# Bridging Global Context Interactions for High-Fidelity Image Completion Supplementary Material

The supplementary material is organized as follows: in Section A, we first present additional visual results, including results of TFill-*Coarse* on Face datasets (CelebA-HQ [22,33], FFHQ [23]), ImageNet [41] and Places2 datasets [57], qualitative comparison to the state-of-the-art models on various datasets, and some examples for free-form editing on high-resolution images. Next, extending the quantitative comparisons of Tables 2 in the main paper, Section B presents additional evaluation results under the traditional pixel-level and patch-level image quality metrics. Finally, we discuss more technological details of the proposed TFill model in Section C.

## A. Additional Examples

### A.1. Additional Results for TFill-*Coarse*

In Figs. A.1, A.2 and A.3, we show more examples on ImageNet [41], Face datasets (CelebA-HQ [22, 33], FFHQ [23]), and Places2 [57] dataset images that were degraded by large center masks.

Here, all examples shown are chosen from the corresponding testing set. In Fig. A.1, we show examples for object completion, such as the various items and animals on the top half. Fig. A.2 shows visual results of TFill-*Coarse* on face datasets. In Fig. A.3, we display the completed images for various natural scenes. These examples are good evidence that our TFill model is suitable for both *foreground* object completion and *background* scene completion, where it can synthesize semantically consistent content with visually realistic appearance based on the presented visible pixels.

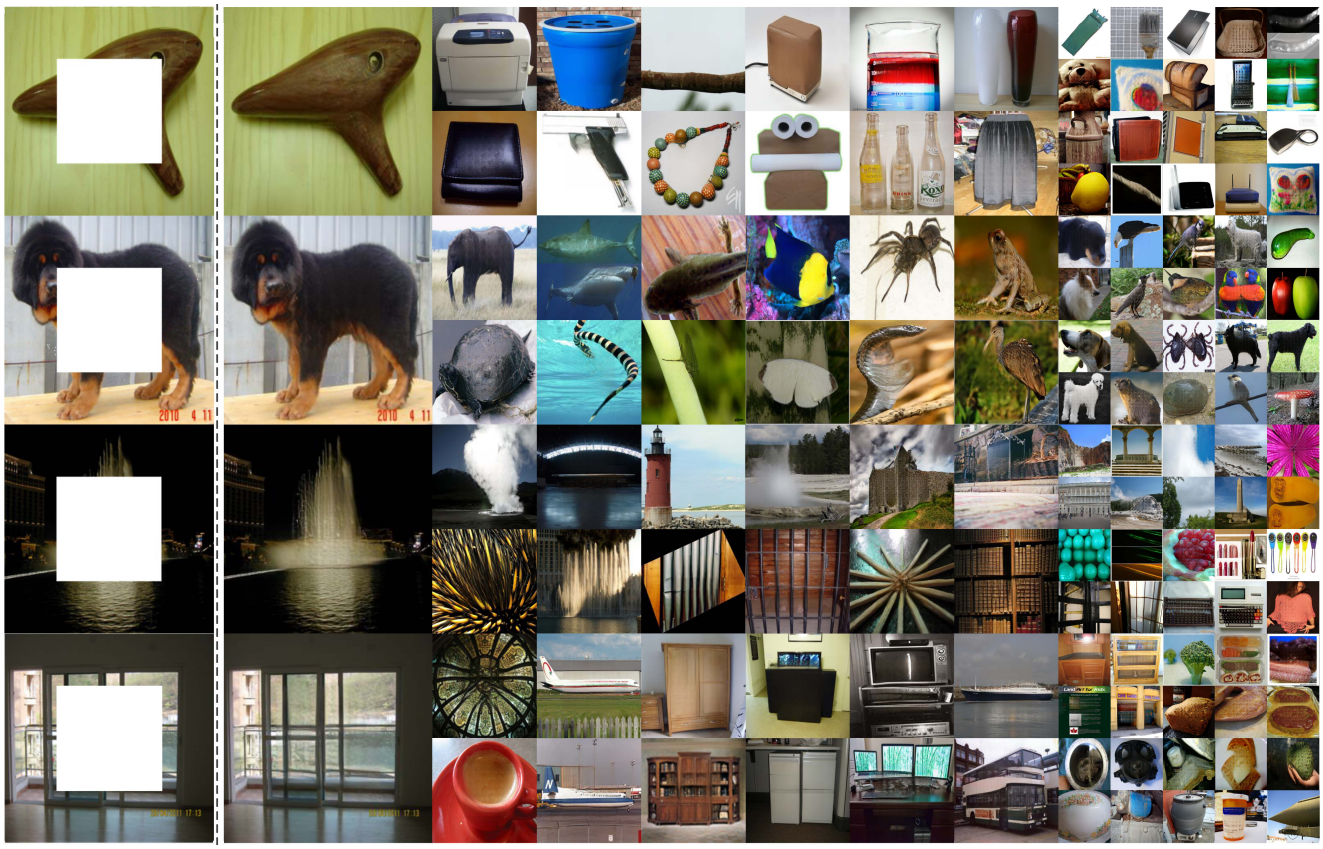


Figure A.1. **Example completion results of our method (config $\mathbb{E}$) on ImageNet datasets [41].** All images come from the corresponding testing set that were **degraded by center masks**. Here, we show results for various categories, such as commodity, animal, plant, natural scene, building, food, furniture and so on. The center masked example inputs are shown on the left. Our model is able to complete both object shape and background scene via a transformer-based architecture to correctly bridge the visible tokens.

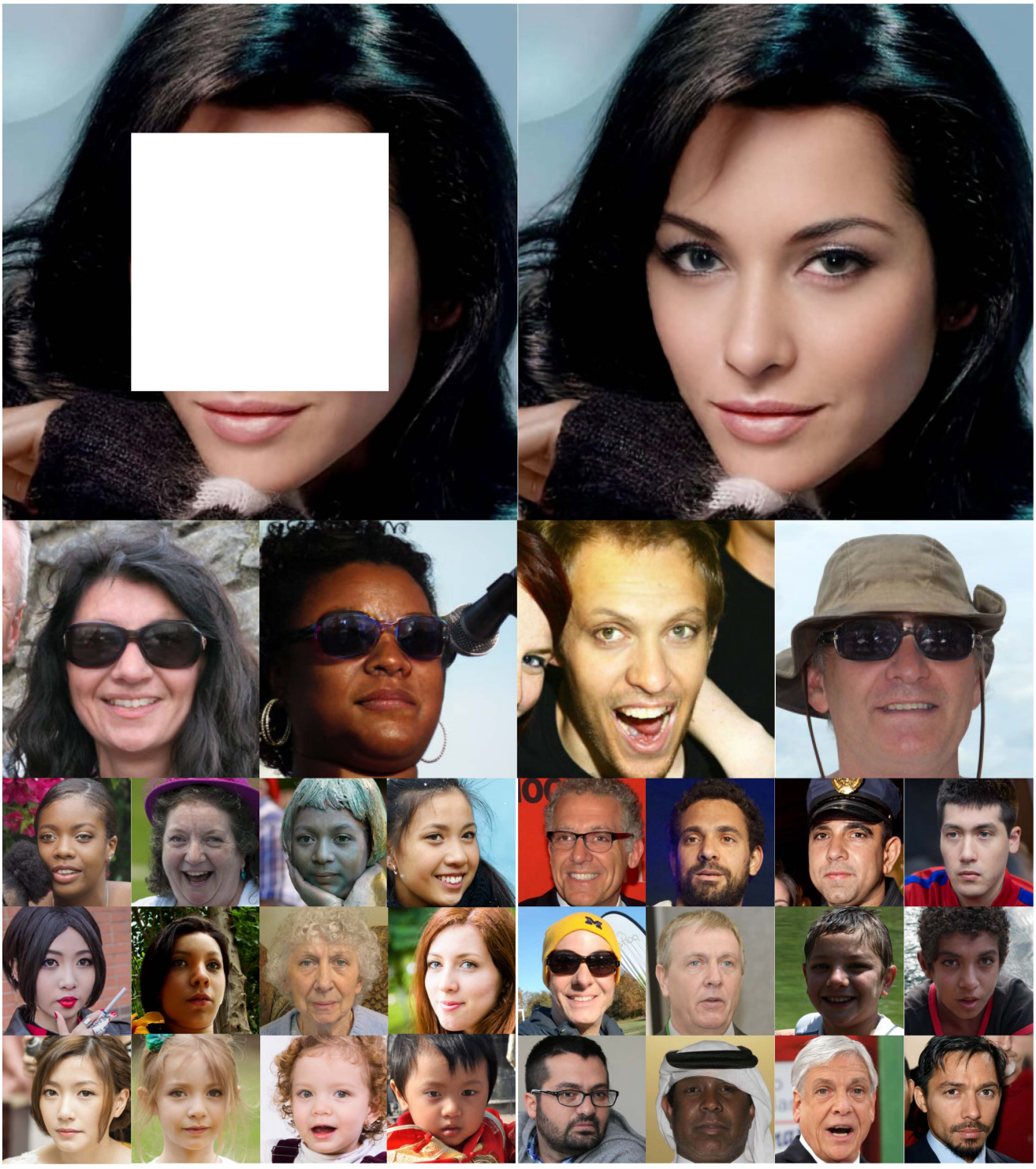


Figure A.2. **Example completion results of our method (config $\mathbb{E}$) on face datasets.** Here, a center mask was used for all input images. One center masked example input is shown top-left. As can be seen, the completed images are on average of high quality. Even for some challenging cases, such as when eyeglasses are center masked, our TFill can correctly repair the face with eyeglasses. Furthermore, it generally works well for varied skin tones, poses, expressions, ages, and illumination.

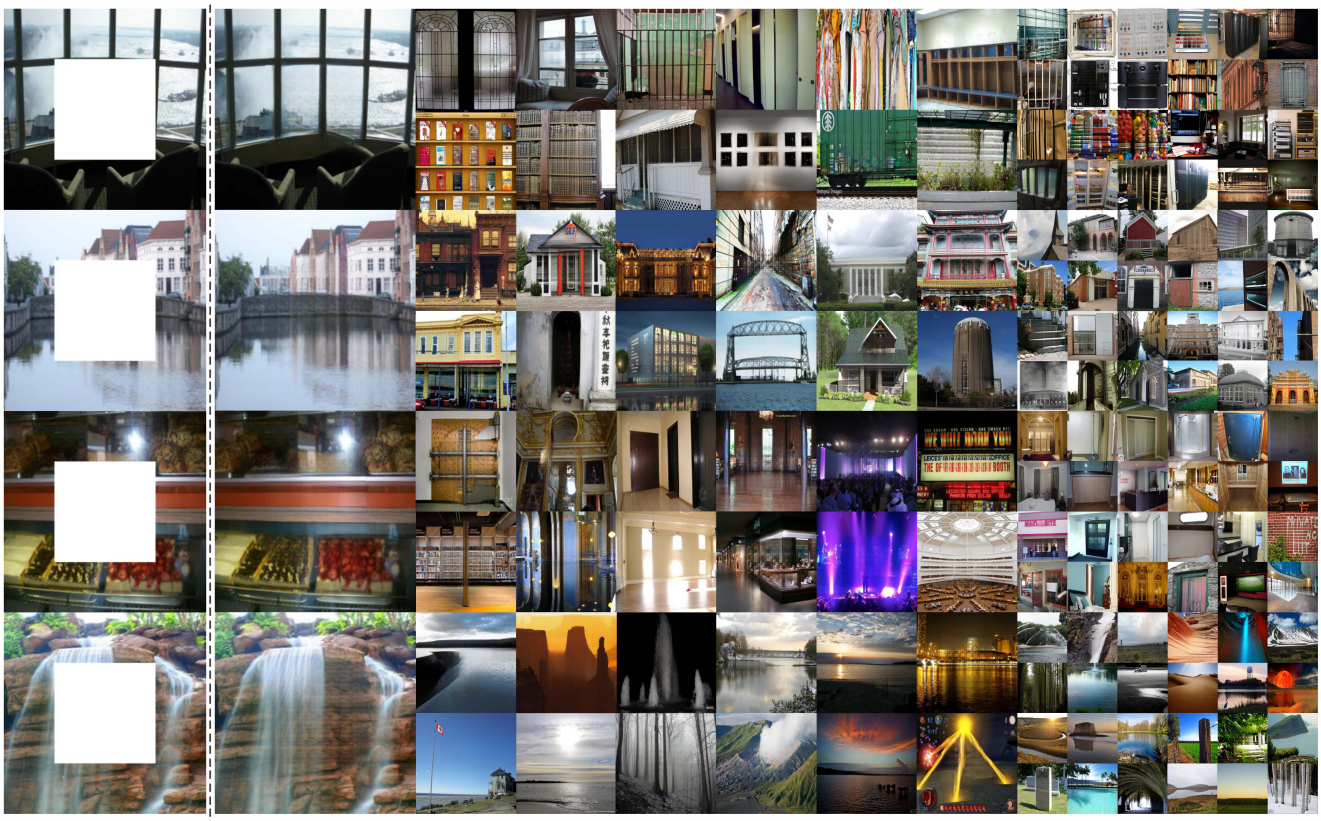


Figure A.3. **Example completion results of our method (config $\mathbb{E}$) on Places2 datasets [57].** Here, we show results for varied scene categories. The center masked example inputs are shown on the left. Our model is able to complete both object shape and background scene via a transformer-based architecture to correctly bridge the visible tokens.

## A.2. Additional Comparisons

In Figs. A.4, A.5 and A.6, we show additional comparison results on various datasets with free-form masks provided in PConv [28]. This is an extension of Figs. 7 and 9 in the main paper. Here, our results on CelebA-HQ [22, 33] and FFHQ [23] testing set are reported for $512 \times 512$ resolution. On the other hand, the results on ImageNet [41] and Places2 [57] are reported for higher resolution images that were resized such that the short side is 512 pixels, with the long side in multiples of $2^5 = 32$, *e.g.* $640 = 32 \times 20$. The size variability is possible due to our fully convolutional encoder-decoder network structures. The 32-base scale is required because our refinement network downsamples the images 5 times with step 2.

As can be seen from these results, our TFill model filled appropriate semantic content with visually realistic appearance into the various masks. For instance, in the third row of Fig. A.4, even with an extensive mask on an obliquely angled face, it was able to generate high-quality results. It achieved good results even under challenging conditions for various objects (Fig. A.5) and scenes (Fig. A.6).

## A.3. Free-Form Editing on High-Resolution Images

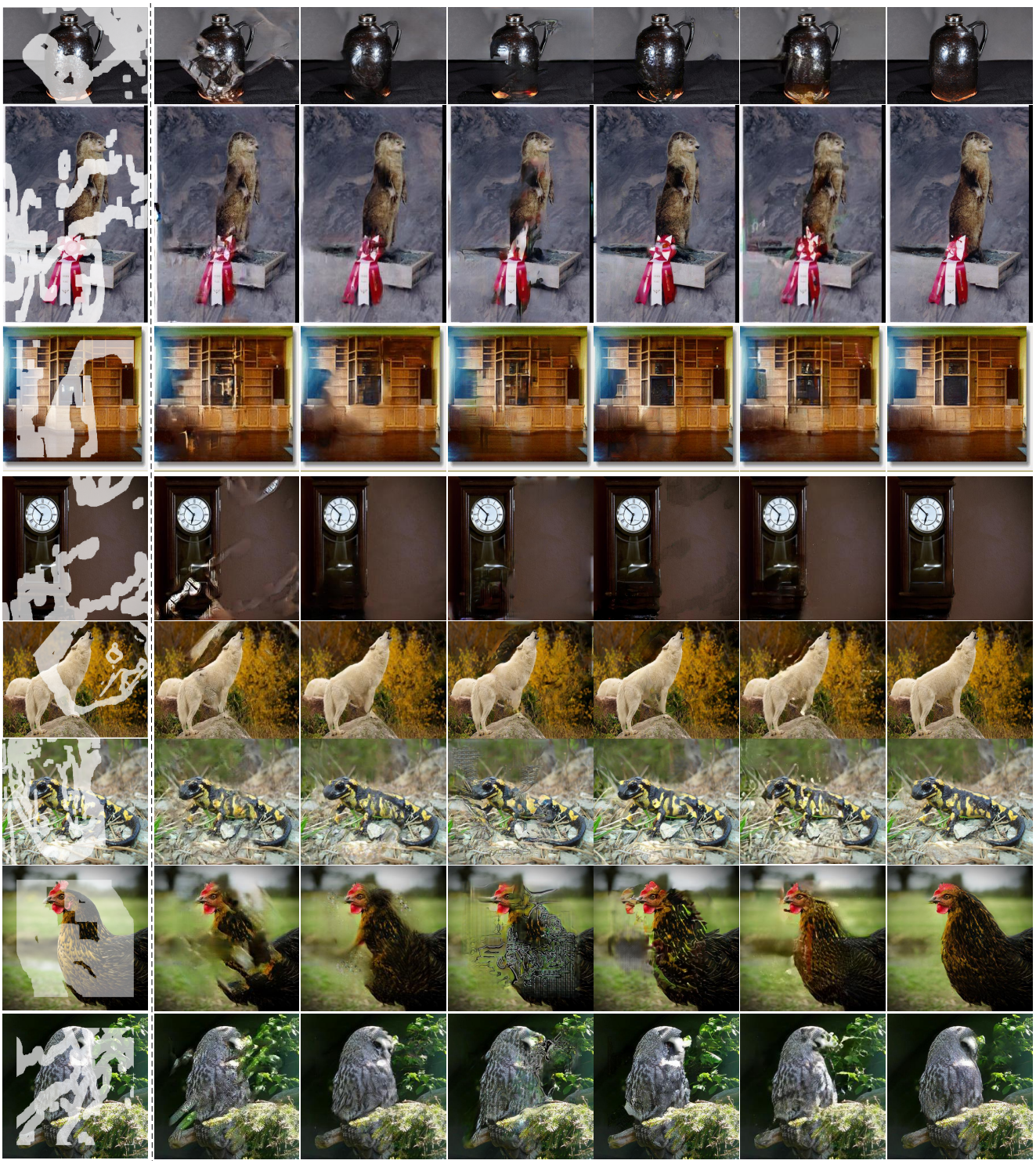
In Figs. A.7, A.8, A.9 and A.10, we show qualitative results for free-form image masking on various higher resolution datasets.

In Fig. A.7, we show some examples for face editing at $512^2$ resolution. For conventional object removal, *e.g.* watermark removal, our TFill addresses them easily. Furthermore, our TFill can handle more extensive face editing, such as removing substantial facial hair and changing mouth expressions.

In Figs. A.8, A.9 and A.10, we show some examples of editing images of natural / outdoor scenes, with object removal being the main task, as it is the main practice for image inpainting. Here, we enforce the input image size to be multiples of 32, *e.g.* $960 \times 640$ and provide the high-resolution results on the corresponding image size. As we can see, our TFill-*Refined* model is able to handle high-resolution images for object removal in traditional image inpainting task.

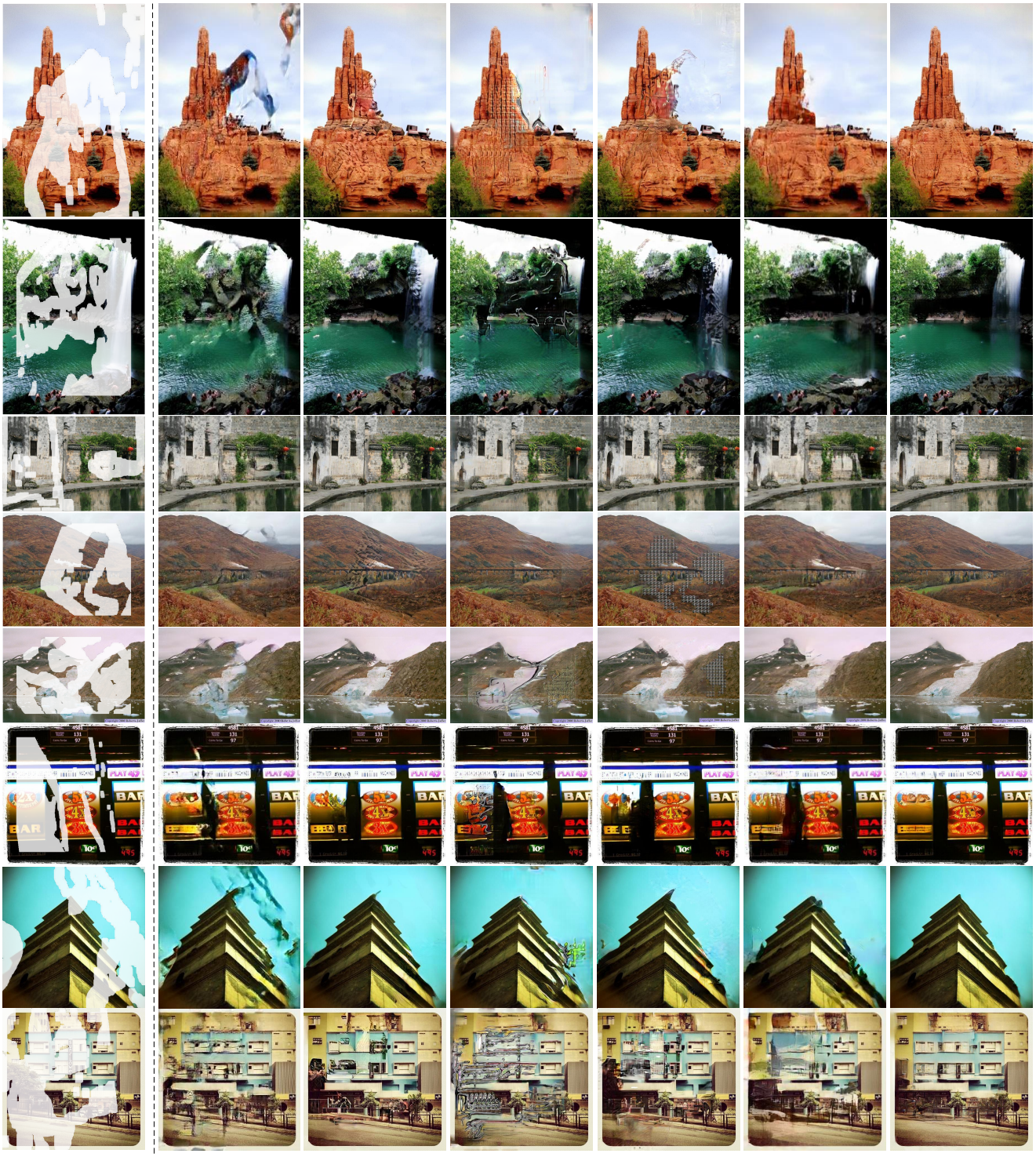
(a) Maksed input (b) CA [52]$_{\text{CVPR'2018}}$ (c) PIC [60]$_{\text{CVPR'2019}}$ (d) ICT [47]$_{\text{ICCV'2021}}$ (e) TFill-*Coarse* (f) TFill-*Refined*

Figure A.4. **Additional results on CelebA-HQ [22, 33] and FFHQ [23] testing set among CA [52], PIC [60], ICT [47] and Ours.** Our results are reported for $512^2$ resolution. While PIC [60] works well for frontal facing faces, it may generate more uncanny faces with mismatched features at larger angles, *e.g*. the examples in third and last row. In contrast, our model generated consistent facial features with photorealistic appearance for various faces angles. As ICT [47] does not use context attention in the second stage to copy information from visible pixels, most of eyes on the completed images are inconsistent. Zoom in to see the details.



(a) Masked input (b) CA [52]$_{\text{CVPR'18}}$ (c) PIC [60]$_{\text{CVPR'19}}$ (d) HiFill [51]$_{\text{CVPR'20}}$ (e) CRFill [54]$_{\text{CVPR'21}}$ (f) ICT [47]$_{\text{ICCV'21}}$ (g) TFill

Figure A.5. **Additional results on ImageNet [41] testing set among CA [52], PIC [60], HiFill [51], CRFill [54], ICT [47] and Ours.** Our results are evaluated in higher resolution, with the short side at 512 pixels and the long side at multiples of $2^5$, *e.g.* 640. Our TFill model generated better visual results even under very challenging situations, *e.g.* the heavily masked chicken in the second last row.

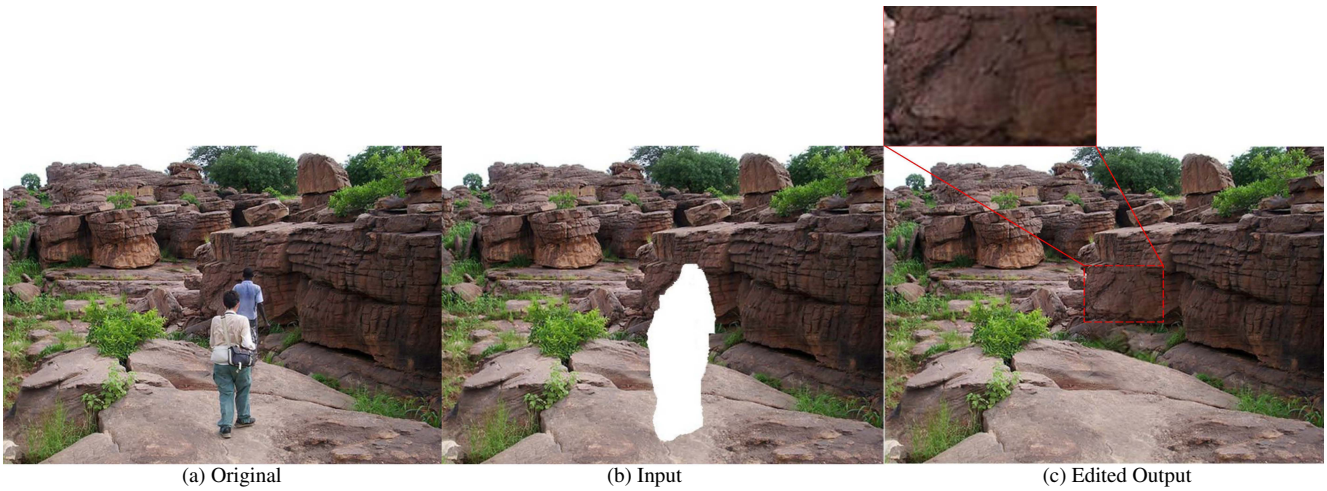


(a) Masked input (b) CA [52]$_{\text{CVPR'18}}$ (c) PIC [60]$_{\text{CVPR'19}}$ (d) HiFill [51]$_{\text{CVPR'20}}$ (e) CRFill [54]$_{\text{CVPR'21}}$ (f) ICT [47]$_{\text{ICCV'21}}$ (g) TFill

Figure A.6. **Additional results on Places2 [57] testing set among CA [52], PIC [60], HiFill [51], CRFill [54], ICT [47] and Ours**. Our results are evaluated in higher resolution, with the short side at 512 pixels and the long side at multiples of $2^5$, *e.g.* 640.

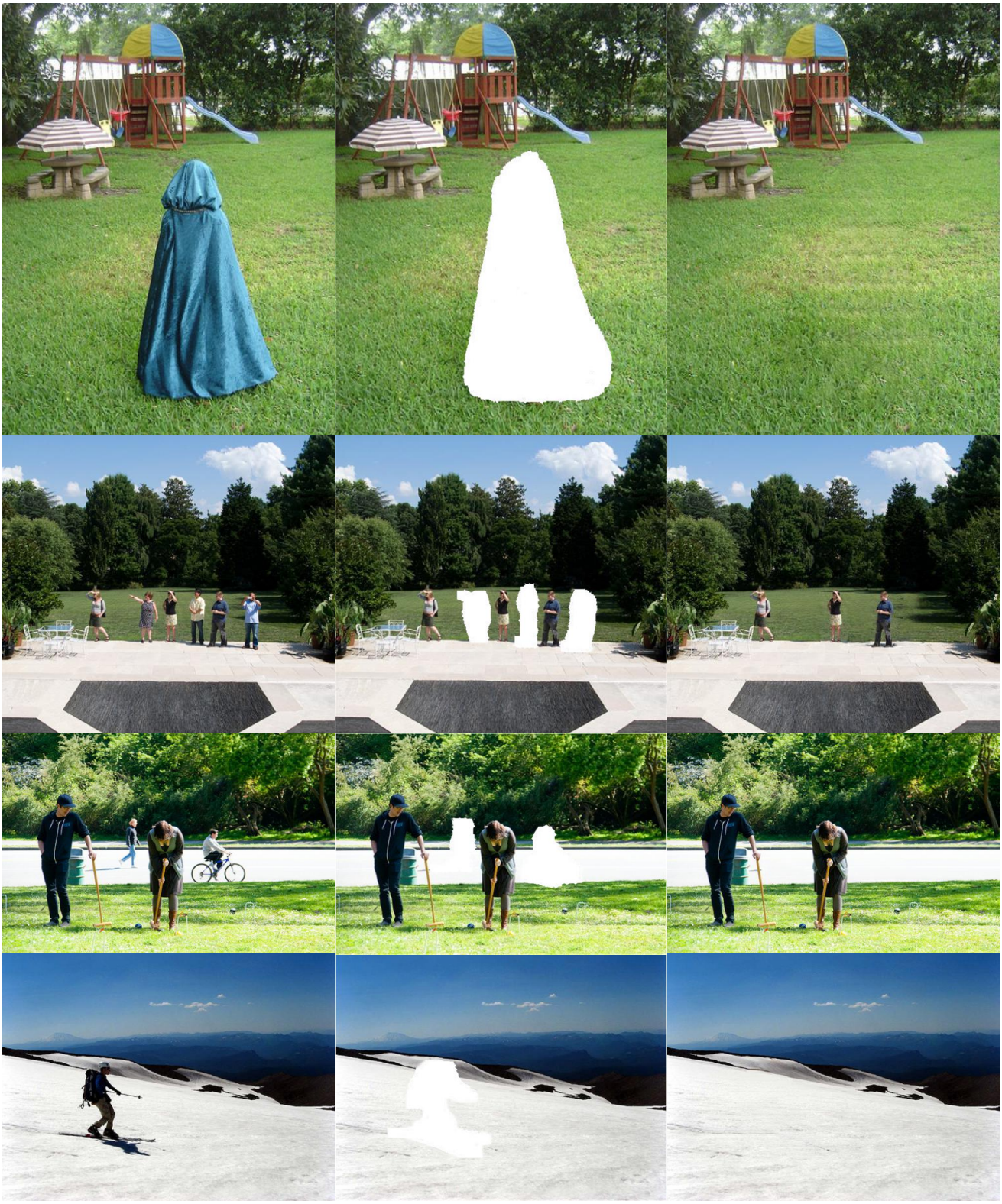

(a) Original (b) Edited Output (a) Original (b) Edited Output

Figure A.7. **Additional results on CelebA-HQ [22, 33] and FFHQ [23] testing set for free-form mask editing.** All results are reported at $512^2$ resolution. Our model works well for traditional object removal, such as removing eyeglasses and watermarks. Furthermore, we provide examples of more substantial modifications, *e.g*. facial hair removal, and expression modification in the last row.



(a) Original (b) Input (c) Edited Output

Figure A.8. **Additional free-form editing results on ImageNet [41].** The original image size in ImageNet is $500 \times 375$. Here, we resized slightly to $512 \times 384$ for image completion. We highlight the generated content, which has consistent textures to those in the visible regions.



(a) Original (b) Input (c) Edited Output

Figure A.9. **Additional results on ImageNet [41] testing set for free-form editing.** Here, we enforce the input image size to be multiples of 32, *e.g.* $960 \times 640$ and provide the high-resolution results on the corresponding image size. Zoom in to see the completed details.


(a) Original (b) Input (c) Edited Output

Figure A.10. **Additional results on Places2 [57] testing set for free-form editing.** Here, we enforce the input image size to be multiples of 32, *e.g*. $960 \times 640$ and provide the high-resolution results on the corresponding image size. Zoom in to see the completed details.

## B. Additional Quantitative Results

We further report quantitative results using traditional pixel-level and patch-level image quality evaluation metrics.

| | Method | CelebA-HQ | | | FFHQ | | |
|---|---|---|---|---|---|---|---|
| | | $\ell_1$loss ↓ | SSIM↑ | PSNR↑ | $\ell_1$loss ↓ | SSIM↑ | PSNR↑ |
| | CA [52] | 0.0310 | 0.8201 | 23.5667 | 0.0337 | 0.8099 | 22.7745 |
| | PIC [60] | 0.0209 | 0.8668 | 24.6860 | 0.0241 | 0.8547 | 24.3430 |
| | MEDFE [29] | 0.0208 | 0.8691 | 24.4733 | - | - | - |
| $\mathbb{A}$ | Traditional *Conv* | 0.0199 | 0.8693 | 24.5800 | 0.0241 | 0.8559 | 24.2271 |
| $\mathbb{B}$ | + Attention in G | 0.0196 | 0.8717 | 24.6512 | 0.0236 | 0.8607 | 24.4384 |
| $\mathbb{C}$ | + Restrictive *Conv* | 0.0191 | 0.8738 | 24.8067 | 0.0220 | 0.8681 | 24.9280 |
| $\mathbb{D}$ | + Transformer | 0.0189 | 0.8749 | 24.9467 | 0.0197 | 0.8751 | 25.1002 |
| $\mathbb{E}$ | + Masked Attention | 0.0183 | 0.8802 | 25.2510 | 0.0188 | 0.8765 | 25.1204 |
| $\mathbb{F}$ | + Refine Network | **0.0180** | **0.8821** | **25.4220** | **0.0184** | **0.8778** | **25.2061** |

Table B.1. Quantitative results for traditional pixel-level and patch-level metrics on center masked images.

Table B.1 provides a comparison of our results to state-of-the-art CNN-based models, as well as various alternative configurations for our design, on the center masked face testing set. This is an extension of Table 2 in the main paper. All images were normalized to the range [0,1] for quantitative evaluation. While there is no necessity to strongly encourage the completed images to be the same as the original ground-truth images, our TFill model nonetheless achieved better performance on these metrics too, including $\ell_1$ loss, structure similarity index (SSIM) and peak signal-to-noise ration (PSNR), suggesting that our TFill model is more capable of generating closer content to the original unmasked images.

## C. Experiment Details

Here we first present the novel layers and loss functions used to train our model, followed by the training details.

### C.1. Multihead *Weighted* Self-Attention

Our transformer encoder is built on the standard **qkv** `self-attention` (SA) [46] with a learned position embedding in each layer. Given an input sequence $\mathbf{z} \in \mathbb{R}^{N\times C}$, we first calculate the pairwise similarity $\mathbf{A}$ between each two elements as follows:

$$[\mathbf{q}, \mathbf{k}, \mathbf{v}] = \mathbf{W}_{qkv}\mathbf{z} \tag{C.1}$$

$$\mathbf{A} = \texttt{softmax}(\mathbf{q}\mathbf{k}^\top/\sqrt{C_h}) \tag{C.2}$$

where $\mathbf{W}_{qkv} \in \mathbb{R}^{C\times 3C_h}$ is the learned parameter to refine the features $\mathbf{z}$ for the query $\mathbf{q}$, the key $\mathbf{k}$ and the value $\mathbf{v}$. $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the dot similarity of N tokens, which is scaled by the square root of feature dimension $C_h$. Then, we compute a weighted sum over all values $\mathbf{v}$ via:

$$\text{SA}(\mathbf{z}) = \mathbf{A}\mathbf{v} \tag{C.3}$$

where the value $z$ in the sequence is connected through their learned similarity $A$, rather than purely depending on a fixed learned weight $w$.

The multihead `self-attention` (MSA) is an extention of SA, in which $H$ heads are run in parallel to get multiple attention scores and the corresponding projected results. Then we get the following function:

$$\text{MSA}(\mathbf{z}) = [\text{SA}_1(\mathbf{z}); \text{SA}_2(\mathbf{z}); \ldots; \text{SA}_h(\mathbf{z})] \tag{C.4}$$

To encourage the model to *bias* to the important visible values, we further modify the MSA with a *masked* self-attention layer, in which a masked weight is applied to scale the attention score $\mathbf{A}$. Given a feature $\mathbf{x}$ and the corresponding mask $\mathbf{m}$ (1

denotes visible pixel and 0 is masked pixel). The original partial convolution operation is operated as:

$$x' = \begin{cases} \mathbf{W}_p(\mathbf{x}_p \bigodot \mathbf{m}_p)\frac{1}{\sum(\mathbf{m}_p)} + b, & \text{if } \sum(\mathbf{m}_p) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{C.5}$$

$$m' = \begin{cases} 1, & \text{if } \sum(\mathbf{m}_p) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{C.6}$$

where $\mathbf{W}_p$ contain the convolution filter weights, $b$ is the corresponding bias, while $\mathbf{x}_p$ and $\mathbf{m}_p$ are the feature values and mask values in the current convolution window (*e.g.* $2 \times 2$ in our *restrictive CNN*), respectively. Here, we replace the $m'$ as a float value:

$$m' = \frac{\sum(\mathbf{m}_p)}{S} \tag{C.7}$$

where $S$ is the size of each convolution filter, $2 \times 2$ used in our *restrictive CNN*. To do this, each token only extracts the visible information. What's more, the final $m$ for each token denotes the percentage of valid values in each token under a small RF. Then, for each sequence $\mathbf{z} \in \mathbb{R}^{N \times C}$, we obtain a corresponding masked weight $\mathbf{m} \in \mathbb{R}^{N \times 1}$ by flattening the updated mask. Finally, we update the original attention score by multiplying with the repeated masked weight $\mathbf{m} \in \mathbb{R}^{N \times 1}$:

$$\mathbf{A}_m = \mathbf{A} \bigodot \mathbf{m}_r \tag{C.8}$$

where $\mathbf{m}_r \in \mathbb{R}^{N \times N}$ is the extension of masked weight $\mathbf{m} \in \mathbb{R}^{N \times 1}$ in the final dimension.

## C.2. Loss Functions

Our work focuses on exploiting the *token representation* in the visual transformer architecture. We do *not* modify the discriminator architecture or design the loss function in any way. Both TFill-*Coarse* and TFill-*Refined* is trained with loss $L = L_{pixel} + L_{per} + L_{GAN}$. In particular, each loss is given as:

$$L_{pixel} = ||\mathbf{I}_{gt} - \mathbf{I}_g||_1 \tag{C.9}$$

$$L_{per} = ||\Phi_n(\mathbf{I}_{gt}) - \Phi_n(\mathbf{I}_g)||_1 \tag{C.10}$$

$$L_{GAN} = log(1 + \exp(-D(\mathbf{I}_g))) \tag{C.11}$$

where $\mathbf{I}_g$ and $\mathbf{I}_{gt}$ is the generated image and original ground truth image, respectively. $\Phi_n$ is the activation map of the $n$th selected layer in VGG. $D$ is the discriminator and here we show only the generator loss for the generative adversarial traning.

## C.3. Training Details

Our model was trained on two NVIDIA A100 GPUs in two stages: **1)** the content inference network was first trained with $256^2$ resolution with batch size of 96; **2)** the visual appearance network was then trained with $512^2$ resolution with batch size of 24. Both networks were optimized using the loss $L = L_{pixel} + L_{per} + L_{GAN}$. The design of the encoder-decoder backbone follows the architecture presented in [12]. For the discriminator, we used the architecture of StyleGANv2 [24].