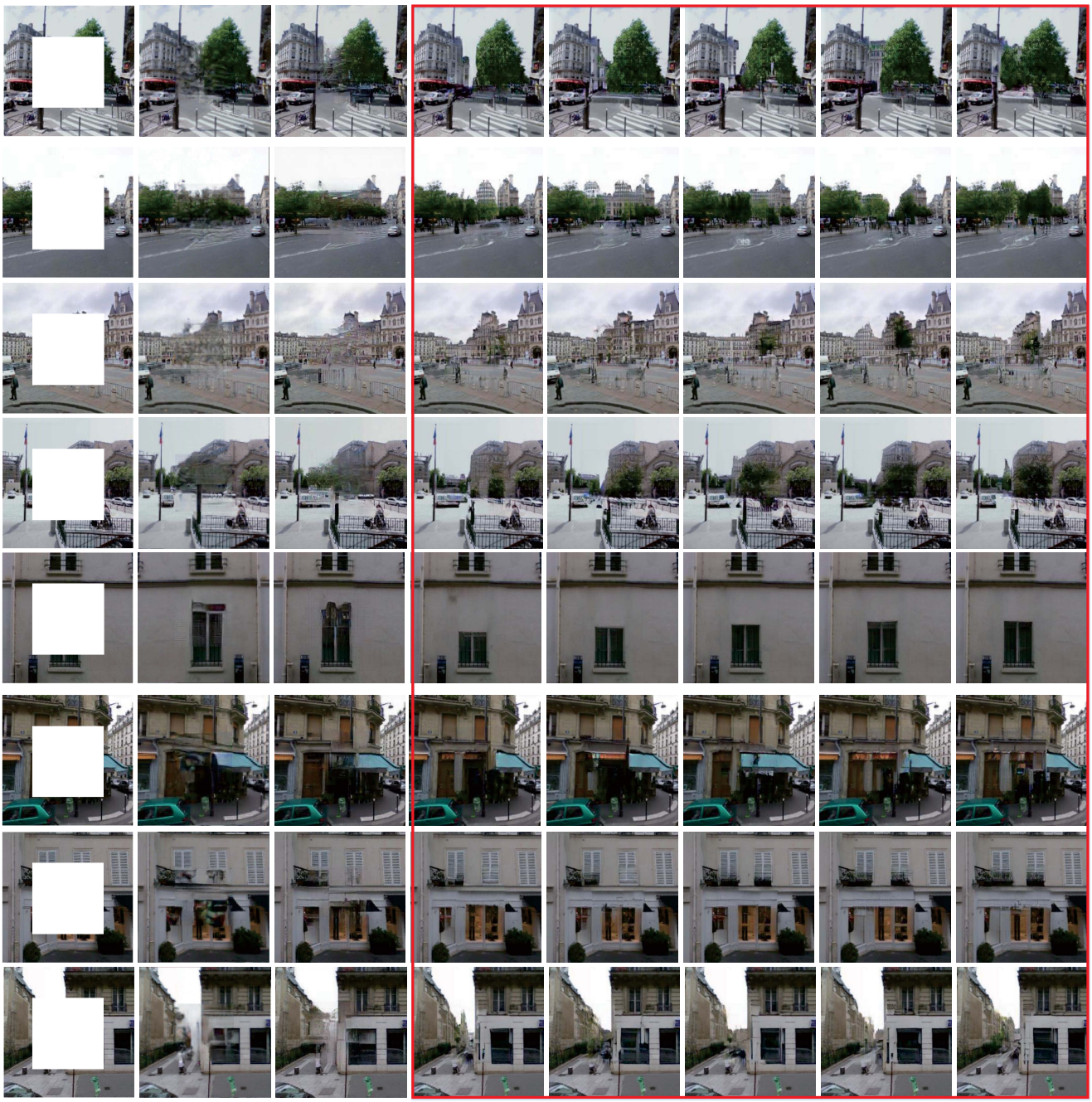
A. Additional Examples

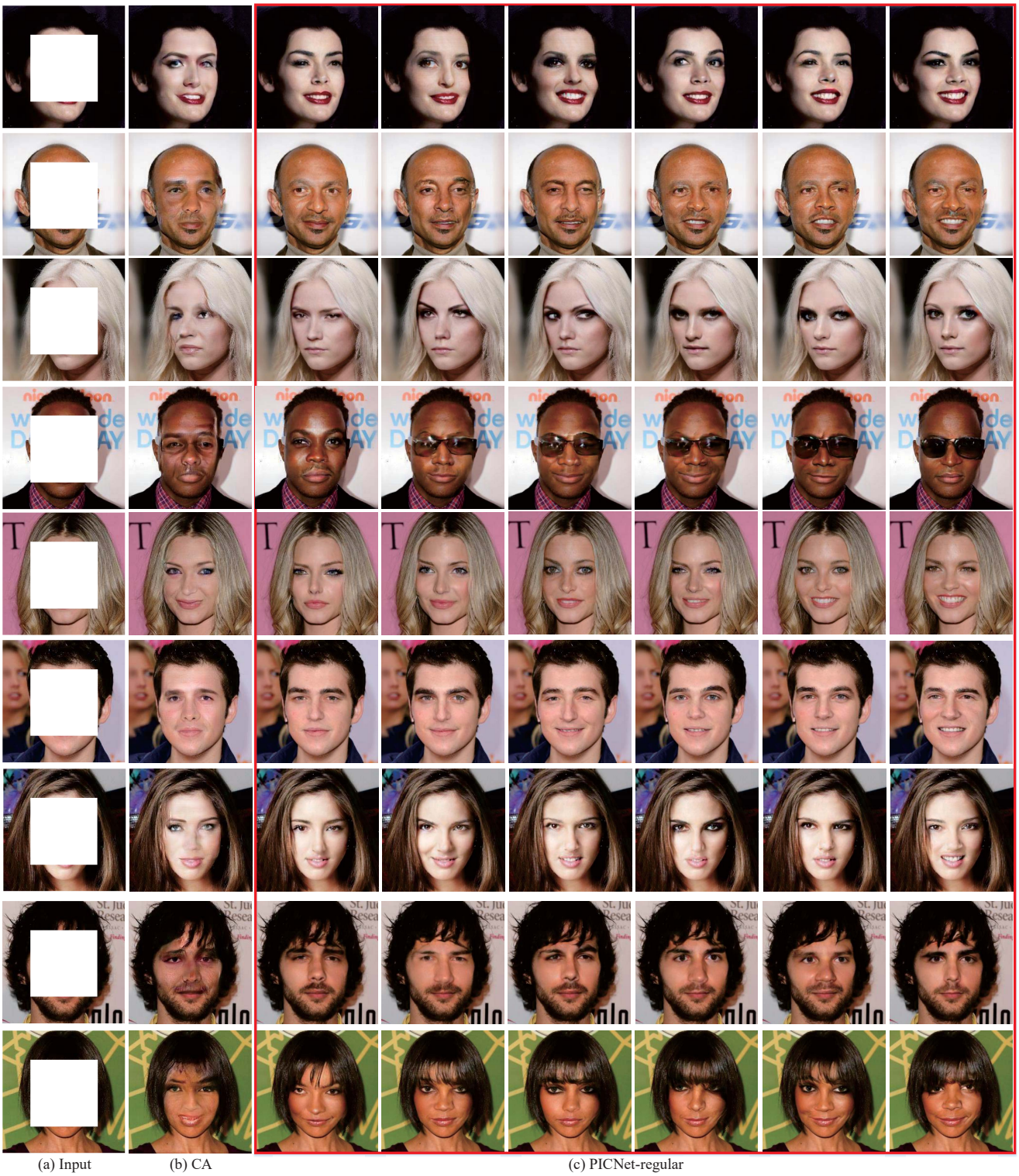
We first show our results on center hole completion, in relation to those from other methods trained on corresponding datasets. As for random irregular and regular holes, we simply present our results so that readers may appreciate the multiple diverse results we can get with differently sized and shaped holes. Finally, we show the interesting application on face editing.

A.1. Comparison with Existing Work on Center Hole Completion



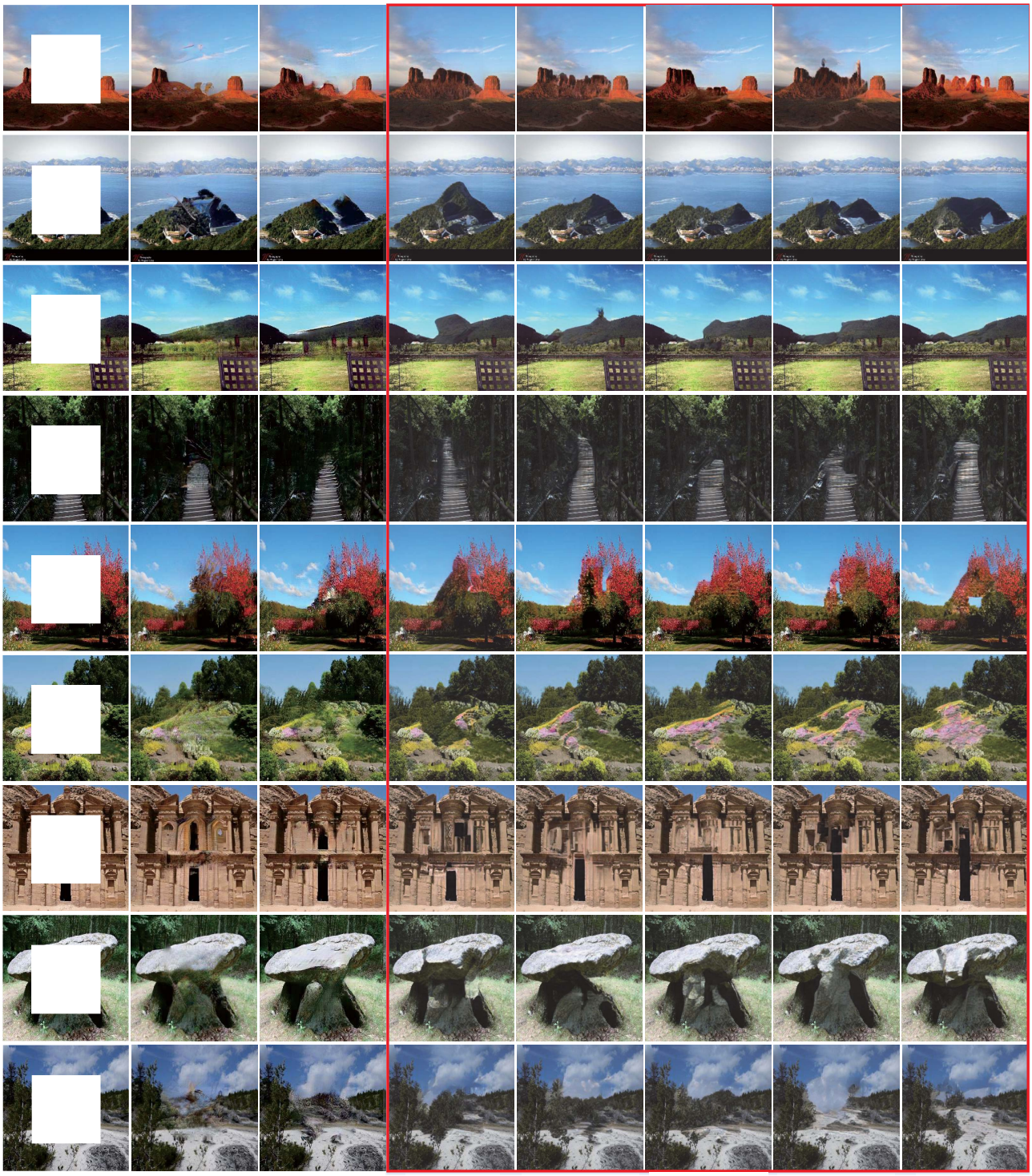
(a) Input (b) CE (c) Shift-Net (d) PICNet-regular

Figure A.1. Additional results on the Paris variation set for center hole completion. This variation dataset contains 100 images, for which we obtained generally more realistic results than the existing methods of CE and Shift-Net. Furthermore, our multiple results had a diverse range of sizes, shapes, colors and textures. Best viewed by zooming in.



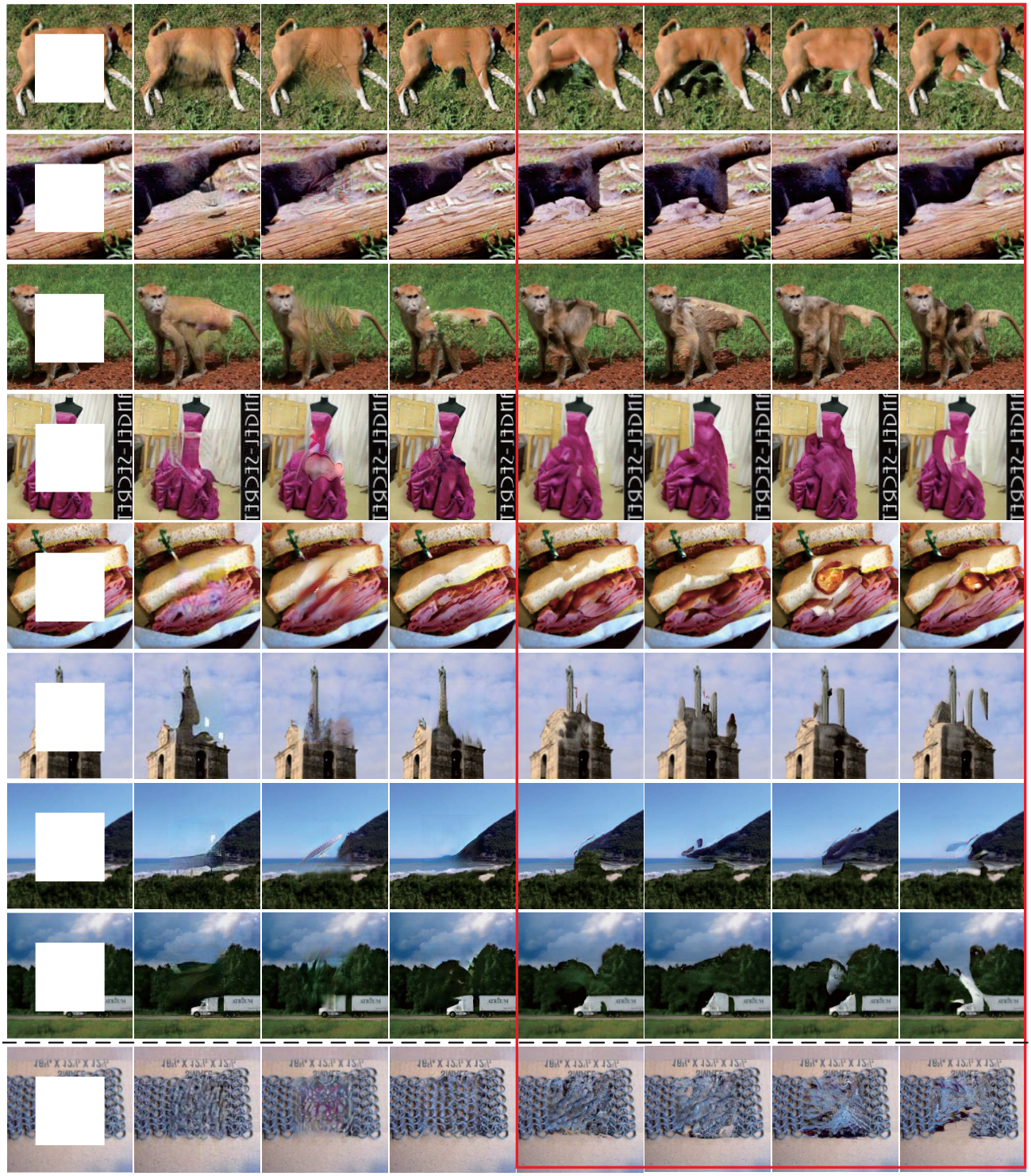
(a) Input (b) CA (c) PICNet-regular

Figure A.2. Additional results on the CelebA-HQ test set for center hole completion. Examples have different genders, skin tones, views and partial visible expressions. Since the occluded content in the large center holes was not repeated in visible regions, CA was unable to create results that were as visually realistic as ours. Moreover, our multiple outputs have different shapes, sizes and colors for eyes, noses and mouths. The details can be viewed by zooming in. Note that, no any other attribute labels (*e.g.* smile) were applied in our approach.



(a) Input (b) GL (c) CA (d) PICNet-regular

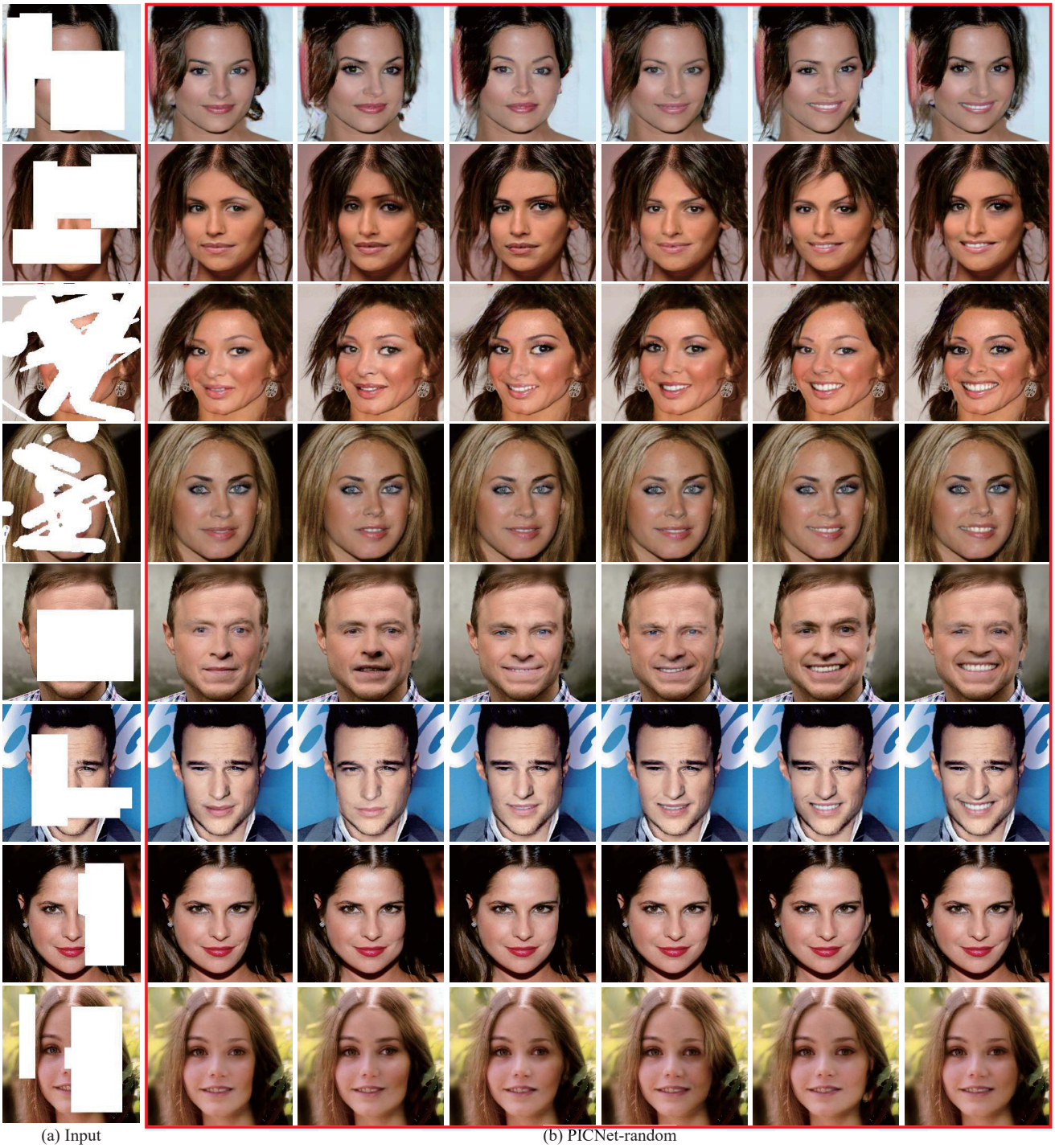
Figure A.3. Additional results on the Places2 variation set for center hole completion. Compared with existing state-of-the-art methods, our model not only generated completion results of comparable quality, but also provided multiple plausible results, with different shapes, colors, textures and content. The shape variations in generating the various prominent hills are obvious. Some changes were at finer scale, *e.g*. color changes of the flowers and texture changes in the boulder are better viewed by zooming in.



(a) Input (b) CE (c) GL (d) CA (e) PICNet-regular

Figure A.4. Additional results for center holes on the ImageNet variation set used in Context Encoder (CE). For our results, four completed images were selected and included failure cases in the last column. The first four rows show examples in which our model generated more visually realistic results than other methods. The next four rows show examples in which the methods all performed with similar realism, while the final row shows an example in which the Context Attention (CA) had the most realistic result.

A.2. Additional Results on Random and Irregular Hole Completion



(a) Input (b) PICNet-random

Figure A.5. Additional results on the CelebA-HQ test set for random and irregular hole completion. One interesting observation is that natural facial symmetry exerts a strong constraint on the completion results. In the examples where both eyes and/or mouth are masked out, the completion results exhibit substantial variation for those facial features when sampled. However, when only one eye is masked out or half of the mouth is visible (last three rows), the completion results for the other eye or the other half of mouth have little variation when sampled. Even when part of an eye is visible (fourth row), it exerts a strong constraint on the variation.

A.3. Additional Results on Free-Form Mask Using Our Interactive Demo

Figure A.6. Face image editing results from our online interactive demo. The white mask regions will be normalized to gray mask as the input. It shows that our PICNet can be used to object removal and face editing.

A.4. Video for Additional Results

Besides this document, we also included two video clips of additional results as part of the supplemental material. The first video, shows free-from mask results on various datasets. The second video consists of four parts to show multiple examples of center hole completion, random hole completion, comparison results with different training strategies and face editing of my self-portraits.

B. Mathematical Derivation and Analysis

B.1. Difficulties with Using the Classical CVAE for Image Completion

Here we elaborate on the difficulties encountered when using the classical CVAE formulation for pluralistic image completion, expanding on the shorter description in section 3.1.

B.1.1 Background: Derivation of the Conditional Variational Auto-Encoder (CVAE)

The broad CVAE framework of Sohn *et al*. [34] is a straightforward conditioning of the classical VAE. Using the notation in our main paper, a latent variable $\mathbf{z}_c$ is assumed to stochastically generate the hidden partial image $\mathbf{I}_c$. When conditioned on the visible partial image $\mathbf{I}_m$, we get the conditional probability:

$$p(\mathbf{I}_c|\mathbf{I}_m) = \int p_\phi(\mathbf{z}_c|\mathbf{I}_m) p_\theta(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m) d\mathbf{z}_c \tag{B.1}$$

The variance of the Monte Carlo estimate can be reduced by importance sampling to get

$$\begin{aligned} p(\mathbf{I}_c|\mathbf{I}_m) &= \int q_\psi(\mathbf{z}_c|\mathbf{I}_c, \mathbf{I}_m) \frac{p_\phi(\mathbf{z}_c|\mathbf{I}_m)}{q_\psi(\mathbf{z}_c|\mathbf{I}_c, \mathbf{I}_m)} p_\theta(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m) d\mathbf{z}_c \\ &= \mathbb{E}_{\mathbf{z}_c \sim q_\psi(\mathbf{z}_c|\mathbf{I}_c, \mathbf{I}_m)} \left[\frac{p_\phi(\mathbf{z}_c|\mathbf{I}_m)}{q_\psi(\mathbf{z}_c|\mathbf{I}_c, \mathbf{I}_m)} p_\theta(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m) \right] \end{aligned} \tag{B.2}$$

Taking logs and apply Jensen's inequality leads to

$$\begin{aligned} \log p(\mathbf{I}_c|\mathbf{I}_m) &\geq \mathbb{E}_{\mathbf{z}_c \sim q_\psi(\mathbf{z}_c|\mathbf{I}_c, \mathbf{I}_m)} \left[\log p_\theta(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m) - \log \frac{q_\psi(\mathbf{z}_c|\mathbf{I}_c, \mathbf{I}_m)}{p_\phi(\mathbf{z}_c|\mathbf{I}_m)} \right] \\ \mathcal{V} &= \mathbb{E}_{\mathbf{z}_c \sim q_\psi(\mathbf{z}_c|\mathbf{I}_c, \mathbf{I}_m)} \left[\log p_\theta(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m) \right] - \mathrm{KL}\left(q_\psi(\mathbf{z}_c|\mathbf{I}_c, \mathbf{I}_m) || p_\phi(\mathbf{z}_c|\mathbf{I}_m) \right) \end{aligned} \tag{B.3}$$

The variational lower bound $\mathcal{V}$ totaled over all training data is jointly maximized w.r.t. the network parameters θ, ϕ and ψ in attempting to maximize the total log likelihood of the observed training instances.

B.1.2 Single Instance Per Conditioning Label

As is typically the case for image completion, there is only one training instance of $\mathbf{I}_c$ for each unique $\mathbf{I}_m$. This means that for the function $q_\psi(\mathbf{z}_c|\mathbf{I}_c, \mathbf{I}_m)$, $\mathbf{I}_c$ can simply be learnt into the network as a hardcoded dependency of the input $\mathbf{I}_m$, so $q_\psi(\mathbf{z}_c|\mathbf{I}_c, \mathbf{I}_m) \cong \hat{q}_\psi(\mathbf{z}_c|\mathbf{I}_m)$. Assuming that the network for $p_\phi(\mathbf{z}_c|\mathbf{I}_m)$ has similar or higher modeling power and there are no other explicit constraints imposed on it, then in training $p_\phi(\mathbf{z}_c|\mathbf{I}_m) \to \hat{q}_\psi(\mathbf{z}_c|\mathbf{I}_m)$, and the KL divergence in (B.3) goes to zero.

In this situation of zero KL divergence, we can rewrite the variational lower bound and replace $\hat{q}_\psi(\mathbf{z}_c|\mathbf{I}_m)$ with $p_\phi(\mathbf{z}_c|\mathbf{I}_m)$ without loss of generality, as

$$\mathcal{V} = \mathbb{E}_{\mathbf{z}_c \sim p_\phi(\mathbf{z}_c|\mathbf{I}_m)} \left[\log p_\theta(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m) \right] \tag{B.4}$$

B.1.3 Unconstrained Learning of the Conditional Prior

We can analyze how$\mathcal{V}$ can be maximized, by using Jensen's inequality again (reversing earlier use)

$$\begin{aligned} \mathcal{V} &\leq \log \mathbb{E}_{\mathbf{z}_c \sim p_\phi(\mathbf{z}_c|\mathbf{I}_m)} \left[p_\theta(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m) \right] \\ &= \log \int p_\phi(\mathbf{z}_c|\mathbf{I}_m) p_\theta(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m) d\mathbf{z}_c \end{aligned} \tag{B.5}$$

By further applying Hölder's inequality (*i.e.* $\|fg\|_1 \leq \|f\|_p \|g\|_q$ for $1/p + 1/q = 1$), we get

$$\begin{aligned} \mathcal{V} &\leq \log \left[\left| \int |p_\phi(\mathbf{z}_c|\mathbf{I}_m)| \, d\mathbf{z}_c \right| \left| \int |p_\theta(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m)|^\infty \, d\mathbf{z}_c \right|^{\frac{1}{\infty}} \right] \quad \text{(by setting } p = 1, q = \infty) \\ &= \log \left[1 \cdot \max_{\mathbf{z}_c} p_\theta(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m) \right] = \max_{\mathbf{z}_c} \log p_\theta(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m) \end{aligned} \tag{B.6}$$

Assuming that there is a unique global maximum for $\log p_\phi(\mathbf{z}_c|\mathbf{I}_m)$, the bound achieves equality when the conditional prior becomes a Dirac delta function centered at the maximum latent likelihood point

$$p_\phi(\mathbf{z}_c|\mathbf{I}_m) \rightarrow \delta(\mathbf{z}_c - \mathbf{z}_c^*) \quad \text{where } \mathbf{z}_c^* = \arg\max_{\mathbf{z}_c} \log p_\theta(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m) \tag{B.7}$$

Intuitively, subject to the vagaries of stochastic gradient descent, the network for $p_\phi(\mathbf{z}_c|\mathbf{I}_m)$ without further constraints will learn a narrow delta-like function that sifts out maximum latent likelihood value of $\log p_\theta(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m)$.

As mentioned in section 3.1, although this narrow conditional prior may be helpful in estimating a single solution for $\mathbf{I}_c$ given $\mathbf{I}_m$ during testing during testing, this is poor for sampling a diversity of solutions. In our framework, the (unconditional) latent priors are imposed for the partial images themselves, which prevent this delta function degeneracy.

B.1.4 CVAE with Fixed Prior

An alternative CVAE variant [37] assumes that conditional prior is independent of the $\mathbf{I}_m$ and fixed, so $p(\mathbf{z}_c|\mathbf{I}_m) \cong p(\mathbf{z}_c)$, where $p(\mathbf{z}_c)$ is a fixed distribution (*e.g.* standard normal). This means

$$p(\mathbf{I}_c|\mathbf{I}_m) = \int p(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m) p(\mathbf{z}_c) d\mathbf{z}_c \tag{B.8}$$

Now we can consider the case for a fixed $\mathbf{I}_m = \mathbf{I}_m^*$, and rewrite (B.8) as

$$p_{\mathbf{I}_m^*}(\mathbf{I}_c) = \int p_{\mathbf{I}_m^*}(\mathbf{I}_c|\mathbf{z}_c) p(\mathbf{z}_c) d\mathbf{z}_c \tag{B.9}$$

Doing so makes it obvious we can then derive the standard (unconditional) VAE formulation from here. Thus an appropriate interpretation of this CVAE variant is that it uses $\mathbf{I}_m$ as a "switch" parameeter to choose between different VAE models that are trained for the specific conditions.

Once again, this is fine if there are multiple training instances per conditional label. However, in the image completion problem, there is only one $\mathbf{I}_c$ per unique $\mathbf{I}_m$, so the condition-specific VAE model will simply ignore the sampling "noise" and learn to predict the single instance of $\mathbf{I}_c$ from $\mathbf{I}_m$ directly, *i.e.* $p(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m) \approx p(\mathbf{I}_c|\mathbf{I}_m)$, which incidentally achieves equality for the variational lower bound. This results in negligible variation of output despite now sampling from $p(\mathbf{z}_c) = \mathcal{N}(0, 1)$.

Our framework resolves this in part by defining all (unconditional) partial images of $\mathbf{I}_c$ as sharing a common latent space with adaptive priors, with the likelihood parameters learned as an unconditional VAE, and further coupling on the conditional portion (*i.e.* the generative path) to get a more distinct but regularized estimate for $p(\mathbf{z}_c|\mathbf{I}_m)$.

B.2. Joint Maximization of Unconditional and Conditional Variational Lower Bounds

The overall training loss function (5) used in our framework has a direct link to jointly maximizing the unconditional and unconditional variational lower bounds, respectively expressed by (2) and (4). Using simplified notation, we rewrite these bounds respectively as:

$$\begin{aligned} \mathcal{B}_1 &= \mathbb{E}_{q_\psi} \log p_\theta^r - \mathrm{KL}(q_\psi||p_{z_c}) \\ \mathcal{B}_2 &= \lambda \left(\mathbb{E}_{q_\psi} \log p_\theta^r - \mathrm{KL}(q_\psi||p_{z_c}) \right) + \mathbb{E}_{p_\phi} \log p_\theta^g \end{aligned} \tag{B.10}$$

To clarify, $\mathcal{B}_1$ is the lower bound related to the unconditional log likelihood of observing $\mathbf{I}_c$, while $\mathcal{B}_2$ relates to the log likelihood of observing $\mathbf{I}_c$ conditioned on $\mathbf{I}_m$. The expression of $\mathcal{B}_2$ reflects a blend of conditional likelihood formulations with and without the use of importance sampling, which are matched to different likelihood models, as explained in section 3.1. Note that the $(1 - \lambda)$ coefficient from (4) is left out here for simplicity, but there is no loss of generality since we can ignore a constant factor of the true lower bound if we are simply maximizing it.

We can then define a combined objective function as our maximization goal

$$\begin{aligned} \mathcal{B} &= \beta\, \mathcal{B}_1 + \mathcal{B}_2 \\ &= (\beta + \lambda) \mathbb{E}_{q_\psi} \log p_\theta^r + \mathbb{E}_{p_\phi} \log p_\theta^g - [\beta \mathrm{KL}(q_\psi||p_{z_c}) + \lambda \mathrm{KL}(q_\psi||p_\phi)] \end{aligned} \tag{B.11}$$

with $\beta \geq 0$.

To understand the relation between $\mathcal{B}$ in (B.11) and $\mathcal{L}$ in (5), we consider the equivalence of:

$$-\mathcal{B} \cong \mathcal{L} = \alpha_{\mathrm{KL}}(\mathcal{L}^r_{\mathrm{KL}} + \mathcal{L}^g_{\mathrm{KL}}) + \alpha_{\mathrm{app}}(\mathcal{L}^r_{\mathrm{app}} + \mathcal{L}^g_{\mathrm{app}}) + \alpha_{\mathrm{ad}}(\mathcal{L}^r_{\mathrm{ad}} + \mathcal{L}^g_{\mathrm{ad}}) \tag{B.12}$$

Comparing terms

$$\mathcal{L}^r_{\mathrm{KL}} \cong \mathrm{KL}(q_\psi || p_{z_c}), \quad \mathcal{L}^g_{\mathrm{KL}} \cong \mathrm{KL}(q_\psi || p_\phi) \quad \Rightarrow \beta = \lambda = \alpha_{\mathrm{KL}} \tag{B.13}$$

For the reconstructive path that involves sampling from the (posterior) importance function $q_\psi(\mathbf{z}_c|\mathbf{I}_c)$ of (3), we can substitute $(\beta + \lambda) = 2\alpha_{\mathrm{KL}}$ and get the reconstructive log likelihood formulation as

$$-\mathbb{E}_{q_\psi} \log p^r_\theta \cong \frac{\alpha_{\mathrm{app}}}{2\alpha_{\mathrm{KL}}}\mathcal{L}^r_{\mathrm{app}} + \frac{\alpha_{\mathrm{ad}}}{2\alpha_{\mathrm{KL}}}\mathcal{L}^r_{\mathrm{ad}} \tag{B.14}$$

Here, $\mathbf{I}_c$ is available, with $\mathcal{L}^r_{\mathrm{app}}$ reconstructing both $\mathbf{I}_c$ and $\mathbf{I}_m$ as in (8), while $\mathcal{L}^r_{\mathrm{ad}}$ involves GAN-based pairwise feature matching (10).

For the generative path that involves sampling from the conditional prior $p_\phi(\mathbf{z}_c|\mathbf{I}_m)$, we have the generative log likelihood formulation as

$$-\mathbb{E}_{p_\phi} \log p^g_\theta \cong \alpha_{\mathrm{app}}\mathcal{L}^g_{\mathrm{app}} + \alpha_{\mathrm{ad}}\mathcal{L}^g_{\mathrm{ad}} \tag{B.15}$$

As explained in sections 3.1 and 3.2, the generative path does not have direct access to $\mathbf{I}_c$, and this is reflected in the likelihood p^g_θ in which the instances of $\mathbf{I}_c$ are ignored. Thus $\mathcal{L}^g_{\mathrm{app}}$ is only for reconstructing $\mathbf{I}_m$ in a deterministic auto-encoder fashion as per (9), while $\mathcal{L}^g_{\mathrm{ad}}$ in (11) only tries to enforce that the generated distribution be consistent with the training set distribution (hence without per-instance knowledge), as implemented in the form of a GAN.

C. Architectural Details

Our **pluralistic image completion** network (**PICNet**) architecture is inspired by SA-GAN [43] and BigGAN, but features several important modifications that enable us to train for this image-conditional generation task. We first replace the batch normalization with instance normalization in the generation network (**ResBlock up** in Fig. C.7), and remove the batch normalization in our other networks, (*i.e.* the representation, inference and discriminator networks comprising **ResBlock start** and **ResBlock** in Fig. C.7), because different holes will affect the means and variances in each batch. **ResBlock down** is similar to **ResBlock**, in which we add the average pooling layer after Conv3 $\times$ 3 and Conv1 $\times$ 1.

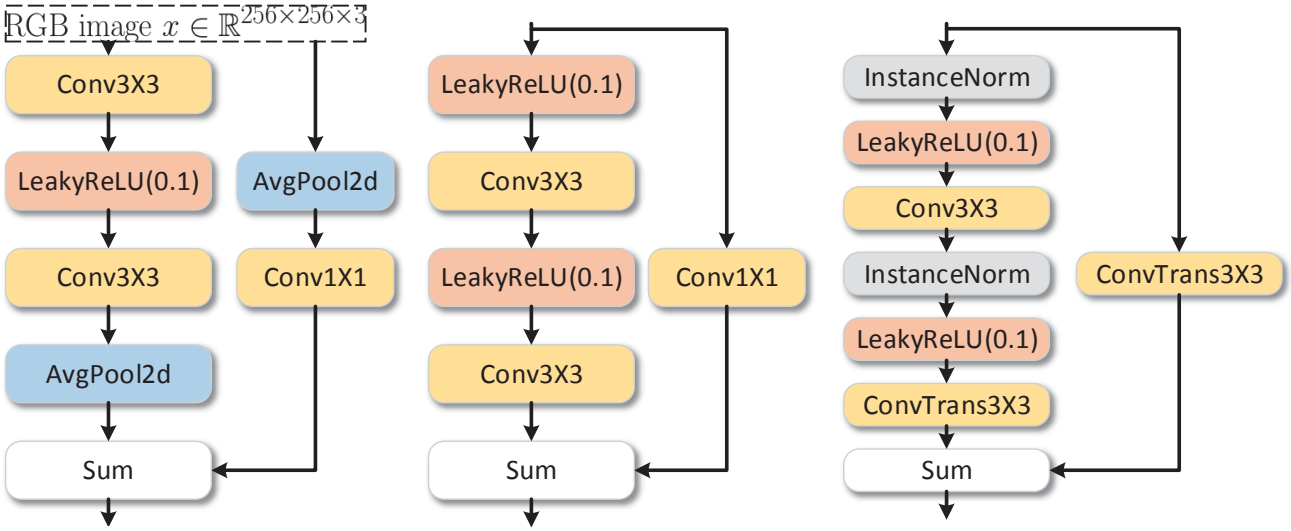
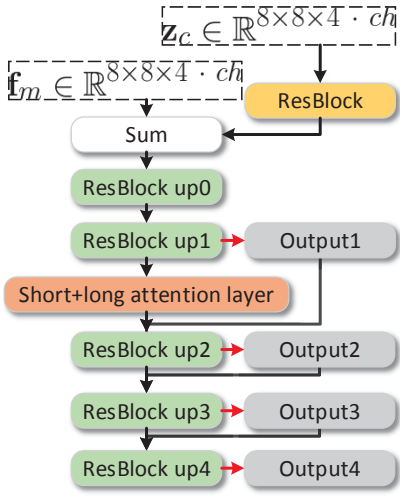


Figure C.7. Illustration of the Residual Block used in our model. (a) The starter Residual Block for the encoder (representation) and discriminator networks. (b) A Residual Block in the encoder (representation), inference and discriminator networks. (c) A Residual Block in the decoder (generator) network.



(a) Decoder (Generator)

RGB image $x \in \mathbb{R}^{256\times256\times3}$
ResBlock start $128 \times 128 \times 1 \cdot ch$
ResBlock down $64 \times 64 \times 2 \cdot ch$
ResBlock down $32 \times 32 \times 4 \cdot ch$
ResBlock down $16 \times 16 \times 4 \cdot ch$
ResBlock down $8 \times 8 \times 4 \cdot ch$

(b) Encoder (Representation)

RGB image $x \in \mathbb{R}^{256\times256\times3}$
ResBlock start $128 \times 128 \times 1 \cdot ch$
ResBlock down $64 \times 64 \times 2 \cdot ch$
ResBlock down $32 \times 32 \times 4 \cdot ch$
Self-Attention Layer $32 \times 32 \times 4 \cdot ch$
ResBlock down $16 \times 16 \times 4 \cdot ch$
ResBlock down $8 \times 8 \times 4 \cdot ch$
ResBlock $7 \times 7 \times 4 \cdot ch$
LeakyReLU(0.1), Conv, $6 \times 6 \times 1$

(c) Discriminator

Table C.1. Architectures for our framework, where ch represents the base channel width. For the output layer, we use the LeakyReLU(0.1), Conv3 $\times$ 3 and Tanh at all scales.

The **Infer1** network only consists of one Residual Block, for self-inferring the latent distribution of the ground truth $\mathbf{I}_c$ (treated as known in the reconstructive path), while the **Infer2** network consists of seven Residual Blocks, which are applied to predict the latent distribution of $\mathbf{I}_c$ (treated as unknown in the generative path) based on the visible pixels $\mathbf{I}_m$.

D. Experimental Details

Our network is implemented in Pytorch v0.4.0, and employs the architectures of Appendix C. To reduce memory cost, we restrained the feature channel width to $4 \cdot ch$ and selected $ch = 32$. We experimented with different channels with largest being $16 \cdot ch = 1024$, but found that the improvement was not obvious. In addition, we applied the self-attention layer of the discriminator and the short+long term attention layer of the generator on a 32×32 feature size. Spectral Normalization is used in all networks. All networks are initialized with Orthogonal Initialization and trained from scratch with a fixed learning rate of $\lambda = 10^{-4}$. We used the Adam optimizer with $\beta_1 = 0$ and $\beta_2 = 0.999$.

The final weights we used were $\alpha_{\text{KL}} = \alpha_{\text{app}}$=20, α_{ad}=1. The KL loss and appearance matching loss weights come from the variational *lower bound*. Since the appearance matching loss is used in four output scales, the final weight for the KL loss is $\alpha_{\text{KL}} = \alpha_{\text{KL}} \times N_{\text{scale}}$, where N_{scale} is the number of output scales. We also tried different values of α_{KL} and α_{app}, and found that the bigger the KL loss weight, the greater the diversity of the generated $\mathbf{I}_c^{'}$, but it was also harder to retain the appearance consistency of the generated $\mathbf{I}_c^{'}$ to the visible region $\mathbf{I}_m$. The values of α_{app} and α_{ad} were obtained from α-GAN. We experimented with the number of D steps per G step (varying it from 1 to 5), and found that one D step per G step gave the best results. When α_{app} is smaller than 1, we can use two or four D steps per G step, but the full generated $\mathbf{I}_g^{'}$ does not reconstruct the original conditional visible regions $\mathbf{I}_m$ well. When α_{app} is larger than 100, we needed two or four G steps per D step, if not the discriminator loss will become zero and the generated $\mathbf{I}_c^{'}$ will be blurry.

We trained each model on a single GPU, with a batch size of 20 on a GTX 1080TI (11GB) and 32 on a NVIDIA V100 (16GB). Training models for centered holes of Paris and CelebA-HQ takes roughly 3 days, while for ImageNet and Places2 it takes roughly 2 weeks. On the other hand, training models for random irregular and un-centered holes takes about twice the time compared to models for centered holes. Moreover, since the prior distribution of random holes $p(\mathbf{z}) = \mathcal{N}_m(\mathbf{0}, \sigma^2(n)\mathbf{I})$ is changed with the number of pixels in each hole n, the training loss may sometimes change abruptly due to the KL loss component.