Supplementary File: Spiral Generative Network for Image Extrapolation

1 Mathematical Derivation of Hue-Color Loss

We formulate our hue-color loss in ImagineGAN (and SliceGAN as well) as:

$$\mathcal{L}^{\mathbf{I}}_{hue} = \frac{1}{h \times w} \sum_{i,j} \left\{1 - \min\left[\cos(\mathbf{I}_{ij}, \bar{\mathbf{Y}}_{ij}), \cos(\mathbb{1} - \mathbf{I}_{ij}, \mathbb{1} - \bar{\mathbf{Y}}_{ij})\right] + \xi\right\}^{\gamma}, \quad (1)$$

where $\mathbf{I}_{ij} = (R_{\mathbf{I}}, G_{\mathbf{I}}, B_{\mathbf{I}})$ and $\bar{\mathbf{Y}}_{ij} = (R_{\bar{\mathbf{Y}}}, G_{\bar{\mathbf{Y}}}, B_{\bar{\mathbf{Y}}})$ denote RGB vectors in pixel (i, j) of imaginary output and ground truth respectively, whose values are scaled down to $[0, 1]$. We propose this hue-color loss to care about real "color" regardless of gray, and we will present the mathematical derivation below.

Propositon 1 *For* $\cos(\mathbf{I}_{ij}, \bar{\mathbf{Y}}_{ij}) = 1$*, the hues of* $\mathbf{I}_{ij}$ *and* $\bar{\mathbf{Y}}_{ij}$ *are the same:*

$$H_{\mathbf{I}} = H_{\bar{\mathbf{Y}}}. \quad (2)$$

Proof For $\cos(\mathbf{I}_{ij}, \bar{\mathbf{Y}}_{ij}) = 1$, it's obvious that:

$$(R_{\mathbf{I}}, G_{\mathbf{I}}, B_{\mathbf{I}}) = (mR_{\bar{\mathbf{Y}}}, mG_{\bar{\mathbf{Y}}}, mB_{\bar{\mathbf{Y}}}), \quad m \in \mathbb{R}^{+}. \quad (3)$$

According to the conversion formulae from RGB space to HSL space [1], we get hue:

$$H = \begin{cases} \text{undefined} & \text{if } \max = \min, \\ 60^\circ \times \frac{G-B}{\max-\min} & \text{if } \max = R \text{ and } G \geq B, \\ 60^\circ \times \frac{G-B}{\max-\min} + 360^\circ & \text{if } \max = R \text{ and } G < B, \\ 60^\circ \times \frac{B-R}{\max-\min} + 120^\circ & \text{if } \max = G, \\ 60^\circ \times \frac{R-G}{\max-\min} + 240^\circ & \text{if } \max = B, \end{cases} \quad (4)$$

where $\max = \max(R, G, B)$ and $\min = \min(R, G, B)$.

Considering Eq. (3) and Eq. (4), we can get:

(a) If $\max_{\mathbf{I}} = \min_{\mathbf{I}}$, then $R_{\mathbf{I}} = G_{\mathbf{I}} = B_{\mathbf{I}}$.
According to Eq. (3), we obtain $R_{\bar{\mathbf{Y}}} = G_{\bar{\mathbf{Y}}} = B_{\bar{\mathbf{Y}}}$.
Therefore, $\max_{\bar{\mathbf{Y}}} = \min_{\bar{\mathbf{Y}}}$ and $H_{\bar{\mathbf{Y}}}$ is also undefined.

(b) If $\max_{\mathbf{I}} \neq \min_{\mathbf{I}}$, taking one case $\max = R$ and $G \geq B$ in Eq. (4) for example, we have:

$$H_{\mathbf{I}} = 60^\circ \times \frac{G_{\mathbf{I}} - B_{\mathbf{I}}}{\max_{\mathbf{I}} - \min_{\mathbf{I}}} = 60^\circ \times \frac{mG_{\bar{\mathbf{Y}}} - mB_{\bar{\mathbf{Y}}}}{m\max_{\bar{\mathbf{Y}}} - m\min_{\bar{\mathbf{Y}}}} = H_{\bar{\mathbf{Y}}}. \quad (5)$$

Similarly, the other cases in Eq. (4) can be proved in a same way as Eq. (5), concluding the proof.

$\square$

Notably, other commonly used color spaces associated with hue, *i.e.*, HSV, HSB, and HSI, can also be proved similarly. As our paper declares, same hue may lead to quite different color appearance. To address this issue, we introduce inverse color cosine distance to our hue-color loss for maintaining consistency of both real "color" (regardless of gray) and the hue as well.

Propositon 2 *For* $\cos(\mathbf{I}_{ij}, \bar{\mathbf{Y}}_{ij}) = 1$ *and* $\cos(\mathbb{1} - \mathbf{I}_{ij}, \mathbb{1} - \bar{\mathbf{Y}}_{ij}) = 1$*, except for the cases of* $R_{\mathbf{I}} = G_{\mathbf{I}} = B_{\mathbf{I}}$ *and* $R_{\bar{\mathbf{Y}}} = G_{\bar{\mathbf{Y}}} = B_{\bar{\mathbf{Y}}}$ *that indicate both gray pixels,* $\mathbf{I}_{ij}$ *and* $\bar{\mathbf{Y}}_{ij}$ *are equal:*

$$\mathbf{I}_{ij} = \bar{\mathbf{Y}}_{ij}, \tag{6}$$

thus maintaining color consistency regardless of gray.

Proof For $\cos(\mathbf{I}_{ij}, \bar{\mathbf{Y}}_{ij}) = 1$ and $\cos(\mathbb{1} - \mathbf{I}_{ij}, \mathbb{1} - \bar{\mathbf{Y}}_{ij}) = 1$, we get:

$$\mathbf{I}_{ij} = m\bar{\mathbf{Y}}_{ij}, \quad m \in \mathbb{R}^+, \tag{7}$$

$$\mathbb{1} - \mathbf{I}_{ij} = n(\mathbb{1} - \bar{\mathbf{Y}}_{ij}), \quad n \in \mathbb{R}^+. \tag{8}$$

Plugging Eq. (7) into Eq. (8), we obtain:

$$\begin{aligned} &\mathbb{1} - m\bar{\mathbf{Y}}_{ij} = n(\mathbb{1} - \bar{\mathbf{Y}}_{ij}) \\ &\Rightarrow \mathbb{1} - \mathbb{n} = (m - n)\bar{\mathbf{Y}}_{ij}. \end{aligned} \tag{9}$$

Then, considering all values of $\mathbb{1} - \mathbb{n}$, we can get:

(a) If $\mathbb{1} - \mathbb{n} = \mathbb{0}$, then $\mathbb{n} = \mathbb{1}$.
For any $\bar{\mathbf{Y}}_{ij}$, $m = n = 1$.
Therefore, $\mathbf{I}_{ij} = \bar{\mathbf{Y}}_{ij}$.

(b) If $\mathbb{1} - \mathbb{n} \neq \mathbb{0}$, then vector $\mathbb{1} - \mathbb{n} = (1-n, 1-n, 1-n)$ has three equal elements.
According to Eqs. (9) and (7), we obtain $R_{\bar{\mathbf{Y}}} = G_{\bar{\mathbf{Y}}} = B_{\bar{\mathbf{Y}}}$ and $R_{\mathbf{I}} = G_{\mathbf{I}} = B_{\mathbf{I}}$.
Therefore, $\mathbf{I}_{ij}$ and $\bar{\mathbf{Y}}_{ij}$ should be both gray pixels.

Thus, except for the cases of both gray pixels, the colors of $\mathbf{I}_{ij}$ and $\bar{\mathbf{Y}}_{ij}$ are consistent, concluding the proof. $\square$

Now, let's go back to our proposed hue-color loss Eq. (1):
As values of $\mathbf{I}_{ij}$ and $\bar{\mathbf{Y}}_{ij}$ belong to $[0, 1]$, $\cos(\mathbf{I}_{ij}, \bar{\mathbf{Y}}_{ij}) \in [0, 1]$ and $\cos(\mathbb{1} - \mathbf{I}_{ij}, \mathbb{1} - \bar{\mathbf{Y}}_{ij}) \in [0, 1]$.
For constraining $\mathcal{L}^{\mathbf{I}}_{hue} \to 0$, $\min\left[\cos(\mathbf{I}_{ij}, \bar{\mathbf{Y}}_{ij}), \cos(\mathbb{1} - \mathbf{I}_{ij}, \mathbb{1} - \bar{\mathbf{Y}}_{ij})\right] \to 1$.
$\Rightarrow$ Then $\cos(\mathbf{I}_{ij}, \bar{\mathbf{Y}}_{ij}) \to 1$ and $\cos(\mathbb{1} - \mathbf{I}_{ij}, \mathbb{1} - \bar{\mathbf{Y}}_{ij}) \to 1$. $\Rightarrow$
According to Proposition 1, $\mathbf{I}_{ij}$ and $\bar{\mathbf{Y}}_{ij}$ try to keep the same hue.
According to Proposition 2, $\mathbf{I}_{ij}$ and $\bar{\mathbf{Y}}_{ij}$ try to maintain color consistency regardless of gray.
$\Rightarrow$ Thus our hue-color loss helps to constrain consistency of both hue and color.

2 Network Architecture Details

Our proposed SpiralNet consists of ImageinGAN and SliceGAN with the corresponding details illustrated below.

2.1 ImagineGAN

Our ImagineGAN is essentially a cGAN with an Encoder-Decoder generator and a discriminator.

Generator. ImagineGAN's generator architecture is shown in Table 1. Notably, the network has 8 residual blocks unlike 6 residual blocks in slice generator.

Discriminator. Besides the architecture shown in Table 2, ImagineGAN's discriminator applies spectral normalization [7] at all layers.

2.2 SliceGAN

Our SliceGAN has a novel spiral architecture with a new structure of slice generator and two well-designed discriminators: a spiral discriminator and an extrapolate discriminator.

Slice Generator. We design a new slice-wise generator which is an Encoder-AdaIN-SPADE|Decoder structure as shown in Table 3. Notably, we introduce AdaIN between layer 3 and layer 4, which fuses sub-image style information extracted by pretrained VGG-19 [9] layer3_4. Layers 16-18 and layers 20-22 represent SPADE modules, taking in charge of the closest slice for combining semantic information.

Spiral/Extrapolate Discriminator. Spiral discriminator has a similar structure to DenseNet [4] shown in Table 4. And we adopt ImagineGAN's discriminator to implement our extrapolate discriminator as shown in Table 2.

3 Additional Qualitative Results

We show additional qualitative results on CelebA-HQ [5], Stanford Cars [6], CUB [10], Flowers [8], Cityscapes [2], Place365 Sky [11], Paris StreetView [3], and Place365 Desert Road [11] in Figs. 1-8 respectively. Besides, Fig. 9 shows additional SpiralNet-UM results.

4 Additional Results of Why Spiral Is Necessary

Fig. 10 shows additional visual comparison results for necessity analysis of our spiral architecture.

5 Additional Results of SliceGAN Analysis

Fig. 11 shows additional qualitative ablation study on ternary SliceGAN inputs. And Fig. 12 shows additional qualitative ablation study on different slice sizes.

6 Additional Results of Hue-Color Loss

We show additional results about efficacy of our hue-color loss in Fig. 13.

References

1. Agoston, M.K.: Computer Graphics and Geometric Modeling: Implementation and Algorithms. Springer (2005)
2. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR. pp. 3213–3223 (2016)
3. Doersch, C., Singh, S., Gupta, A., Sivic, J., Efros, A.A.: What makes paris look like paris? ACM TOG **31**(4), 101 (2012)
4. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR. pp. 4700–4708 (2017)
5. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: ICLR (2018)
6. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3D object representations for fine-grained categorization. In: ICCVW. pp. 554–561 (2013)
7. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: ICLR (2018)
8. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: ICVGIP. pp. 722–729 (2008)
9. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
10. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011)
11. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE TPAMI pp. 1452–1464 (2017)

Table 1. The architecture of ImagineGAN generator. "Act." denotes activation type, "K" denotes kernel size, "S" denotes stride, "P" denotes padding, "D" denotes dilation, "Out" denotes output channels' number, and "Skip" denotes the concatenated layer-id.

Layer ID	Type	Act.	K	S	P	D	Out	Skip
1	Conv	ReLU	7	1	3	1	64	None
2	Conv	ReLU	3	2	1	1	128	None
3	Conv	ReLU	3	2	1	1	256	None
4	Conv	ReLU	3	1	2	2	256	None
5	Conv	None	3	1	1	1	256	4
6	Conv	ReLU	3	1	2	2	256	None
7	Conv	None	3	1	1	1	256	6
8	Conv	ReLU	3	1	2	2	256	None
9	Conv	None	3	1	1	1	256	8
10	Conv	ReLU	3	1	2	2	256	None
11	Conv	None	3	1	1	1	256	10
12	Conv	ReLU	3	1	2	2	256	None
13	Conv	None	3	1	1	1	256	12
14	Conv	ReLU	3	1	2	2	256	None
15	Conv	None	3	1	1	1	256	14
16	Conv	ReLU	3	1	2	2	256	None
17	Conv	None	3	1	1	1	256	16
18	Conv	ReLU	3	1	2	2	256	None
19	Conv	None	3	1	1	1	256	18
20	Deconv	ReLU	3	2	1	1	128	None
21	Deconv	ReLU	3	2	1	1	64	None
22	Deconv	Tanh	7	1	3	1	3	None

Table 2. The architecture of ImagineGAN discriminator and SliceGAN extrapolate discriminator as well. "Act." denotes activation type, "K" denotes kernel size, "S" denotes stride, "P" denotes padding, "D" denotes dilation, and "Out" denotes output channels' number.

Layer ID	Type	Act.	K	S	P	D	Out
1	Conv	LeakyReLU	4	2	1	1	64
2	Conv	LeakyReLU	4	2	1	1	128
3	Conv	LeakyReLU	4	2	1	1	256
4	Conv	LeakyReLU	4	1	1	1	512
5	Conv	None	4	1	1	1	1

Table 3. The architecture of SliceGAN slice generator. "Act." denotes activation type, "K" denotes kernel size, "S" denotes stride, "P" denotes padding, "D" denotes dilation, "Out" denotes output channels' number, and "Skip" denotes the concatenated layer-id.

<table>
<tr><th>Layer ID</th><th colspan="2">Type</th><th>Act.</th><th>K</th><th>S</th><th>P</th><th>D</th><th>Out</th><th>Skip</th></tr>
<tr><td>1</td><td colspan="2">Conv</td><td>ReLU</td><td>7</td><td>1</td><td>3</td><td>1</td><td>64</td><td>None</td></tr>
<tr><td>2</td><td colspan="2">Conv</td><td>ReLU</td><td>4</td><td>2</td><td>1</td><td>1</td><td>128</td><td>None</td></tr>
<tr><td>3</td><td colspan="2">Conv</td><td>ReLU</td><td>4</td><td>2</td><td>1</td><td>1</td><td>256</td><td>None</td></tr>
<tr><td>4</td><td colspan="2">Conv</td><td>ReLU</td><td>3</td><td>1</td><td>2</td><td>2</td><td>256</td><td>None</td></tr>
<tr><td>5</td><td colspan="2">Conv</td><td>None</td><td>3</td><td>1</td><td>1</td><td>1</td><td>256</td><td>4</td></tr>
<tr><td>6</td><td colspan="2">Conv</td><td>ReLU</td><td>3</td><td>1</td><td>2</td><td>2</td><td>256</td><td>None</td></tr>
<tr><td>7</td><td colspan="2">Conv</td><td>None</td><td>3</td><td>1</td><td>1</td><td>1</td><td>256</td><td>6</td></tr>
<tr><td>8</td><td colspan="2">Conv</td><td>ReLU</td><td>3</td><td>1</td><td>2</td><td>2</td><td>256</td><td>None</td></tr>
<tr><td>9</td><td colspan="2">Conv</td><td>None</td><td>3</td><td>1</td><td>1</td><td>1</td><td>256</td><td>8</td></tr>
<tr><td>10</td><td colspan="2">Conv</td><td>ReLU</td><td>3</td><td>1</td><td>2</td><td>2</td><td>256</td><td>None</td></tr>
<tr><td>11</td><td colspan="2">Conv</td><td>None</td><td>3</td><td>1</td><td>1</td><td>1</td><td>256</td><td>10</td></tr>
<tr><td>12</td><td colspan="2">Conv</td><td>ReLU</td><td>3</td><td>1</td><td>2</td><td>2</td><td>256</td><td>None</td></tr>
<tr><td>13</td><td colspan="2">Conv</td><td>None</td><td>3</td><td>1</td><td>1</td><td>1</td><td>256</td><td>12</td></tr>
<tr><td>14</td><td colspan="2">Conv</td><td>ReLU</td><td>3</td><td>1</td><td>2</td><td>2</td><td>256</td><td>None</td></tr>
<tr><td>15</td><td colspan="2">Conv</td><td>None</td><td>3</td><td>1</td><td>1</td><td>1</td><td>256</td><td>14</td></tr>
<tr><td>16</td><td colspan="2">Conv</td><td>ReLU</td><td>3</td><td>1</td><td>1</td><td>1</td><td>256</td><td>None</td></tr>
<tr><td>17</td><td>Conv</td><td>Conv</td><td>None</td><td>3</td><td>1</td><td>1</td><td>1</td><td>256</td><td>None</td></tr>
<tr><td>18</td><td colspan="2">Conv</td><td>None</td><td>3</td><td>1</td><td>1</td><td>1</td><td>256</td><td>16</td></tr>
<tr><td>19</td><td colspan="2">Deconv</td><td>ReLU</td><td>3</td><td>2</td><td>1</td><td>1</td><td>128</td><td>None</td></tr>
<tr><td>20</td><td colspan="2">Conv</td><td>ReLU</td><td>3</td><td>1</td><td>1</td><td>1</td><td>128</td><td>None</td></tr>
<tr><td>21</td><td>Conv</td><td>Conv</td><td>None</td><td>3</td><td>1</td><td>1</td><td>1</td><td>128</td><td>None</td></tr>
<tr><td>22</td><td colspan="2">Conv</td><td>None</td><td>3</td><td>1</td><td>1</td><td>1</td><td>128</td><td>20</td></tr>
<tr><td>23</td><td colspan="2">Deconv</td><td>ReLU</td><td>3</td><td>2</td><td>1</td><td>1</td><td>64</td><td>None</td></tr>
<tr><td>24</td><td colspan="2">Conv</td><td>Tanh</td><td>7</td><td>1</td><td>3</td><td>1</td><td>3</td><td>None</td></tr>
</table>

Table 4. The architecture of SliceGAN spiral discriminator. "Max Pool" and "Average Pool" denote max pooling and average pooling layers respectively. "Fully Connected" denotes fully-connected layer. "Act." denotes activation type, "K" denotes kernel size, "S" denotes stride, "P" denotes padding, "D" denotes dilation, "Out" denotes output channels' number, and "Skip" denotes the concatenated layer-id.

Layer ID	Type	Act.	K	S	P	D	Out	Skip
1	Conv	ReLU	7	2	3	1	64	None
2	Max Pool	None	3	2	1	1	32	None
3	Conv	ReLU	1	1	3	1	128	None
4	Conv	ReLU	3	1	1	1	32	None
5	Conv	ReLU	1	1	3	1	128	3
6	Conv	ReLU	3	1	1	1	32	3,4
7	Conv	ReLU	1	1	3	1	128	3,4,5
8	Conv	ReLU	3	1	1	1	32	3,4,5,6
9	Conv	ReLU	1	1	0	1	32	None
10	Average Pool	None	2	2	0	1	16	None
11	Conv	ReLU	1	1	3	1	128	None
12	Conv	ReLU	3	1	1	1	16	None
13	Conv	ReLU	1	1	3	1	128	11
14	Conv	ReLU	3	1	1	1	16	11,12
15	Conv	ReLU	1	1	3	1	128	11,12,13
16	Conv	ReLU	3	1	1	1	16	11,12,13,14
17	Conv	ReLU	1	1	0	1	16	None
18	Average Pool	None	2	2	0	1	8	None
19	Conv	ReLU	1	1	3	1	128	None
20	Conv	ReLU	3	1	1	1	8	None
21	Conv	ReLU	1	1	3	1	128	19
22	Conv	ReLU	3	1	1	1	8	19,20
23	Conv	ReLU	1	1	3	1	128	19,20,21
24	Conv	ReLU	3	1	1	1	8	19,20,21,22
25	Average Pool	None	7	1	0	1	4	None
26	Fully Connected	Sigmoid	-	-	-	-	1000	None

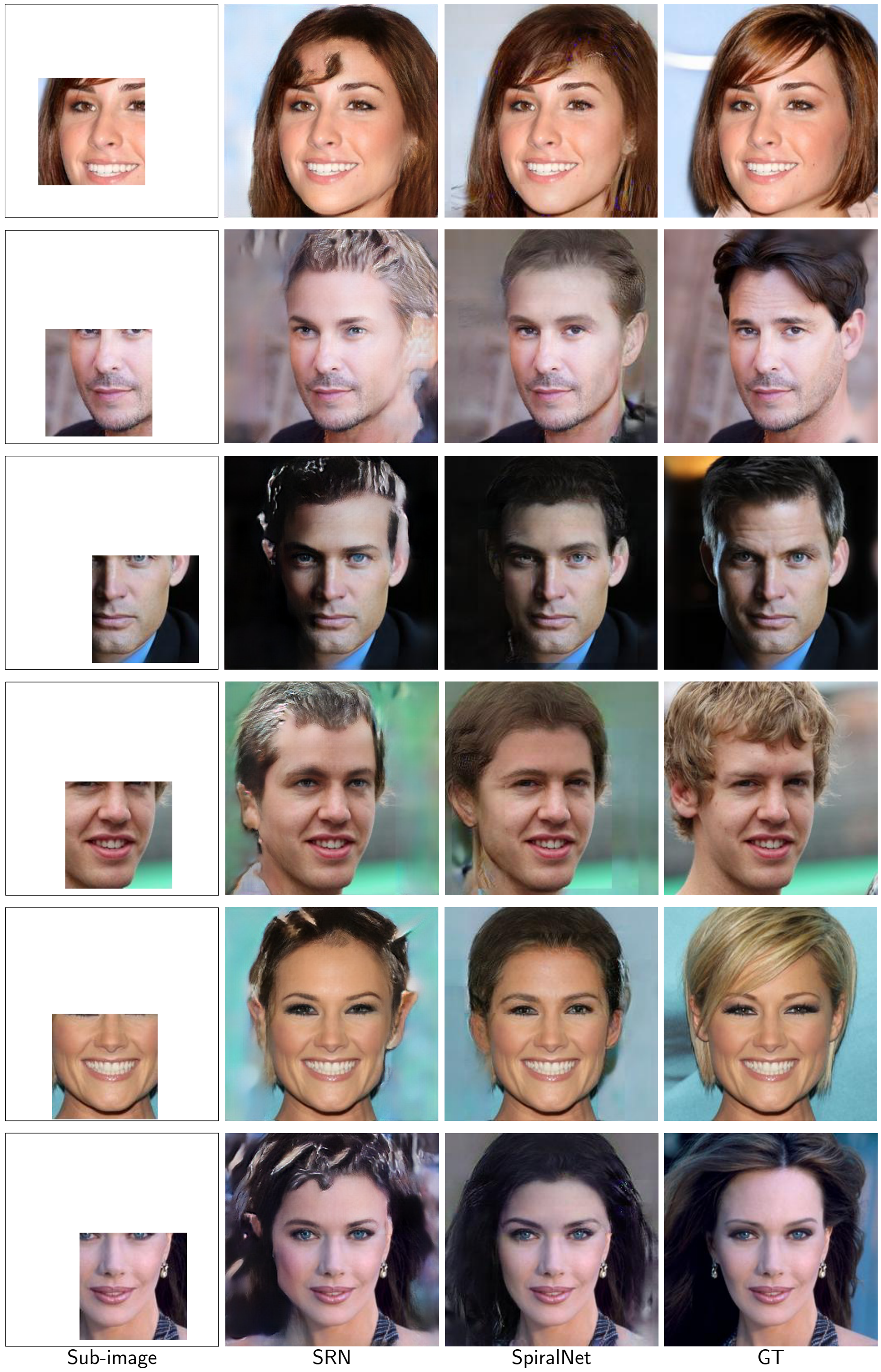


Fig. 1. Additional qualitative results on CelebA-HQ.

Fig. 2. Additional qualitative results on Stanford Cars.

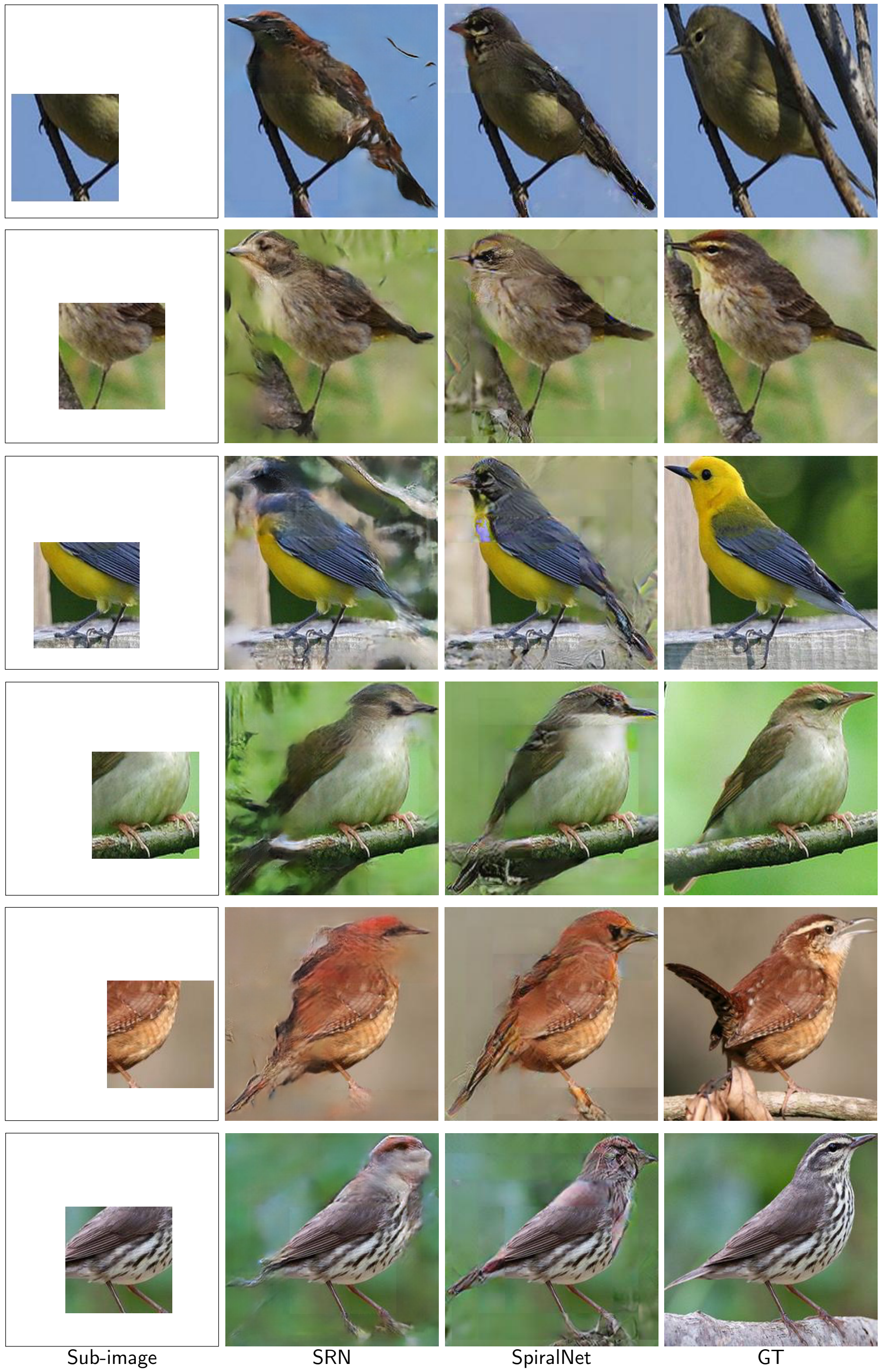


Fig. 3. Additional qualitative results on CUB.

Fig. 4. Additional qualitative results on Flowers.

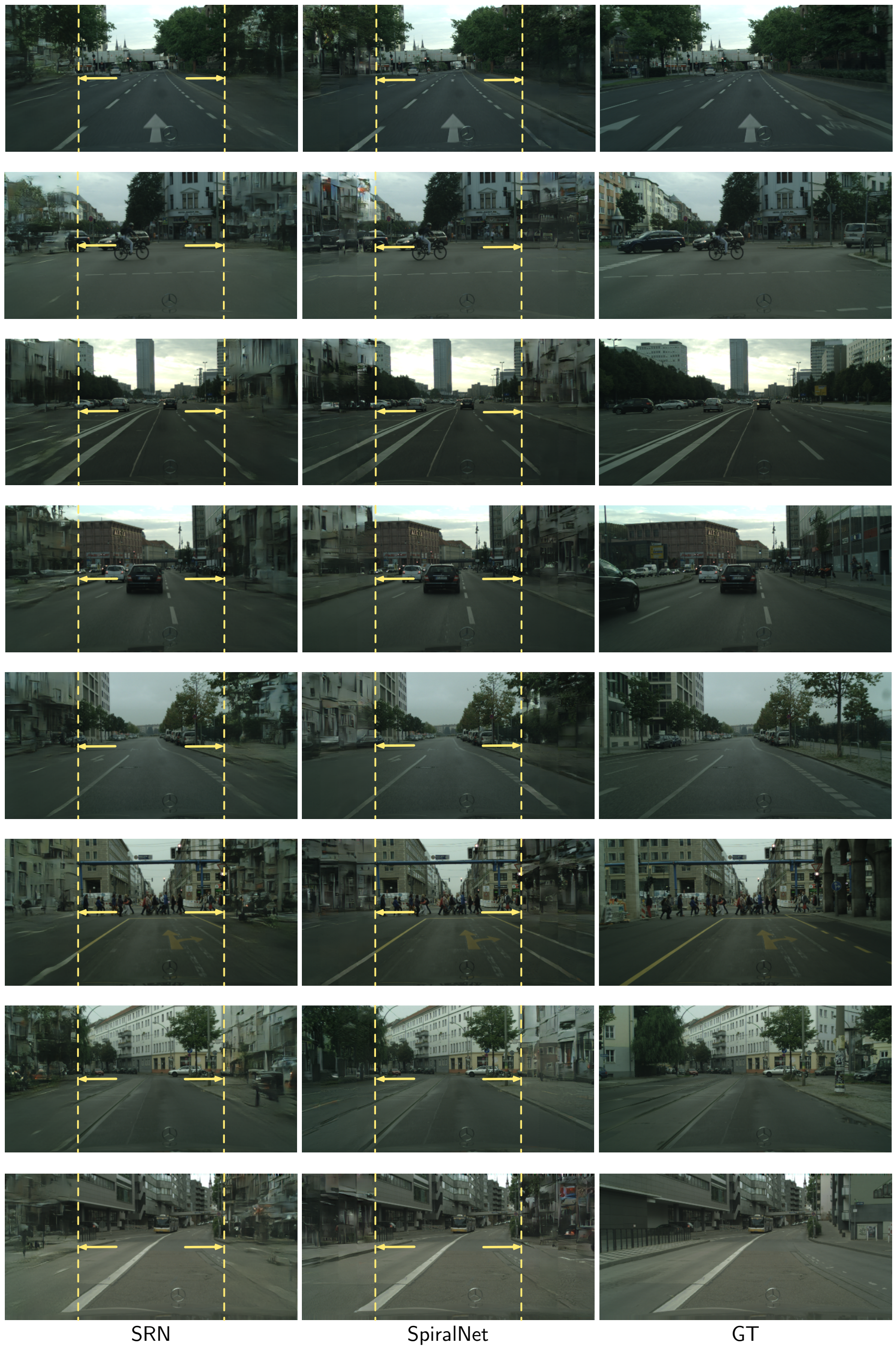


Fig. 5. Additional qualitative results on Cityscapes.

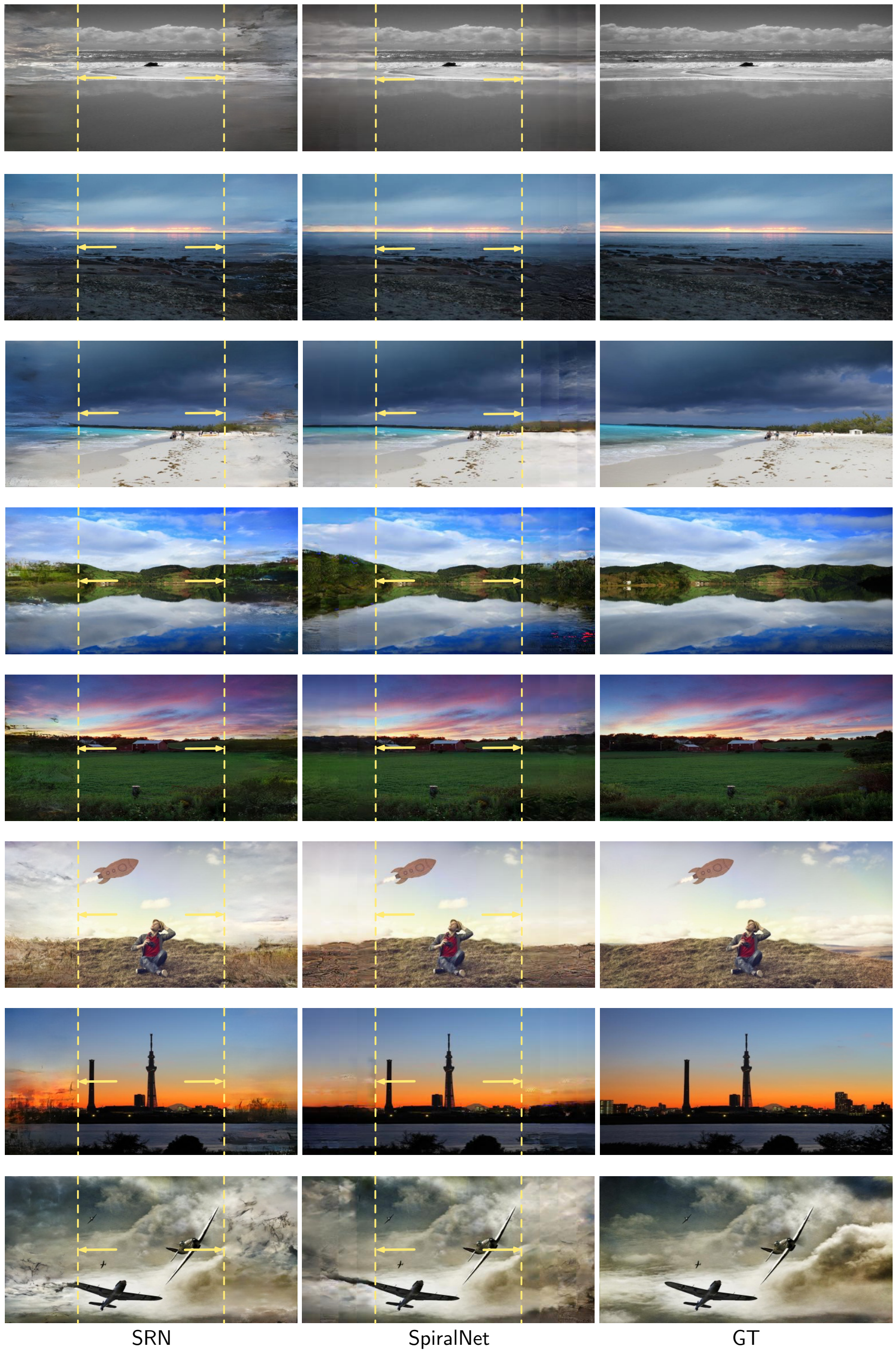


Fig. 6. Additional qualitative results on Place365 Sky.

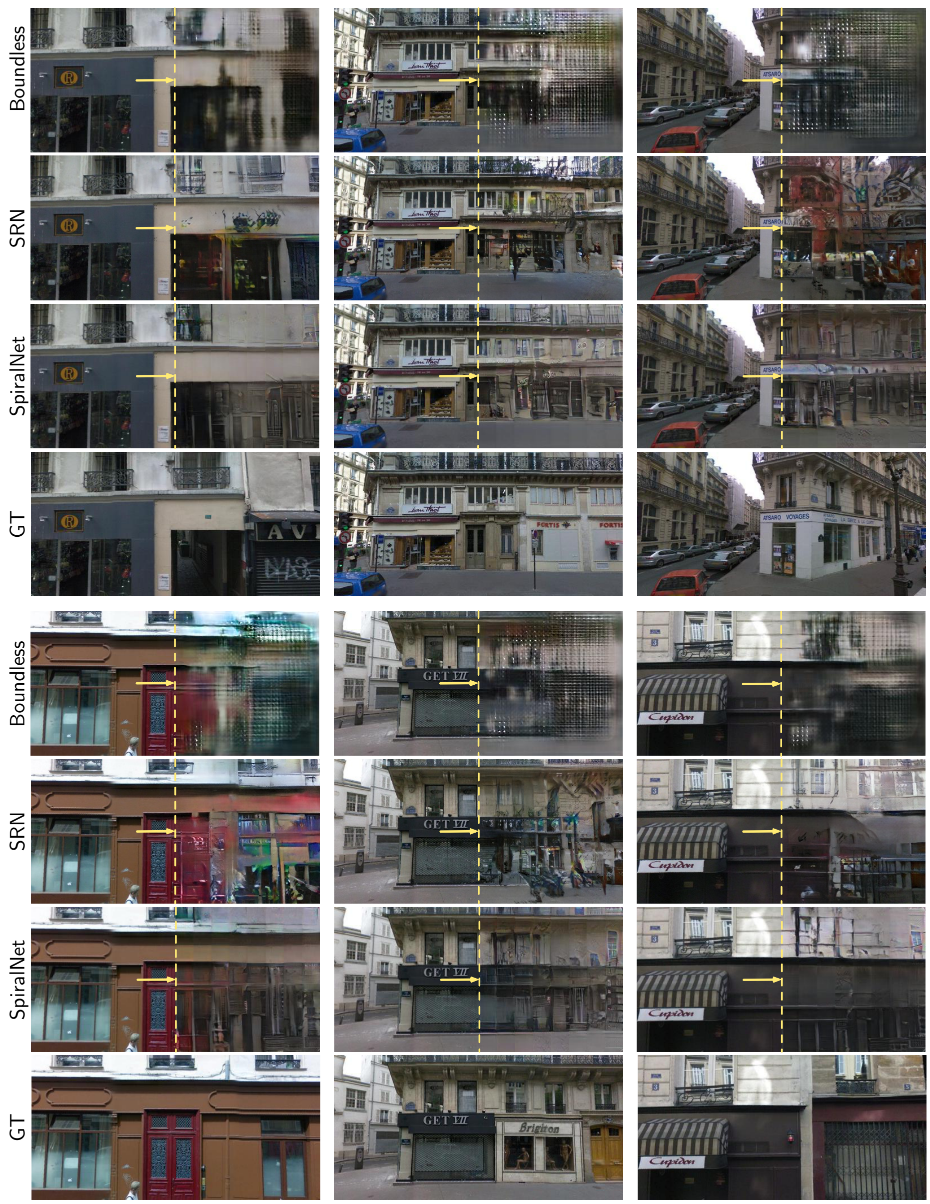


Fig. 7. Additional qualitative results on Paris StreetView.

Fig. 8. Additional qualitative results on Place365 Desert Road.

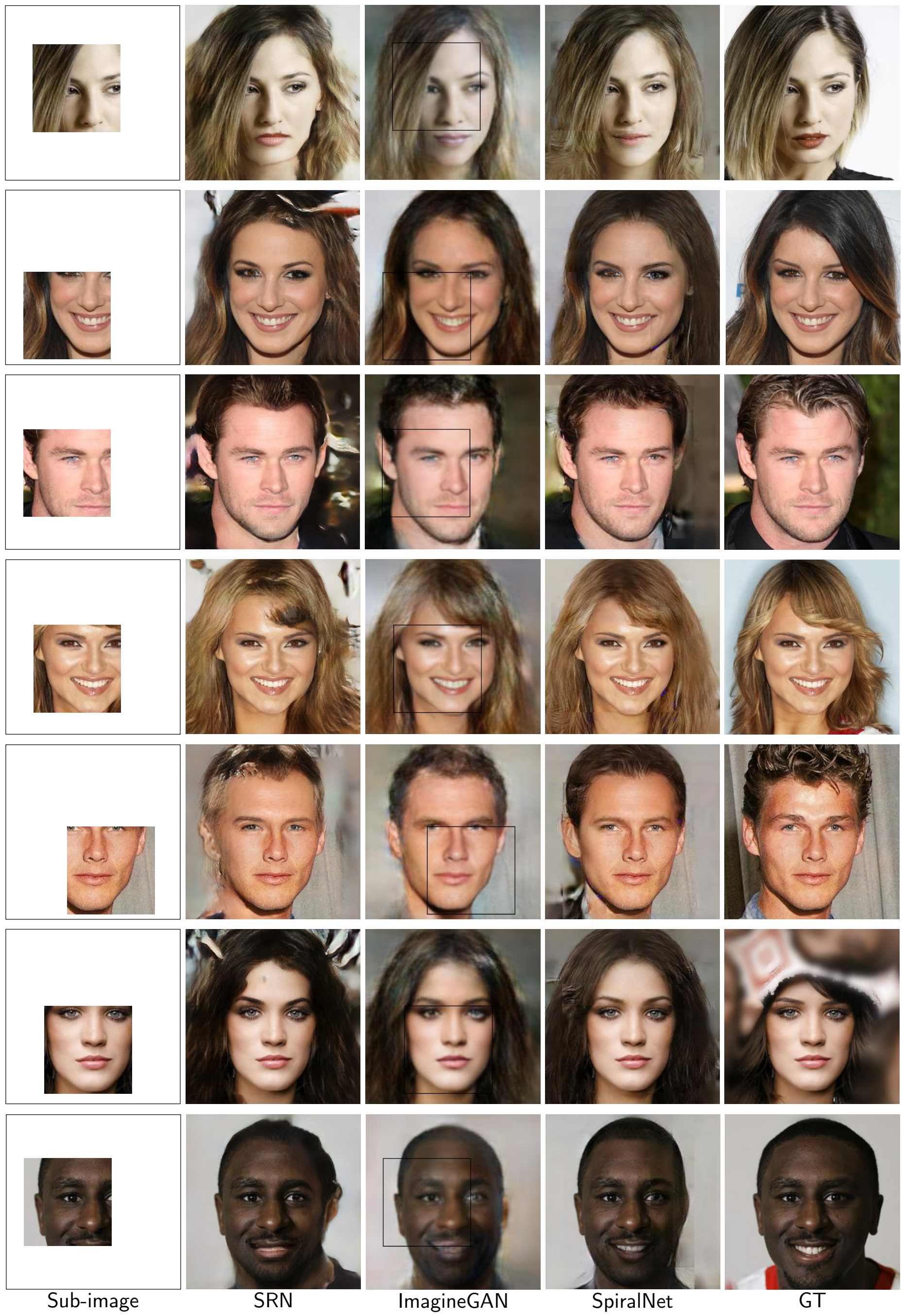


Fig. 9. Additional qualitative results of unknown margin case on CelebA-HQ.

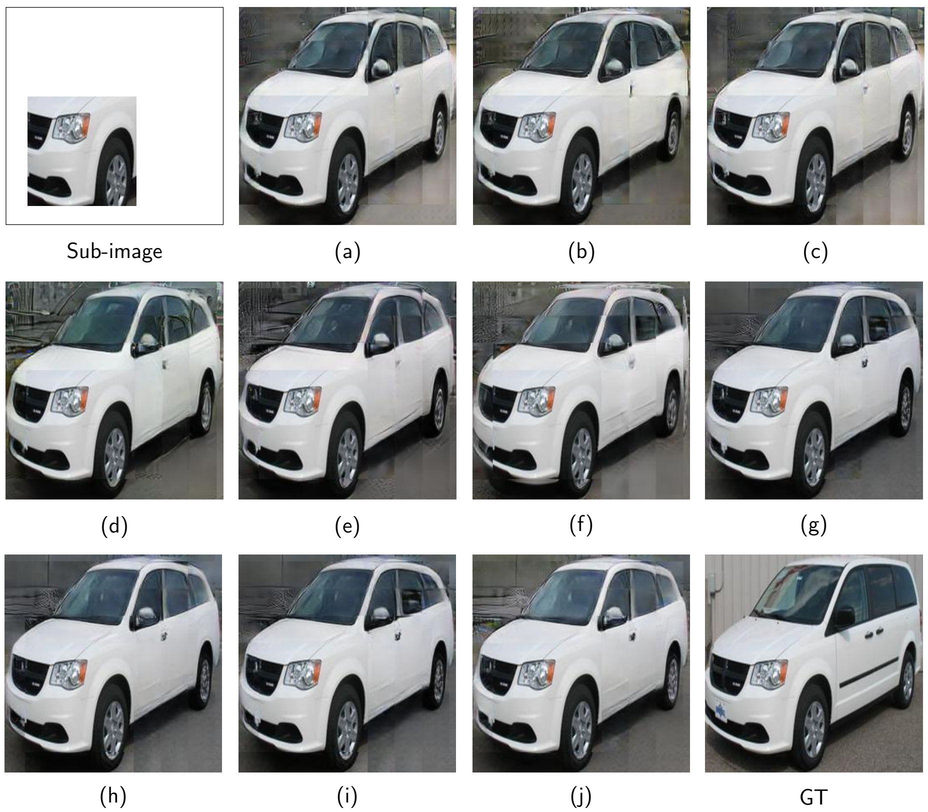


Fig. 10. Additional qualitative results on why spiral is necessary. (a) A.one-by-one. (b) A.horizontal-vertical. (c) A.vertical-horizontal. (d) B.w/o closest slice. (e) B.w/ sub-image slice. (f) C.simultaneous. (g) C.horizontal-vertical. (h) C.vertical-horizontal. (i) SpiralNet.anticlockwise. (j) SpiralNet.clockwise.

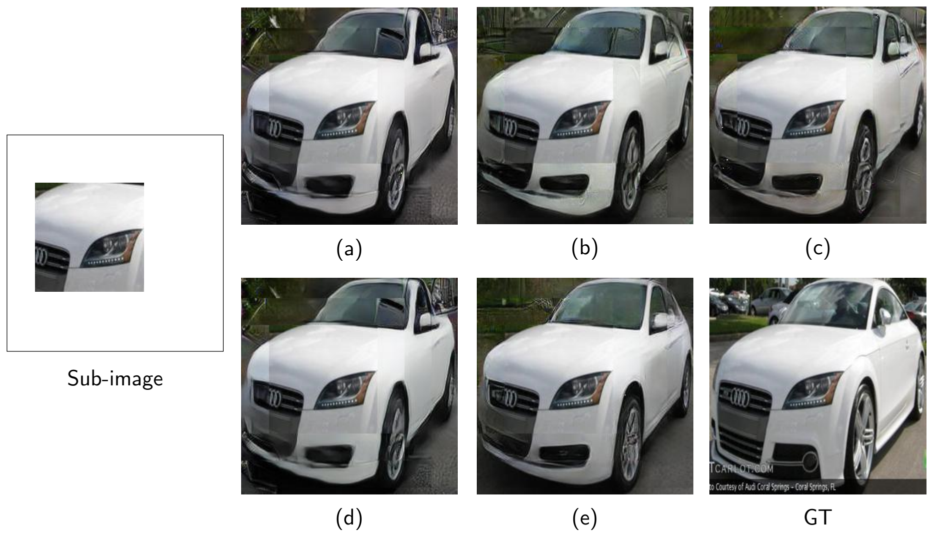


Fig. 11. Additional qualitative ablation study on ternary SliceGAN inputs. (a) Baseline. (b) w/ sub-image. (c) w/ closest slice. (d) exchange imaginary and closest slices. (e) SpiralNet.

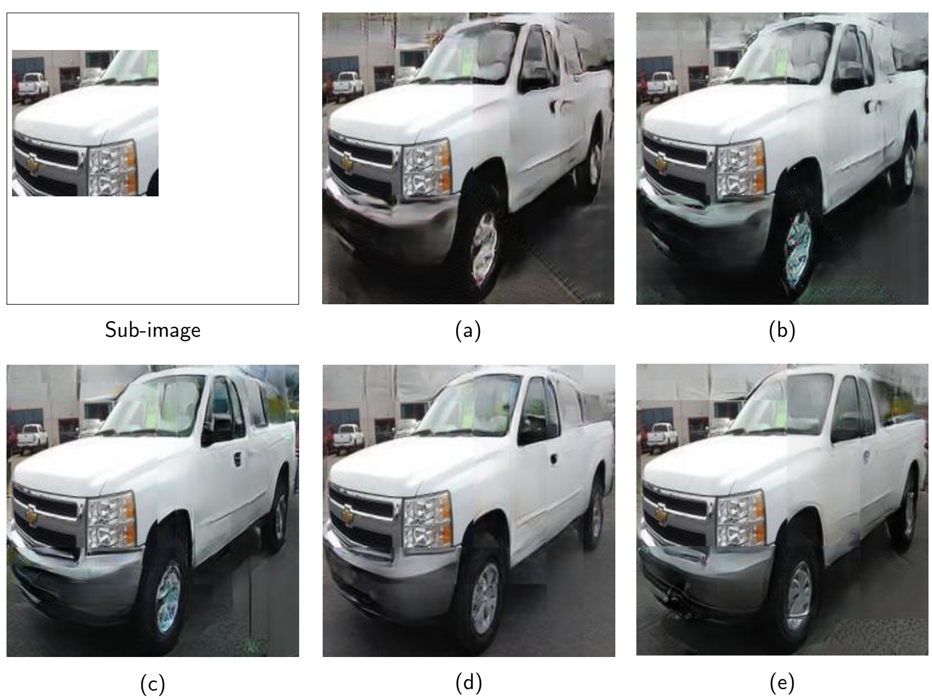
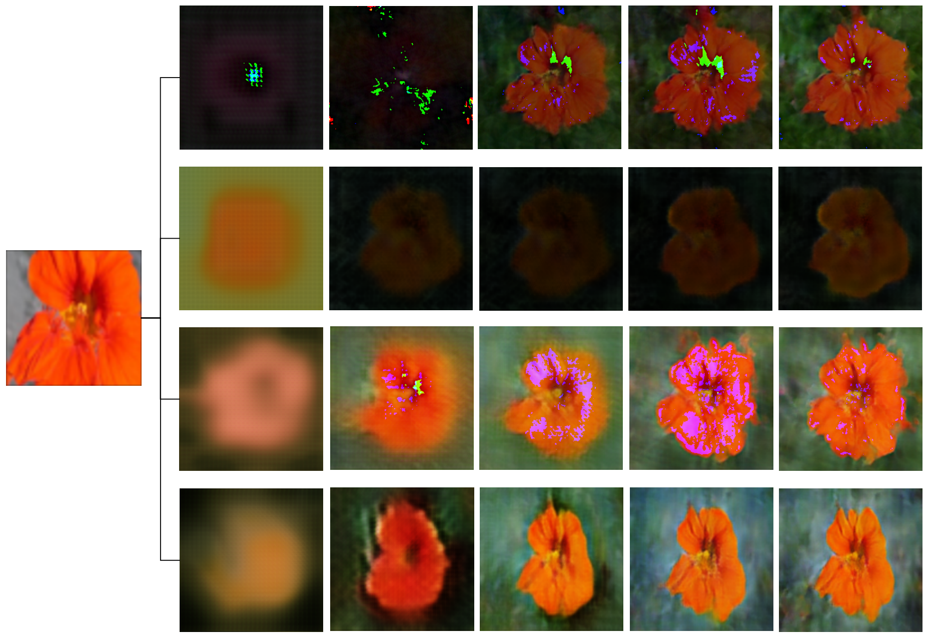


Fig. 12. Additional qualitative ablation study on different slice sizes. (a) $\tau = 4$. (b) $\tau = 8$. (c) $\tau = 16$. (d) $\tau = 32$. (e) $\tau = 64$.



(a)

(b)

Fig. 13. Additional comparison results of ImagineGAN with different losses on Flowers and Stanford Cars. (a) and (b): Qualitative results on Flowers and Stanford Cars: baseline, with color loss, with L1 loss, and with hue-color loss, from top to bottom.