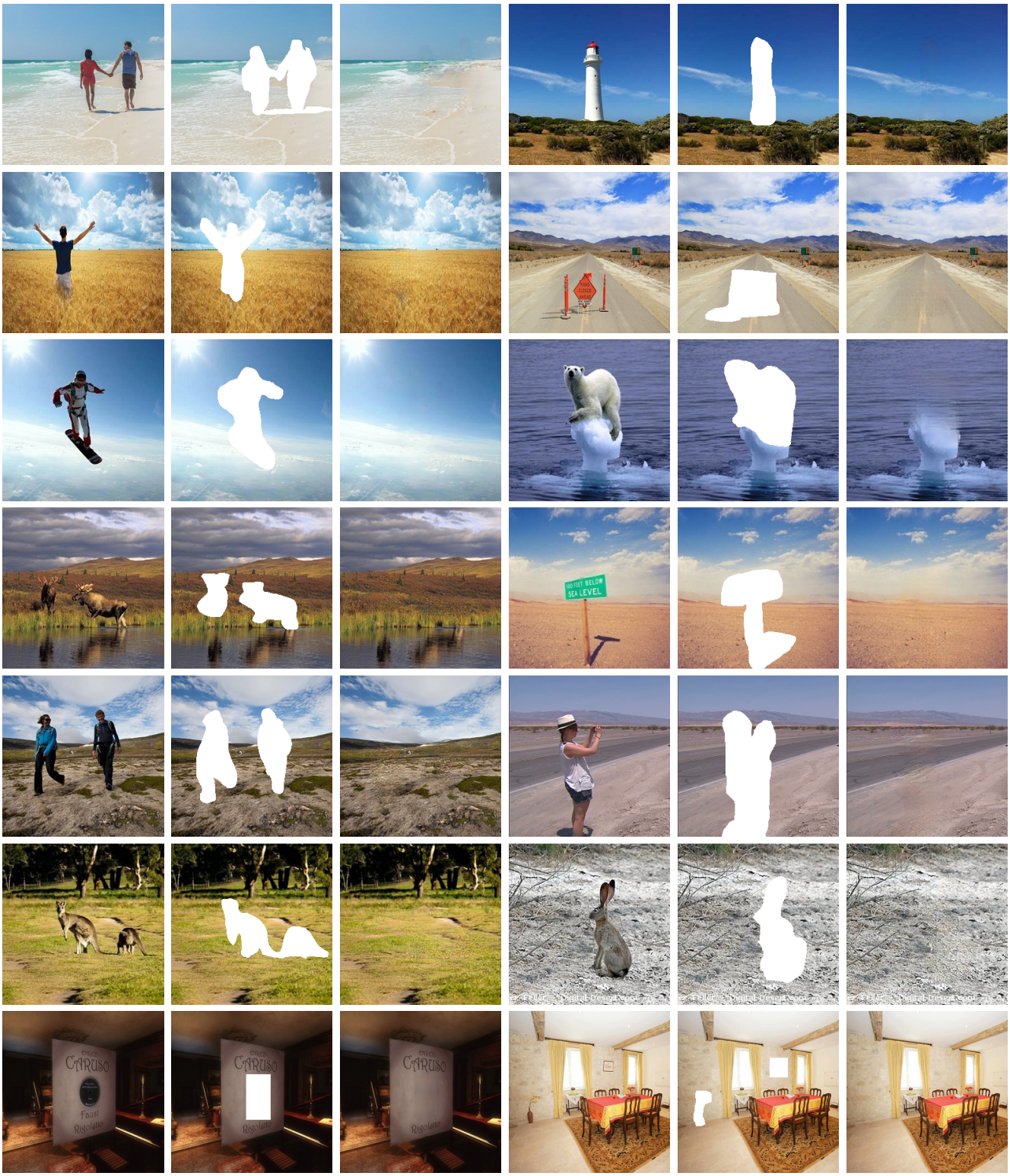


Figure 7. Results of our LBAM on object removal task of real world images. All images are scaled to 256×256.