STGAN: A Unified Selective Transfer Network
for Arbitrary Image Attribute Editing
(Supplementary Material)

A. Content

The content of this supplementary material involves:
• Network structure of STGAN in Sec. B.
• Additional facial attribute editing results in Sec. C.
• Additional season translation results in Sec. D.

B. Network Structure

Our STGAN is comprised of two components, i.e., a generator $G$ and a discriminator $D$.

The generator $G$ has an encoder $G_{enc}$ for abstract latent representation, a decoder $G_{dec}$ for target image generation, and $G_{st}$ consists of a series of selective transfer units (STUs) for selective feature transfer.

The discriminator $D$ has two branches $D_{adv}$ and $D_{att}$. $D_{adv}$ distinguishes whether an image is a fake image or a real one, and $D_{att}$ predicts an attribute vector.

Fig. A shows the overall architecture of our STGAN, and Table A illustrates the details of the network layers (excluding STU, whose detailed structure and formulation have been discussed in the main text).

The generator $G$ has an encoder $G_{enc}$ for abstract latent representation, a decoder $G_{dec}$ for target image generation, and $G_{st}$ consists of a series of selective transfer units (STUs) for selective feature transfer.

The discriminator $D$ has two branches $D_{adv}$ and $D_{att}$. $D_{adv}$ distinguishes whether an image is a fake image or a real one, and $D_{att}$ predicts an attribute vector.

Fig. A shows the overall architecture of our STGAN, and Table A illustrates the details of the network layers (excluding STU, whose detailed structure and formulation have been discussed in the main text).

![Overall architecture of STGAN](image)

Figure A: Overall architecture of our STGAN. Taking the image above as an example, in the difference attribute vector $\text{att}_\text{diff}$, Young is set to 1, Mouth Open is set to $-1$, and others are set to zeros. The outputs of $D_{adv}$ and $D_{att}$ are the scalar $D_{adv}(G(x, \text{att}_\text{diff}))$ and the vector $D_{att}(G(x, \text{att}_\text{diff}))$, respectively.

<table>
<thead>
<tr>
<th>$l$</th>
<th>$G_{enc}$</th>
<th>$G_{dec}$</th>
<th>$D_{adv}$</th>
<th>$D_{att}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Conv(d,4,2), BN, Leaky ReLU</td>
<td>DeConv(3,4,2), Tanh</td>
<td>Conv(d,4,2), IN, Leaky ReLU</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Conv(d*2,4,2), BN, Leaky ReLU</td>
<td>DeConv(d*2,4,2), BN, ReLU</td>
<td>Conv(d*2,4,2), IN, Leaky ReLU</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Conv(d*4,4,2), BN, Leaky ReLU</td>
<td>DeConv(d*4,4,2), BN, ReLU</td>
<td>Conv(d*4,4,2), IN, Leaky ReLU</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Conv(d*8,4,2), BN, Leaky ReLU</td>
<td>DeConv(d*8,4,2), BN, ReLU</td>
<td>Conv(d*8,4,2), IN, Leaky ReLU</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Conv(d*16,4,2), BN, Leaky ReLU</td>
<td>DeConv(d*16,4,2), BN, ReLU</td>
<td>Conv(d*16,4,2), IN, Leaky ReLU</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>FC(1024), Leaky ReLU</td>
<td>FC(1024), Leaky ReLU</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>FC(1)</td>
<td>FC(c), Sigmoid</td>
<td></td>
</tr>
</tbody>
</table>

Table A: Architecture of STGAN components excluding $G_{st}$. Conv(dim, k, s) and DeConv(dim, k, s) denote the convolutional layer and transposed convolutional layer, whose output channel is dim, kernel size is k, stride is s. BN and IN represent batch normalization [3] and instance normalization [6], respectively. d is the base dimension of the network, which is set to 64 for 128 × 128 images and 32 for larger ones. c means the amount of attributes.
C. Facial Attribute Editing Results

In Fig. B, we show more images generated by all competing methods (i.e., IcGAN [5], FaderNet [4], AttGAN [2], StarGAN [1] and our STGAN) with single attribute manipulated. The Hair Color is set to one of Black Hair, Blond Hair and Brown Hair distinct from the original one, and the others are modified by inversion.

![Figure B: Facial attribute editing results generated by IcGAN [5], FaderNet [4], AttGAN [2], StarGAN [1] and our STGAN. Please zoom in for better observation.](image-url)
Furthermore, we show images generated by three top-performance methods (i.e., AttGAN [2], StarGAN [1] and STGAN) with complex and/or multiple attributes changed in Fig. C and high resolution results of STGAN in Fig. D.

Figure C: Facial attribute editing results generated by AttGAN [2], StarGAN [1] and our STGAN, with complex and/or multiple attributes changed. Please zoom in for better observation.
Figure D: High resolution (384 × 384) results of STGAN on facial attribute editing task.
To evaluate difference attribute vector, in addition to attribute generation accuracy reported in the main text, we also give qualitative results of AttGAN/AttGAN-diff, StarGAN/StarGAN-diff and STGAN-dst/STGAN in Fig. E. All three methods can better manipulate the attributes with difference attribute vector, and less other attributes are wrongly edited. Besides, the difference attribute vector also benefits the reconstruction quality as shown in Table B. And the qualitative results of STU variants are given in Fig. F.

Furthermore, the training and inference time of AttGAN and our STGAN are given in Table C.

<table>
<thead>
<tr>
<th>Method</th>
<th>AttGAN</th>
<th>StarGAN</th>
<th>STGAN-dst</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR/SSIM</td>
<td>24.07/0.841</td>
<td>22.80/0.819</td>
<td>30.22/0.942</td>
</tr>
</tbody>
</table>

Table B: Reconstruction evaluation on difference attribute vector.

<table>
<thead>
<tr>
<th>Method</th>
<th>Training</th>
<th>Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttGAN</td>
<td>20 min/epoch</td>
<td>7.21 ms/image</td>
</tr>
<tr>
<td>STGAN</td>
<td>27 min/epoch</td>
<td>11.78 ms/image</td>
</tr>
</tbody>
</table>

Table C: Comparison of training and inference time. Note that we count inference time of image generation without preprocessing.
To further explore the selective mechanism and verify \( z^l_l (l = 1, \ldots, 4) \) in STUs can learn to correctly select the edited attributes-irrelevant features, we show the visualization of representative \( z^l_l \)s in Fig. H. We can know that \( z^l_l \) is able to learn to selectively transfer features, especially when the edited attributes are local ones (e.g., Bald, Mustache).

We also show two representative failure cases in Fig. G. In the first row, even the sunglasses are removed, there is nothing about the occluded eyes to transfer from input image, and STGAN fails to generate high-quality eyes. In the second row, STGAN fails to recognize the hat and falsely intends to edit it to blond hair. Actually, these failure cases may be tackled by deploying hierarchical noise vectors and introducing extra attributes (i.e., hat), which will be further investigated in our future work.

![Figure G: Failure cases on removing eyeglasses and changing to blond hair.](image1)

![Figure H: Visualization of representative \( z^l_l \)s, each column represents a specific task.](image2)

### D. Season Translation Results

Fig. I shows more season translation results.

![Figure I: Season translation results by AttGAN [2], StarGAN [1], CycleGAN [7] and STGAN. On the left are results on summer→winter task, and on the right are results on winter→summer task. Please zoom in for better observation.](image3)
References


