Supplementary Material

A. Network Architecture

Module	In Channel	Out channel	Kernel Size	Down/Up/No	Batch Norm	Activation Func	Edge Inpput	Feature Input	Product
VSR1	3	64	7	Down	T	ReLU	Masked Edge	Masked Image	Size 256 Edge Part 1
VSR2	64	128	5	Down	T	ReLU	VSR1 Out(Down)	VSR1	Size 128 Edge Part 1
PConv1	128	256	3	Down	T	ReLU		VSR2	
PConv2	256	512	3	Down	T	ReLU		PConv1	
PConv3	512	512	3	Down	T	ReLU		PConv2	
PConv4	512	512	3	Down	T	ReLU		PConv3	
PConv5	512	512	3	Down	T	ReLU		PConv4	
PConv6	512	512	3	Down	T	ReLU		PConv5	
DeConv	512	512	4	Up	F	None		PConv6	
Partial-Deconv1	512+512	512	3&4	Up	T	LeakyReLu		Deconv+PConv5	
Partial-Deconv2	512+512	512	3&4	Up	T	LeakyReLu		PartialDeconv1+PConv4	
Partial-Deconv3	512+512	512	3&4	Up	T	LeakyReLu		PartialDeconv2+PConv3	
Partial-Deconv4	512+512	512	3&4	Up	T	LeakyReLu		PartialDeconv3+PConv2	
Partial-Deconv5	512+256	256	3&4	Up	T	LeakyReLu		PartialDeconv4+PConv1	
PixelAttention	256	256		No	F	None		PartialDeconv5	
Partial-Deconv5	256+128	128	3&4	Up	T	LeakyReLu		PixelAttention+VSR2	
VSR3(Deconv)	128+64	64	3&4	Up	T	LeakyReLu	VSR2 Out	PartialDeconv6+VSR1	Size 128 Edge
VSR4(NoDeconv)	64+3	64	3	No	T	LeakyReLu	VSR1 Out	VSR3+MaskedImage	Size 256 Edge
Bottleneck Block	64	64	1&3&1	No	T	ReLU		VSR4	
Output	64+64	3	1	No	F	None		Residual Block + VSR4	RGB Image

Table 3: Design Detail of Network Architecture. "In Channel" and "Out Channel" mean the number of channels in input feature. "Kernel Size" is the size of convolutional or deconvolutional kernel. For Partial-Deconvolution layers, the first number is the kernel size of partial convolution and the second number is the size of deconvolutional kernel. Down/Up/No means down-sampling, up-sampling or no special operation on feature size. All down-sampling and up-sampling operations are implemented with stride 2. BatchNorm is whether the layer contains batch normalization calculation. Activation Func is the activation function chosen for the layer. Edge input is where the edge comes from. Feature input is where the feature map comes from. Product means the final output tensor from the model.

A.1. Generator

The network has 8 down-sampling layers and 8 up-sampling layers, including 2 VSR layers in the start and 2 VSR layers in the end of our model. As shown in Table 5. Both down-sampling and up-sampling operation use the stride length 2. The convolution sizes are 7 and 5 for the first and second VSR layers respectively. The kernel sizes for edge generation are consistent with that of convolution layers. The edge from VSR1 is down-sampled as the input of VSR 2. The implementation of pixel attention is modified from contextual attention module, where we used the patch size of 1 and propagation size of 3. A bottleneck residual block is added to the end of model.

A.2. Discriminator

The discriminator is constituted by a pre-trained and fixed VGG-16 network and a patch discriminator. For the patch discriminator, we use 5 convolution layers, where the strides lengths are 2 and the kernel sizes are 4. Spectral normalization is used. LeakyReLU is chosen to be the activation functions of the layers. For the output of discriminator, Sigmoid is used as the activation function. With a fully convolutional design, the discriminator can accept inputs of different scales.

B. Proof

This appendix collects all the proofs omitted from the main text.

B.1. Proof of Theorem 1

This section provides a detailed proof for Theorem 1 which is committed from the main text. We first recall two lemmas by Bartlett *et al*. [2] .

Lemma 1 (cf. [2], Lemma A.7)**.** *Suppose there are L weight matrices in a chain-like neural network. Let $(\varepsilon_1, \ldots, \varepsilon_L)$ be given. Suppose the L weight matrices $(A_1, \ldots, A_L)$ lies in $\mathcal{B}_1 \times \ldots \times \mathcal{B}_L$, where $\mathcal{B}_i$ is a ball centered at 0 with the radius*

of s_i, i.e., $\mathcal{B}_i = \{A_i : \|A_i\| \leq s_i\}$. Furthermore, suppose the input data matrix X is restricted in a ball centred at 0 with the radius of B, i.e., $\|X\| \leq B$. Suppose F is a hypothesis function computed by the neural network. If we define:

$$\mathcal{H} = \{F(X) : A_i \in \mathcal{B}_i, A_t^{u,v,s} \in \mathcal{B}_t^{u,v,s}\}, \tag{B.1}$$

where $i = 1, \ldots, L$, $(u, v, s) \in I_V$, and $t \in \{1, \ldots, L^{u,v,s}\}$. Let $\varepsilon = \sum_{j=1}^{L} \varepsilon_j \rho_j \prod_{l=j+1}^{L} \rho_l s_l$. Then we have the following inequality:

$$\mathcal{N}(\mathcal{H}) \leq \prod_{i=1}^{L} \sup_{\mathbf{A}_{i-1} \in \boldsymbol{\mathcal{B}}_{i-1}} \mathcal{N}_i, \tag{B.2}$$

where $\mathbf{A}_{i-1} = (A_1, \ldots, A_{i-1})$, $\boldsymbol{\mathcal{B}}_{i-1} = \mathcal{B}_1 \times \ldots \times \mathcal{B}_{i-1}$, and

$$\mathcal{N}_i = \mathcal{N}\left(\left\{A_i F_{\mathbf{A}_{i-1}}(X) : A_i \in \mathcal{B}_i\right\} \varepsilon_i, \|\cdot\|\right). \tag{B.3}$$

Here, the radius of each covers are respectively,

$$\varepsilon_i = \frac{\alpha_i \varepsilon}{\rho_i \prod_{j>i} \rho_j s_j}, \tag{B.4}$$

where

$$\alpha_i = \frac{1}{\bar{\alpha}} \left(\frac{b_i}{s_i}\right)^{2/3}, \tag{B.5}$$

$$\bar{\alpha} = \sum_{j=1}^{L} \left(\frac{b_j}{s_j}\right)^{2/3}. \tag{B.6}$$

Lemma 2 (cf. [2], Lemma 3.2)**.** *Let conjugate exponents (p, q) and (r, s) be given with $p \leq 2$, as well as positive reals (a, b, ε) and positive integer m. Let matrix $X \in \mathbb{R}^{n \times d}$ be given with $\|X\|_p \leq b$. Let $\mathcal{H}_A$ denote the family of matrices obtained by evaluating X with all choices of matrix A:*

$$\mathcal{H}_A \triangleq \left\{XA | A \in \mathbb{R}^{d \times m}, \|A\|_{q,s} \leq a\right\}. \tag{B.7}$$

Then

$$\log \mathcal{N}\left(\mathcal{H}_A, \varepsilon, \|\cdot\|_2\right) \leq \left\lceil \frac{a^2 b^2 m^{2/r}}{\varepsilon^2} \right\rceil \log(2dm). \tag{B.8}$$

This covering bound constrains the hypothesis complexity contributed by a single weight matrix.

Proof of Theorem 1. Suppose the hypothesis spaces of the output functions $F_{(A_1,\ldots,A_{i-1})}$ of the weight matrices $A_i, i = 1, \ldots, 5$ are respectively $\mathcal{H}_i, i = 1, \ldots, 5$. From Lemma 1, we can directly get the following inequality,

$$\begin{aligned}
&\log \mathcal{N}(\mathcal{F}|S) \\
\leq &\log \left(\prod_{i=1}^{5} \sup_{\mathbf{A}_{i-1} \in \boldsymbol{\mathcal{B}}_{i-1}} \mathcal{N}_i \right) \\
\leq &\sum_{i=1}^{5} \log \left(\sup_{\substack{(A_1,\ldots,A_{i-1}) \\ \forall j<i, A_j \in B_j}} \mathcal{N}\left(\left\{A_i F_{(A_1,\ldots,A_{i-1})}\right\}, \varepsilon_i, \|\cdot\|_2\right) \right).
\end{aligned} \tag{B.9}$$

Employ eq. (B.8), we can get the following inequality,

$$\log \mathcal{N}(\mathcal{F}|S) \leq \sum_{i=1}^{5} \frac{b_i^2 \|F_{(A_1,\ldots,A_{i-1})}(X)\|_\sigma^2}{\varepsilon_i^2} \log\left(2W^2\right). \tag{B.10}$$

Meanwhile,

$$\begin{aligned}\|F_{(A_1,\ldots,A_{i-1})}(X)\|_\sigma^2 =&\|\sigma_{i-1}(A_{i-1}F_{(A_1,\ldots,A_{i-2})}(X)) - \sigma_{i-1}(0)\|_2\\ \leq&\|\sigma_{i-1}\|\|A_{i-1}F_{(A_1,\ldots,A_{i-2})}(X) - 0\|_2\\ \leq&\rho_{i-1}\|A_{i-1}\|_\sigma\|F_{(A_1,\ldots,A_{i-2})}(X)\|_2\\ \leq&\rho_{i-1}s_{i-1}\|F_{(A_1,\ldots,A_{i-2})}(X)\|_2.\end{aligned} \tag{B.11}$$

Therefore,

$$\|F_{(A_1,\ldots,A_{i-1})}(X)\|_\sigma^2 \leq \|X\|^2\prod_{j=1}^{i-1}s_i^2\rho_i^2. \tag{B.12}$$

Suppose the covering number radius of the final output hypothesis space is ε, then we can get the formulation of each covering number radius ε_i throughout the neural network in term of the radius ε. Specifically, motivated by the proof given in [2], we can get the following equations:

$$\varepsilon_{i+1} = \rho_i s_{i+1}\varepsilon_i. \tag{B.13}$$

Then,

$$\varepsilon_5 = \rho_1\prod_{i=2}^{4}s_i\rho_i s_5\epsilon_1, \tag{B.14}$$

$$\varepsilon = \rho_1\prod_{i=2}^{5}s_i\rho_i\epsilon_1. \tag{B.15}$$

Therefore,

$$\varepsilon_i = \frac{\rho_i\prod_{j=1}^{i-1}s_j\rho_j}{\prod_{j=1}^{5}s_j\rho_j}\varepsilon. \tag{B.16}$$

Therefore,

$$\log\mathcal{N}\left(\mathcal{F}|_S,\varepsilon,\|\cdot\|_2\right) \leq \frac{\log\left(2W^2\right)\|X\|_2^2}{\varepsilon^2}\left(\prod_{i=1}^{5}s_i\rho_i\right)^2\sum_{i=1}^{5}\frac{b_i^2}{s_i^2}. \tag{B.17}$$

Here $\rho_1 = \rho_2 = \rho_3 = \rho_4 = 1$ and $\rho_5 = \rho$. Therefore,

$$\log\mathcal{N}\left(\mathcal{F}|_S,\varepsilon,\|\cdot\|_2\right) \leq \frac{\log\left(2W^2\right)\|X\|_2^2}{\varepsilon^2}\left(\rho\prod_{i=1}^{5}s_i\right)^2\sum_{i=1}^{5}\frac{b_i^2}{s_i^2}. \tag{B.18}$$

which is exactly eq. (4.3) of Theorem 1.

The proof is completed. □

B.2. Proof of Theorem 2

This section provides a detailed proof for Theorem 2 which is committed from the main text. We first recall a recent lemma generally addressing the generalisation ability of GAN and a classic lemma in statistical learning theory.

Lemma 3 (cf. [34], p. 8, Theorem 3.1)**.** *Assume that the discriminator set F is even, i.e., $f \in \mathcal{F}$ implies $-f \in \mathcal{F}$, and that all discriminators are bounded by Δ, i.e., $\|f\|_\infty \leq \Delta$ for any $f \in \mathcal{F}$. Let $\hat{\mu}_N$ be an empirical measure of an independent and identical (i.i.d.) sample of size N drawn from a distribution μ. Assume $\nu_N \in \mathcal{G}$ satisfies*

$$d_{\mathcal{F}}(\hat{\mu}_N,\nu_N) \leq \inf_{\nu\in\mathcal{G}}d_{\mathcal{F}}(\hat{\mu}_N,\nu) + \phi. \tag{B.19}$$

Then with probability at least $1-\delta$, we have

$$d_{\mathcal{F}}(\mu,\nu_N) - \inf_{\nu\in\mathcal{G}}d_{\mathcal{F}}(\mu,\nu) \leq 2\mathfrak{R}_N^{(\mu)}(\mathcal{F}) + 2\Delta\sqrt{\frac{2\log(\frac{1}{\delta})}{N}} + \phi, \tag{B.20}$$

where

Computing the empirical Rademacher complexity of neural network could be extremely difficult and thus still remains an open problem. Fortunately, the empirical Rademacher complexity can be upper bounded by the corresponding ε-covering number $N(\mathcal{F}, \varepsilon, \|\cdot\|_2)$ as the following lemma states.

Lemma 4 (cf. [2], Lemma A.5)**.** *Suppose* $\mathbf{0} \in \mathcal{H}$ *and all conditions in Lemma 3 hold. Then*

$$\mathfrak{R}_N^{(\mu)}(\mathcal{F}) \leq \inf_{\alpha>0}\left(\frac{4\alpha}{\sqrt{n}} + \frac{12}{n}\int_{\alpha}^{\sqrt{n}} \sqrt{\log \mathcal{N}(l\circ\mathcal{H}, \varepsilon, \|\cdot\|_2)}d\varepsilon\right). \tag{B.21}$$

Proof of Theorem 2. Apply Lemma 4 directly to Theorem 1, we can get the following equation

$$\begin{aligned}\mathfrak{R}_N^{(\mu)} &\leq \inf_{\alpha>0}\left(\frac{4\alpha}{\sqrt{n}} + \frac{12}{n}\int_{\alpha}^{\sqrt{n}} \sqrt{\log \mathcal{N}(\mathcal{H}_\lambda|_D, \varepsilon, \|\cdot|_2)}\mathrm{d}\varepsilon\right)\\ &\leq \inf_{\alpha>0}\left(\frac{4\alpha}{\sqrt{n}} + \frac{12}{n}\int_{\alpha}^{\sqrt{n}} \frac{R}{\varepsilon}\mathrm{d}\varepsilon\right)\\ &\leq \inf_{\alpha>0}\left[\frac{4\alpha}{\sqrt{n}} + \frac{12}{n}\sqrt{R}\log\left(\frac{\sqrt{n}}{\alpha}\right)\right].\end{aligned} \tag{B.22}$$

Apparently, the infinimum is reached uniquely at $\alpha = 3\sqrt{\frac{R}{n}}$ and the infinitum is as follows,

$$\mathfrak{R}_N^{(\mu)} \leq \frac{12R}{N}\left[1 + \log\left(\frac{N}{3R}\right)\right]. \tag{B.23}$$

Apply eq. (B.22) to eq. (B.20) of Lemma 3, we can directly get the fowling equation,

$$d_{\mathcal{F}}(\mu, \nu_N) - \inf_{\nu\in\mathcal{G}} d_{\mathcal{F}}(\mu, \nu) \leq \frac{24R}{N}\left(1 + \log\frac{N}{3R}\right) + 2\Delta\sqrt{\frac{2\log(\frac{1}{\delta})}{N}} + \phi, \tag{B.24}$$

which is exactly eq. (4.5). The proof is completed. □

C. More results

PSV-SSIM	P-UNet	GatedConv	Edge-Connect	PRVS(Ours)	CelebA-SSIM	P-UNet	GatedConv	Edge-Connect	PRVS(Ours)
10%-20%	0.953	0.959	0.956	0.964	10%-20%	0.979	0.976	0.978	0.982
20%-30%	0.910	0.920	0.917	0.928	20%-30%	0.958	0.954	0.957	0.962
30%-40%	0.858	0.873	0.869	0.885	30%-40%	0.930	0.927	0.928	0.937
40%-50%	0.780	0.815	0.811	0.832	40%-50%	0.896	0.892	0.891	0.905
50%-60%	0.678	0.684	0.698	0.724	50%-60%	0.816	0.805	0.801	0.832
PSV-PSNR	P-UNet	GatedConv	Edge-Connect	PRVS(Ours)	CelebA-PSNR	P-UNet	GatedConv	Edge-Connect	PRVS(Ours)
10%-20%	30.76	31.42	31.05	32.00	10%-20%	32.68	32.69	32.53	33.23
20%-30%	27.62	28.12	28.05	28.79	20%-30%	29.33	29.45	29.19	29.83
30%-40%	25.51	25.80	25.98	26.62	30%-40%	26.87	27.01	26.72	27.32
40%-50%	23.81	23.93	24.29	24.87	40%-50%	24.90	24.98	24.67	25.34
50%-60%	21.56	21.06	21.94	22.48	50%-60%	22.08	21.83	21.44	22.53
PSV-MAE	P-UNet	GatedConv	Edge-Connect	PRVS(Ours)	CelebA-MAE	P-UNet	GatedConv	Edge-Connect	PRVS(Ours)
10%-20%	0.0123	0.0126	0.0111	0.0105	10%-20%	0.0082	0.0085	0.0085	0.0077
20%-30%	0.0212	0.0207	0.0195	0.0182	20%-30%	0.0150	0.0154	0.0154	0.0140
30%-40%	0.0309	0.0300	0.0287	0.0266	30%-40%	0.0231	0.0233	0.0237	0.0216
40%-50%	0.0421	0.0406	0.0393	0.0363	40%-50%	0.0327	0.0329	0.0337	0.0306
50%-60%	0.0607	0.0621	0.0576	0.0534	50%-60%	0.0512	0.0529	0.0541	0.0482

Table 4: Quantitative comparisons on CelebA dataset and Paris Street View dataset. We compare our model with P-UNet[12], Edge-Connect [16] and Gated Convolution [31]. We cropped the center 178×178 pixels in CelebA and resize them to 256×256.

PSV-SSIM	P-UNet*	PD	PD+VSR	PSV-PSNR	P-UNet*	PD	PD+VSR	PSV-MAE	P-UNet*	PD	PD+VSR
10%-20%	0.954	0.959	0.963	10%-20%	30.92	31.36	31.88	10%-20%	0.0123	0.0113	0.0106
20%-30%	0.913	0.920	0.927	20%-30%	27.93	28.27	28.68	20%-30%	0.0208	0.0194	0.0184
30%-40%	0.865	0.874	0.882	30%-40%	25.91	26.17	26.49	30%-40%	0.0298	0.0282	0.0269
40%-50%	0.808	0.818	0.827	40%-50%	24.27	24.49	24.73	40%-50%	0.0400	0.0382	0.0369
50%-60%	0.698	0.707	0.716	50%-60%	22.04	22.19	22.31	50%-60%	0.0573	0.0556	0.0545

Table 5: Full comparisons of modules. The tests are conducted in Paris Street View dataset and the percentage in the first column means the ratio of the mask. P-UNet* is P-UNet with changed hyper-parameter. PD means partial-deconvolution. VSR means the VSR layer.

Figure 8: More result from Paris Street View and CelebA datasets (which is omitted in Section 5), from left to right: Ground truth, masked image, recoverd edge, recovered image