The Supplementary of 'Incremental Transformer Structure Enhanced Image Inpainting with Masking Positional Encoding'

Qiaole Dong*, Chenjie Cao*, Yanwei Fu†
School of Data Science, Fudan University
{18307130096,20110980001,yanweifu}@fudan.edu.cn

1. Broader Impacts

All generated results of both the main paper and the supplementary are based on learned statistics of the training dataset. Therefore, the results only reflect biases in those data without our subjective opinion. This work is only researched for the algorithmic discussion, and related societal impacts should not be ignored by users.

2. Detailed Network Settings

We show some detailed network settings in Tab. 1. Besides, the transformer block and Fast Fourier Convolution (FFC) block [11] have been introduced in the main paper. The dilated resnet block is from the middle layer of [9] with dilate=2.

3. More Training Details

Training a model with dynamic resolutions of 256∼512 reduces the training speed with frequent GPU memory swaps. Therefore, we train the model with regular resolutions, *i.e.*, resizing images from 512 to 256 and then back to 512. For Indoor, there is one cycle for each epoch. For Places2, there are 64 cycles for each epoch. Such a local monotonic resizing makes the training smooth without missing diversity. And the dynamic resolution based training can effectively save the training cost compared with the training with a full 512 image size. Moreover, it benefits to learn relative position encoding for our proposed MPE as discussed in [13].

Our TSR can be trained in batch size 30 with 3 NVIDIA(R) Tesla(R) V100 16GB GPUs. 256×256 based FTR and SFE can be trained in batch size 30 with 3 V100 GPUs. For the dynamic resolution based training, we use batch size 18 with 6 V100 GPUs. The ZeroRA based finetuning cost only about half a day and one day for 256×256 and 256∼512 resolutions respectively.

*Equal contributions.
†Corresponding authors.

4. Upsampling Iteratively with SSU

Our Simple Structure Upsampler (SSU) introduced in Sec 3.2 can also work iteratively for larger image sizes. First, we should process the output edges $\mathbf{I}'_e$ and lines $\mathbf{I}'_l$ of SSU through shifted sigmoid as

$$\begin{aligned}\mathbf{I}'_e &= \text{sigmoid}[\gamma(\mathbf{I}'_e + \beta)],\\ \mathbf{I}'_l &= \text{sigmoid}[\gamma(\mathbf{I}'_l + \beta)],\end{aligned} \tag{1}$$

where $\gamma = 2, \beta = 2$ in our evaluation, and γ, β are randomly selected from $[1.5, 3]$ for the finetuning. Since the output size of SSU is doubled, we can repeat the inputs $\mathbf{I}_e, \mathbf{I}_l \in \mathbb{R}^{h\times w}$ for q times to achieve $\mathbf{I}'_e, \mathbf{I}'_l \in \mathbb{R}^{2^q h\times 2^q w}$. Then, the outputs can further be resized with the bilinear interpolation for the target size. In general, our SSU can get good and robust upsampled results for large sizes as shown in Fig. 1.

5. Supplementary Experiments

In this section, we provide some more qualitative and quantitative results to show the effects of our proposed components. Moreover, some details about the post-processing are also discussed.

5.1. More Qualitative Results

More qualitative results of Indoor and Places2 are shown in Fig. 2 and Fig. 3. Note that our method not only achieves better results in many man-made scenes, but also gets competitive results in natural scenes benefited from MPE and edges.

5.2. Quantitative Results with Different Masks

We show more quantitative results with different masking rates from 10% to 50% and mixture of segmentation and irregular masks in Tab. 2.

5.3. More Structural Experiments

TSR Ablations. For the Indoor dataset, we conducted several ablation experiments on our Transformer Structure

Table 1. Model settings of Transformer Structure Restoration (TSR), Structure Feature Encoder (SFE), and Fourier CNN Texture Restoration (FTR). GC, BN mean Gated Convolution [15] and BatchNorm; TConv2d, TGC indicate Transposed Conv2d and GC.

Transformer Structure Restoration (TSR)	Structure Feature Encoder (SFE)	Fourier CNN Texture Restoration (FTR)
Conv2d+ReLU($256 \times 256 \times 64$)	GC+BN+ReLU($256 \times 256 \times 64$)	Conv2d+BN+ReLU($256 \times 256 \times 64$)
Conv2d+ReLU($128 \times 128 \times 128$)	GC+BN+ReLU($128 \times 128 \times 128$)	Conv2d+BN+ReLU($128 \times 128 \times 128$)
Conv2d+ReLU($64 \times 64 \times 256$)	GC+BN+ReLU($64 \times 64 \times 256$)	Conv2d+BN+ReLU($64 \times 64 \times 256$)
Conv2d+ReLU($32 \times 32 \times 256$)	GC+BN+ReLU($32 \times 32 \times 512$)	Conv2d+BN+ReLU($32 \times 32 \times 512$)
TransformerBlock$\times 8$	DilatedResnetBlock$\times 3$	FFCBlock$\times 9$
TConv2d+ReLU($64 \times 64 \times 256$)	TGC+BN+ReLU($64 \times 64 \times 256$)	TConv2d+BN+ReLU($64 \times 64 \times 256$)
TConv2d+ReLU($128 \times 128 \times 128$)	TGC+BN+ReLU($128 \times 128 \times 128$)	TConv2d+BN+ReLU($128 \times 128 \times 128$)
TConv2d+ReLU($256 \times 256 \times 64$)	TGC+BN+ReLU($256 \times 256 \times 64$)	TConv2d+BN+ReLU($256 \times 256 \times 64$)
Conv2d+Sigmoid($256 \times 256 \times 2$)	–	Conv2d+Tanh($256 \times 256 \times 3$)

(a) Image (256) (b) Edge (512) (c) Edge (1024) (d) Edge (2048) (e) Line (512) (f) Line (1024) (g) Line (2048)

Figure 1. Iteratively outputs of SSU which have sizes from 512 to 2048. The results are consistent and robust.

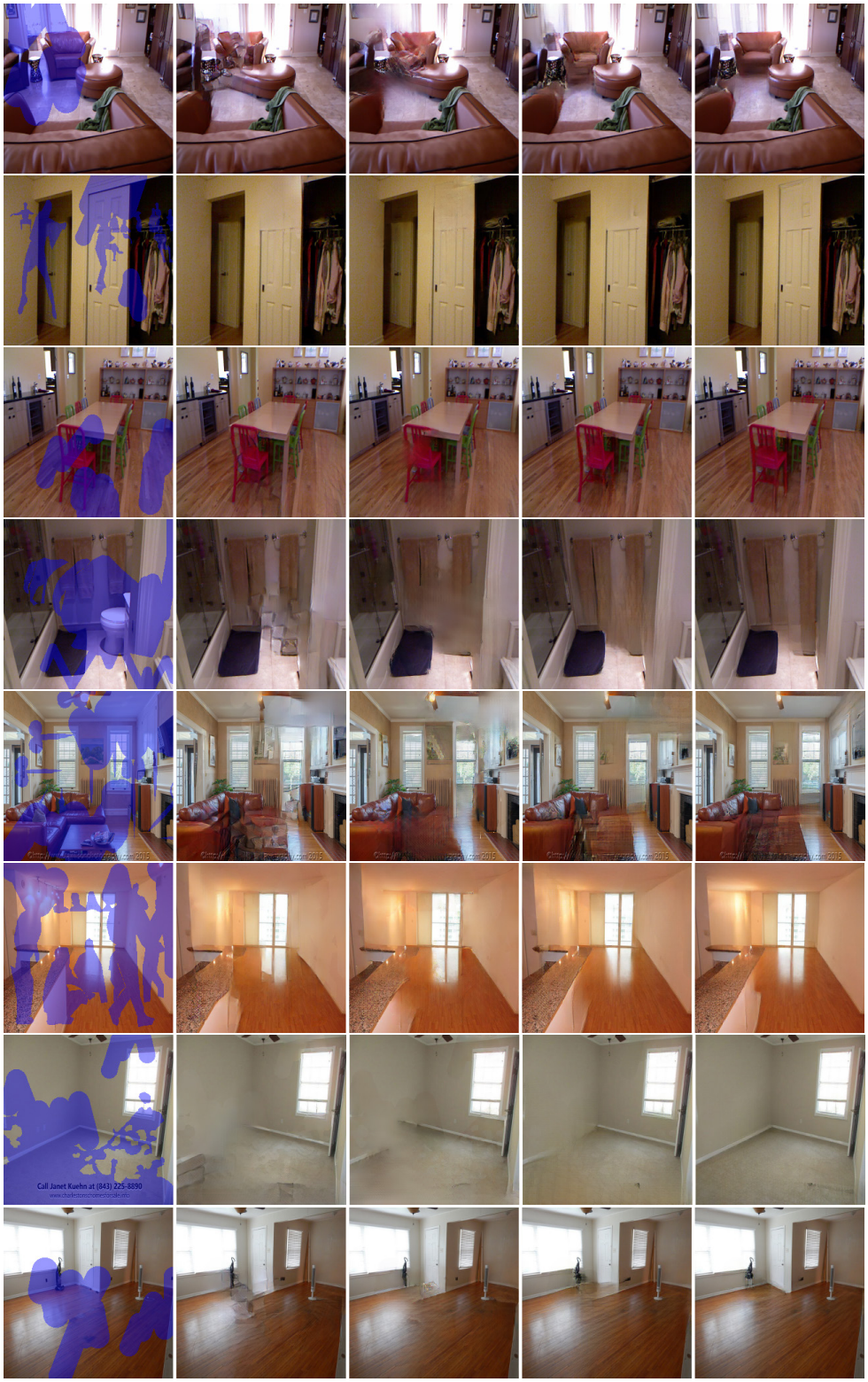
Restoration (TSR), and the results are displayed in Tab. 3 and Tab. 4. As illustrated in Tab. 3 and the first two rows of Tab. 4, replacing one standard self-attention module [12] with an axial attention module [8] in our Transformer Block can greatly reduce the GPU memory usage and speed up the model inference while keeping all metrics basically unchanged. Furthermore, we add the relative position encoding (RPE) [10] into our axial attention module, which can boost our results. Note that the RPE must be incorporated with the axial attention module in row-wise and column-wise, while standard attention based RPE costs much more GPU memory due to the long sequence. On the other hand, as we think that a higher recall will benefit the later image inpainting, we further multiply the line logits by 4 before feeding it through the sigmoid activation function in all the experiments. This strategy enhances recall while only compromising a little precision.

More Structural Qualitative Results. We show some more structural results of TSR in Fig. 4 compared with MST [1]. Our TSR can outperform the CNN based method. Furthermore, edges and lines from TSR can effectively guide the final inpainted results.

5.4. Effects of Mask-Predict

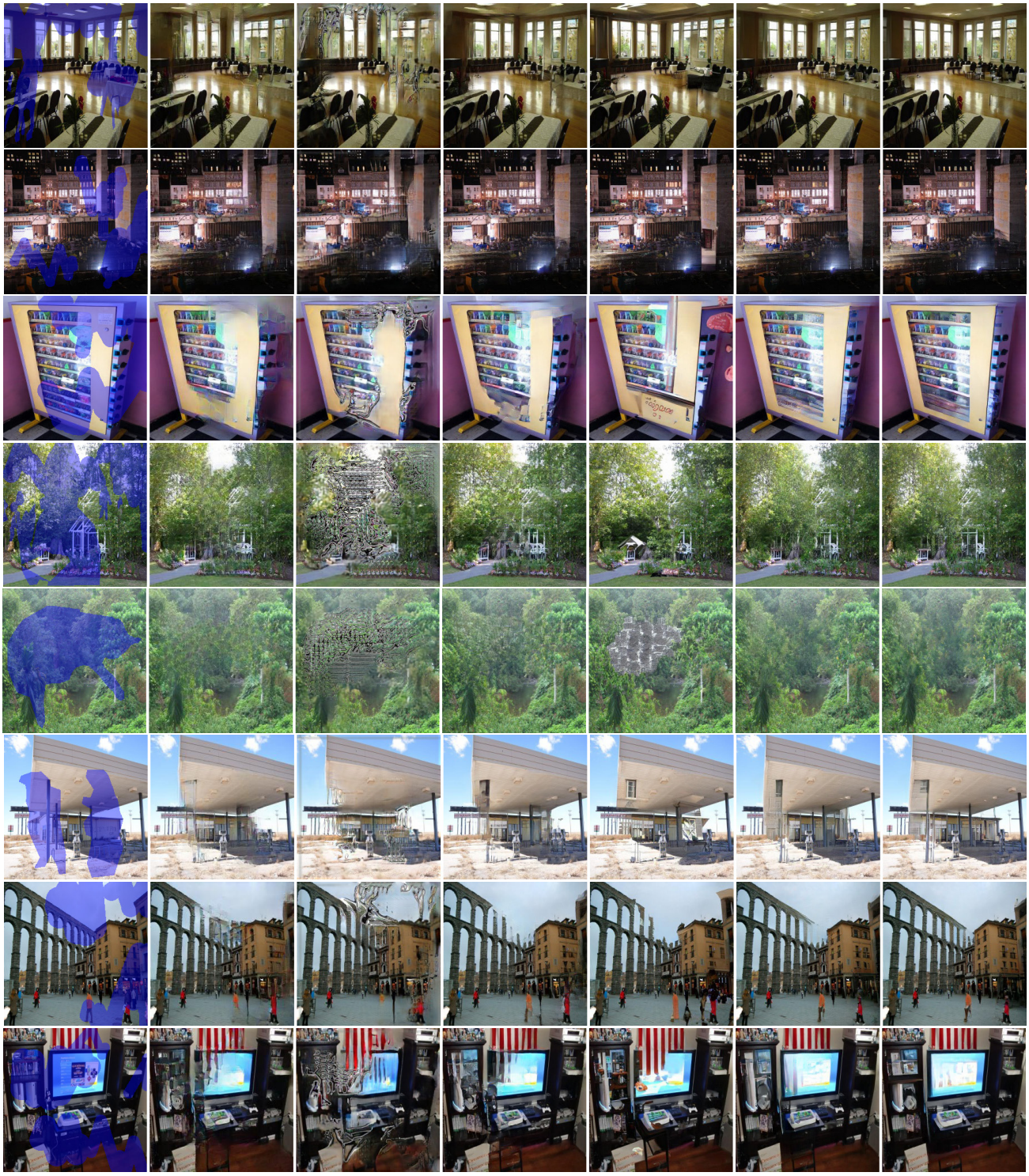
During the inference, Mask-Predict [3, 6, 7] is used in TSR which is a non-autoregressive sampling method. Mask-Predict predicts all target pixels at the first iteration. Then we re-mask and re-predict pixels with low-confidence iteratively. Mask-Predict can greatly enrich the generated results without heavy costs.

Since our TSR can output a 256×256 probability map for edge and line respectively; of course, we can directly use this probability map as the repair result of our edge and line. However, we find that the recall is still insufficient. Fig. 5(b)(c) show that it can only predict a few regions with high confidence. But the probability confidence for the inner masked region is low, which leads to incomplete structures. As a result, we employ the Mask-Predict [3, 6, 7] strategy. It predicts all target pixels at the first iteration. Then we re-mask and re-predict pixels with low-confidence iteratively for a constant number of iterations. Note that we just re-mask edge and line without re-masking the input masked image. This technique can achieve more complete structural information as illustrated in Fig. 5(d)(e). We set the total number of iterations to 5 in our experiment for a trade-off between inference time and recall. Fig. 6 shows



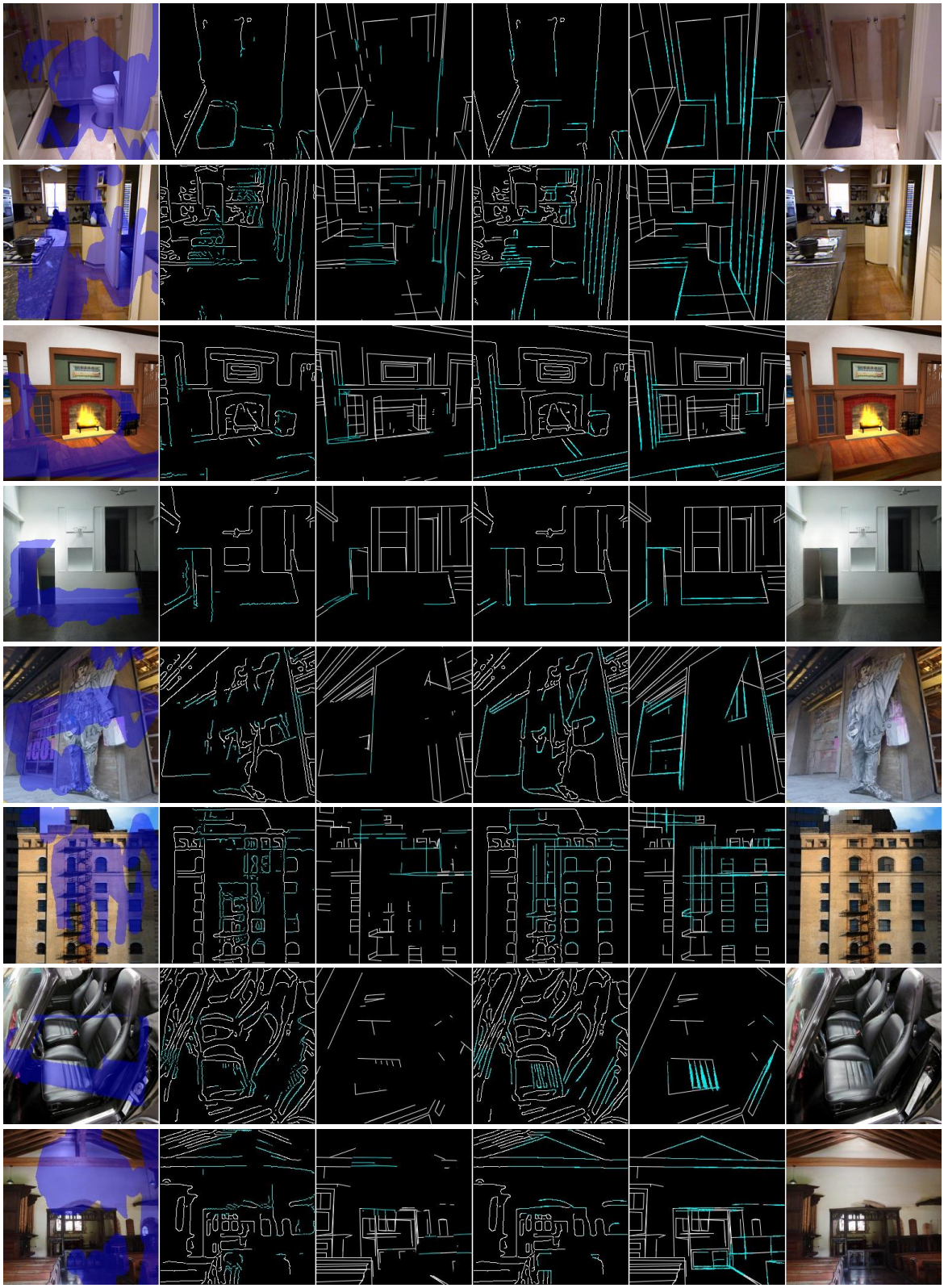
(a) Masked Input (b) EC (c) MST (d) LaMa (e) Ours

Figure 2. Qualitative results of Indoor dataset compared among EC [9], MST [1], LaMa [11], and ours. Zoom-in for details



(a) Masked Input (b) EC (c) HiFill (d) MST (e) Co-Mod (f) LaMa (g) Ours

Figure 3. Qualitative results of Places2 dataset compared among EC [9], HiFill [14], MST [1], Co-Mod [16], LaMa [11], and ours. Zoom-in for details



(a) Masked Input (b) Edges from MST (c) Lines from MST (d) Edges from ours (e) Lines from ours (f) Our results

Figure 4. Predicted edges, lines and inpainted results in Indoor and Places2 compared with MST [1]. The first four examples are from the Indoor dataset; the last four examples are from the Places2 dataset. Blue edges (lines) indicate reconstruction from models.

Table 2. Quantitative inpainting results on Indoor and Places2 with different mask ratios.

		Indoor (256×256)				Places2 (256×256)					
	Mask	EC	MST	LaMa	Ours	EC	HiFill	Co-Mod	MST	LaMa	Ours
PSNR↑	10~20%	28.18	28.72	29.05	**29.87**	26.60	24.04	26.40	28.13	28.23	**28.31**
	20~30%	25.14	25.66	25.96	**26.66**	24.26	21.64	23.61	25.07	25.31	**25.40**
	30~40%	23.02	23.53	23.87	**24.64**	22.59	19.96	21.73	23.07	23.43	**23.51**
	40~50%	21.55	22.02	22.39	**23.13**	21.27	18.63	20.28	21.53	22.03	**22.11**
	Mixed	24.07	24.52	25.20	**25.57**	23.31	20.76	22.57	24.02	24.37	**24.42**
SSIM↑	10~20%	0.951	0.954	0.956	**0.961**	0.913	0.883	0.926	0.941	**0.942**	**0.942**
	20~30%	0.916	0.922	0.925	**0.933**	0.872	0.818	0.880	0.898	0.901	**0.902**
	30~40%	0.876	0.886	0.890	**0.901**	0.828	0.751	0.831	0.852	0.859	**0.860**
	40~50%	0.835	0.848	0.855	**0.870**	0.783	0.682	0.781	0.803	0.814	**0.817**
	Mixed	0.884	0.894	0.902	**0.907**	0.839	0.770	0.843	0.862	0.869	**0.870**
FID↓	10~20%	9.56	8.56	8.01	**7.18**	1.95	4.71	0.52	0.76	0.45	**0.43**
	20~30%	16.22	15.88	13.23	**12.13**	3.79	11.93	1.00	1.86	0.95	**0.88**
	30~40%	23.48	22.69	18.77	**16.51**	6.98	25.16	1.64	3.83	1.72	**1.55**
	40~50%	31.16	31.06	23.47	**20.87**	11.49	44.68	**2.38**	6.80	2.81	2.51
	Mixed	22.02	21.65	16.97	**15.93**	6.21	21.33	1.49	3.53	1.63	**1.47**
LPIPS↓	10~20%	0.054	0.050	0.044	**0.038**	0.073	0.119	0.053	0.047	0.047	**0.042**
	20~30%	0.094	0.087	0.078	**0.068**	0.111	0.189	0.098	0.082	0.083	**0.073**
	30~40%	0.140	0.129	0.117	**0.101**	0.152	0.265	0.140	0.120	0.121	**0.107**
	40~50%	0.189	0.172	0.156	**0.136**	0.194	0.343	0.184	0.160	0.161	**0.143**
	Mixed	0.135	0.122	0.112	**0.098**	0.149	0.137	0.246	0.122	0.155	**0.108**

Table 3. Efficient ablations of axial attention module. FPS is the Frames Per Second during the inference. The GPU memory is test on single Tesla V100 with batch size 8.

	FPS	GPU Memory (MB)
w./o. Axial	6.41	14845
with Axial	**7.89**	**10547**

our outcomes with different Mask-Predict iterations. Note that we ignore pixels with low confidence for each iteration in Fig. 6, so the iteration 1 results of Fig. 6(c) look different from the results without Mask-Predict. The inside portion of the mask can be gradually restored with larger iterations as shown in Fig. 6.

5.5. User Study

We conduct user study on several models to validate the effectiveness of our model from the perspective of human. Specifically, we invite 10 volunteers who are not familiar with image inpainting to judge the quality of inpainted images. On Indoor and Places2, four methods are compared, which including EC [9], MST [1] LaMa [11] and ours. Given the masked inputs, we randomly shuffle and combine the results of four methods together. Then, volunteers are required to choose the best one from each group. As shown in Fig. 7, our method outperforms other three competitors on both two datasets. Especially, our method can achieve a great advantage compared with the baseline method *i.e.*, LaMa.

5.6. Results of Rectangular Masks

Here we provide some results of 40% center rectangular masks of 1k Places(512) images without any retraining in Tab. 5. Note that Co-Mod [16] is the only one trained with some rectangular masks while other methods have not been trained with similar masks. Moreover, we compare related qualitative results in Fig. 8. And the classical exemplar-based inpainting [5] is also included. Traditional exemplar-based method fails to work properly and is time-consuming. Co-Mod has hallucinated artifacts instead of generating plausible results. And LaMa results are blur with still high PSNR.

5.7. Comparisons of Texture Images

We further compare our method with LaMa on 1,880 texture images [4] in Tab. 6 and Fig. 9, which contain strong periodic textures. Since this dataset is very suitable to LaMa [11], our method still has competitive performance.

5.8. Results of MatterPort3D

We use the test set of MatterPort3D [2] to evaluate the effectiveness of our method in the high-resolution structure recovery. MatterPort3D images tested in this paper are consisted of 1,965 indoor images in 1280×1024. We resized them into 1024×1024 as shown in Fig. 10. We provide some qualitative results of our method and LaMa compared on MatterPort3D in Fig. 11. For these structural images, our results enjoy better structures.

Table 4. Ablation studies of TSR on the Indoor dataset, where P., R., F1 mean Precision, Recall, and F1-score.

Axial	RPE	Edge P.	Edge R.	Edge F1	Line P.	Line R.	Line F1	Avg F1
		38.27	33.12	34.78	52.93	65.79	57.73	46.26
✔		**38.30**	32.90	34.64	52.74	**66.48**	57.87	46.26
✔	✔	37.34	**34.25**	**35.10**	**53.60**	66.23	**58.35**	**46.72**

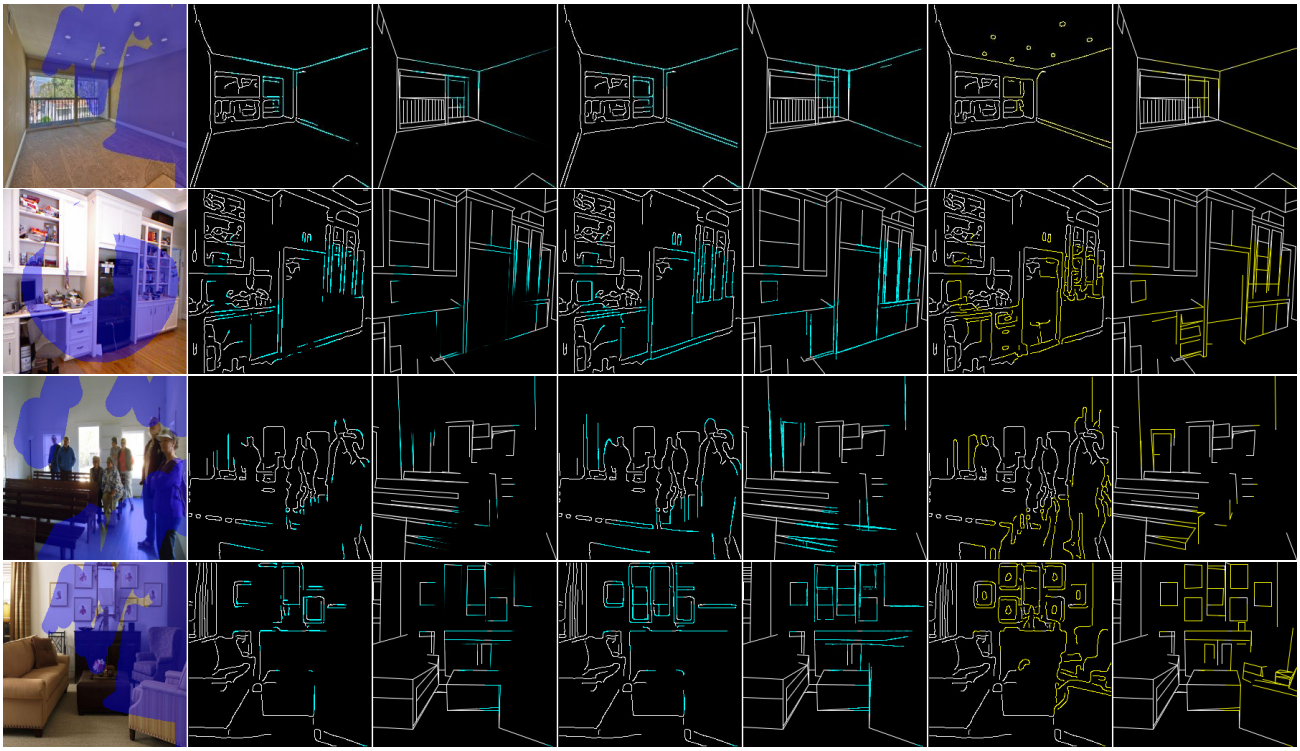


Figure 5. Ablation studies on Mask-Predict. From left to right: (a) Image, (b) Reconstructed edge w/o Mask-Predict, (c) Reconstructed line w/o Mask-Predict, (d) Reconstructed edge with Mask-Predict, (e) Reconstructed line with Mask-Predict, (f) Ground truth edge, (g) Ground truth line. The first two rows are from the Indoor dataset; the last two rows are from the Places2 dataset. Blue and yellow edges (lines) indicate reconstruction and ground truth within mask region respectively.

Table 5. Quantitative results on 1k Places 512 images with 40% center rectangular masks.

	PSNR	FID	LPIPS
Co-Mod	17.59	**52.38**	0.262
LaMa	**19.69**	61.67	0.268
Ours	19.65	55.85	**0.239**

Table 6. Quantitative results on 512 texture images from [4].

	LaMa	Ours
PSNR	**25.82**	25.67
SSIM	**0.875**	0.869
FID	12.86	**11.67**
LPIPS	0.138	**0.134**

6. More High Resolution Results

In Fig. 12, Fig. 13, and Fig. 14 we provide some object removal instances in large images from 1k to 2k resolutions compared with LaMa [11]. Some cases are selected from the open-source testset of LaMa. Note that our method outperforms LaMa in scenes with weak textures such as row 2 in Fig. 12 and row 1 in Fig. 13. For the cases with sparse regular textures and lines (rows 1,3 of Fig. 12), our method can still achieve more clear borderlines. For the cases with dense regular textures (rows 2,3 of Fig. 13), LaMa gets competitive results, which shows that FFC in frequency fields

Figure 6. Mask-Predict for edges and lines in Indoor. The first two examples are from the Indoor dataset; the last two examples are from the Places2 dataset. For each example, the structure in the first row is the edge; the structure in the second row is the line. Blue and yellow edges (lines) indicate our reconstruction and ground truth within mask region respectively.

has solved these problems properly. However, our method can also achieve results with less blur that benefited from precise structural constraints. For the larger case with 2048 image size in Fig. 14, our method can still get more consistent result compared with LaMa.

7. Limitations

We summarize the limitation of our method in this section. As shown in Fig. 15, since our method only recovers edges and lines in 256×256, some distant views failed to be described correctly with the limited size. Therefore, some

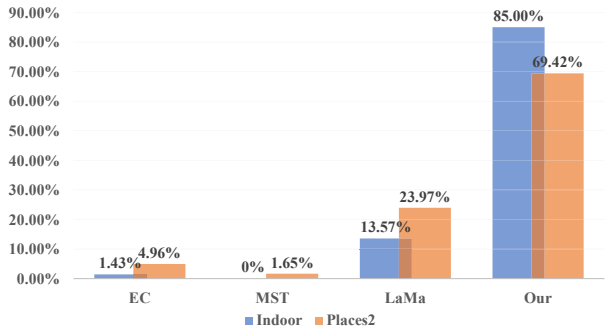


Figure 7. Average scores of Indoor and Places2 for user studies, which are collected from volunteers who select the best one from shuffled inpainted images.

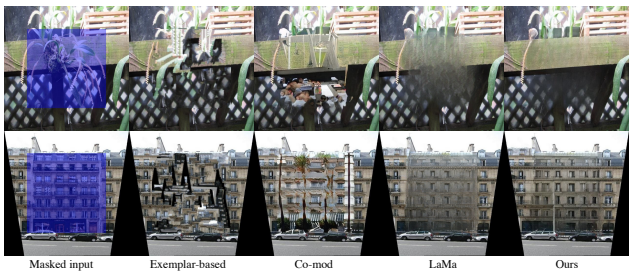


Figure 8. Inpainting results of 512 images compared with Exemplar-based inpainting [5], Co-Mod, LaMa, and ours.

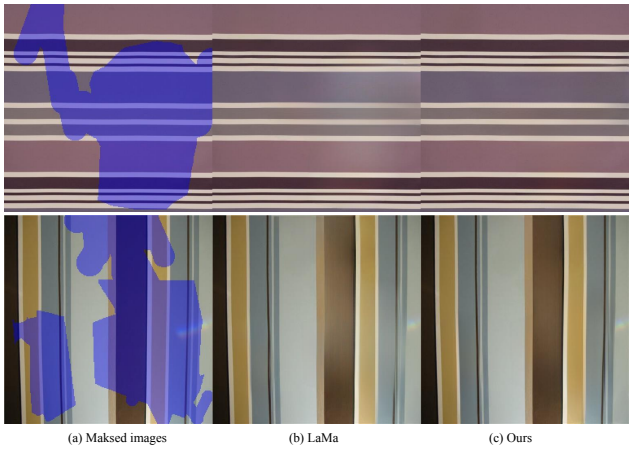


Figure 9. Inpainting results of 512 texture images [4] compared with LaMa and ours.

complex urban distant scenes can not be enhanced by the structures of canny edges and wireframe lines.

References

[1] Chenjie Cao and Yanwei Fu. Learning a sketch tensor space for image inpainting of man-made scenes. *arXiv preprint arXiv:2103.15087*, 2021. 2, 3, 4, 5, 6

Figure 10. Examples of resized 1024×1024 MatterPort3D images.

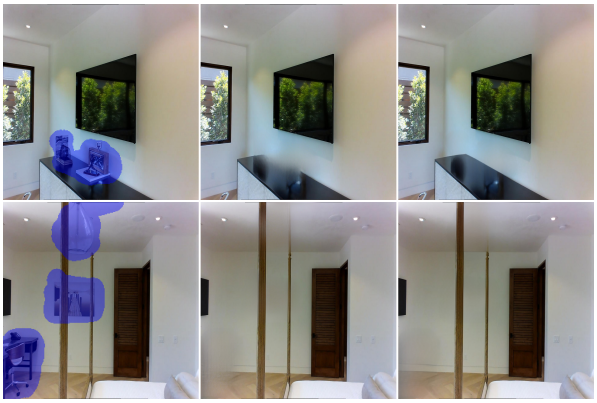


Figure 11. Inpainting results of LaMa [11] and ours in 1024×1024 MatterPort3D images.

[2] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. 6

[3] Jaemin Cho, Jiasen Lu, Dustin Schwenk, Hannaneh Hajishirzi, and Aniruddha Kembhavi. X-lxmert: Paint, caption and answer questions with multi-modal transformers, 2020. 2

[4] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. 6, 7, 9

[5] Antonio Criminisi, Patrick Perez, and Kentaro Toyama. Ob-

Figure 12. High-resolution object removal results. From left to right are masked inputs, results from LaMa [11], results from our method. Please zoom-in for more details.

ject removal by exemplar-based inpainting. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–II. IEEE, 2003. 6, 9

[6] Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. *arXiv preprint arXiv:1904.09324*, 2019. 2

[7] Junliang Guo, Zhirui Zhang, Linli Xu, Hao-Ran Wei, Boxing Chen, and Enhong Chen. Incorporating bert into parallel sequence decoding with adapters. *arXiv preprint arXiv:2010.06138*, 2020. 2

[8] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers.

Figure 13. High-resolution object removal results. From left to right are masked inputs, results from LaMa [11], results from our method. Please zoom-in for more details.

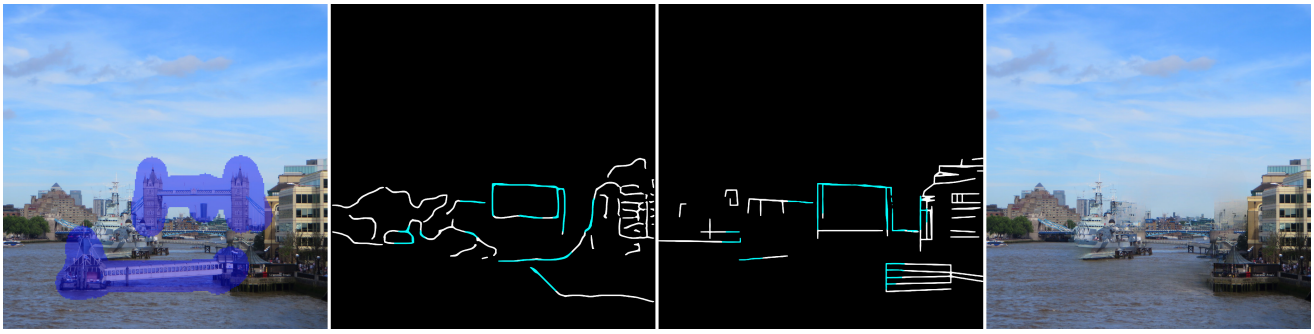
arXiv preprint arXiv:1912.12180, 2019. 2

[9] 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 Workshops*, 2019. 1, 3, 4, 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. *arXiv preprint arXiv:1910.10683*, 2019. 2

[11] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor

Figure 14. The high-resolution inpainting comparison of 2048×2048. From left to right are the masked input, the result from LaMa [11], the result from our method. Please zoom-in for more details.



(a) Masked input (b) Predicted edges (c) Predicted lines (d) Inpainted results

Figure 15. Failed 1024×1024 results of our method. Some distant views failed to be described correctly by our grayscale sketch space of edges and lines. So these distant views are blurry.

Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. *arXiv preprint arXiv:2109.07161*, 2021. 1, 3, 4, 6, 7, 9, 10, 11, 12

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2

[13] Rui Xu, Xintao Wang, Kai Chen, Bolei Zhou, and Chen Change Loy. Positional encoding as spatial inductive bias in gans, 2020. 1

[14] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7508–7517, 2020. 4

[15] 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/CVF International Conference on Computer Vision*, pages 4471–4480, 2019. 2

[16] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. *arXiv preprint arXiv:2103.10428*, 2021. 4, 6