AdaAttN: Revisit Attention Mechanism in Arbitrary Neural Style Transfer (Supplementary Material)

1. Network Details

1.1. Decoder

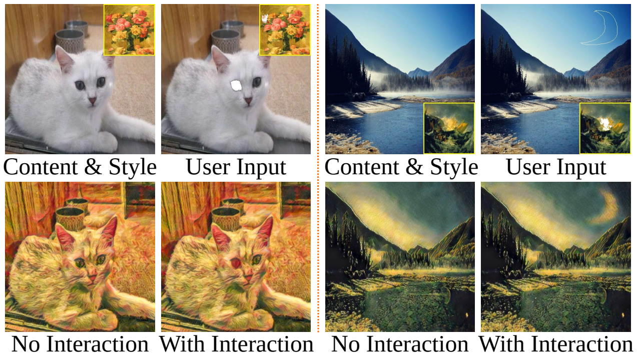
The decoder of our framework takes results of three AdaAttN modules on *ReLU-3_1*, *ReLU-4_1*, and *ReLU-5_1* layers as input. Similar to the decoder of SANet, feature on *ReLU-5_1* is upsampled to the same size as that of *ReLU-4_1*, followed by element-wise addition. Then, there is a learnable 3×3 convolution block for feature transformation. The following architecture is symmetrical with VGG encoder (up to *ReLU-4_1*), except that the number of input channels is twice on *ReLU-3_1* layer in order to incorporate AdaAttN output on this level. Full decoder configuration is shown in Table 1.

1.2. AdaAttN

We provide *PyTorch* code of AdaAttN module. The implementation is elegant and its overall time and space complexities are the same as SANet.

```
class AdaAttN(nn.Module):
   def __init__(self, v_dim, qk_dim):
      super().__init__()
      self.f = nn.Conv2d(qk_dim, qk_dim, 1)
      self.g = nn.Conv2d(qk_dim, qk_dim, 1)
      self.h = nn.Conv2d(v_dim, v_dim, 1)

   def forward(self, c_x, s_x, c_1x, s_1x):
      Q = self.f(mean_variance_norm(c_1x))
      Q = Q.flatten(-2, -1).transpose(1, 2)
      K = self.g(mean_variance_norm(s_1x))
      K = K.flatten(-2, -1)
      V = self.h(s_x)
      V = V.flatten(-2, -1).transpose(1, 2)
      A = torch.softmax(torch.bmm(Q, K), -1)
      M = torch.bmm(A, V)
      Var = torch.bmm(A, V ** 2) - M ** 2
      S = torch.sqrt(Var.clamp(min=0))
      M = M.transpose(1, 2).view(c_x.size())
      S = S.transpose(1, 2).view(c_x.size())
      return S * mean_variance_norm(c_x) + M
```



Content & Style User Input Content & Style User Input

No Interaction With Interaction No Interaction With Interaction

Figure 1. User controlling demos.

Stage	Output	Architecture
F^5	$512 \times \frac{H}{8} \times \frac{W}{8}$	Input F_{cs}^5 Upsample, scale 2 Add F_{cs}^4 3×3 Conv, 512, ReLU
F^4	$256 \times \frac{H}{4} \times \frac{W}{4}$	3×3 Conv, 256, ReLU Upsample, scale 2
F^3	$128 \times \frac{H}{2} \times \frac{W}{2}$	Concatenate F_{cs}^3 (3×3 Conv, 256, ReLU)×3 3×3 Conv, 128, ReLU Upsample, scale 2
F^2	$64 \times H \times W$	3×3 Conv, 128, ReLU 3×3 Conv, 64, ReLU Upsample, scale 2
F^1	$3 \times H \times W$	3×3 Conv, 64, ReLU 3×3 Conv, 3

Table 1. Architecture of our decoder network.

2. More Results

2.1. Image Style Transfer

User Control. Our method can support user-controlled stylization conveniently. User-specified content regions would adopt features of user-specified style regions by manipulating attention map used in AdaAttN module. In practice, user can either choose points on content and style images by mouse click (*e.g.*, Figure 1 (left)), or outlining re-

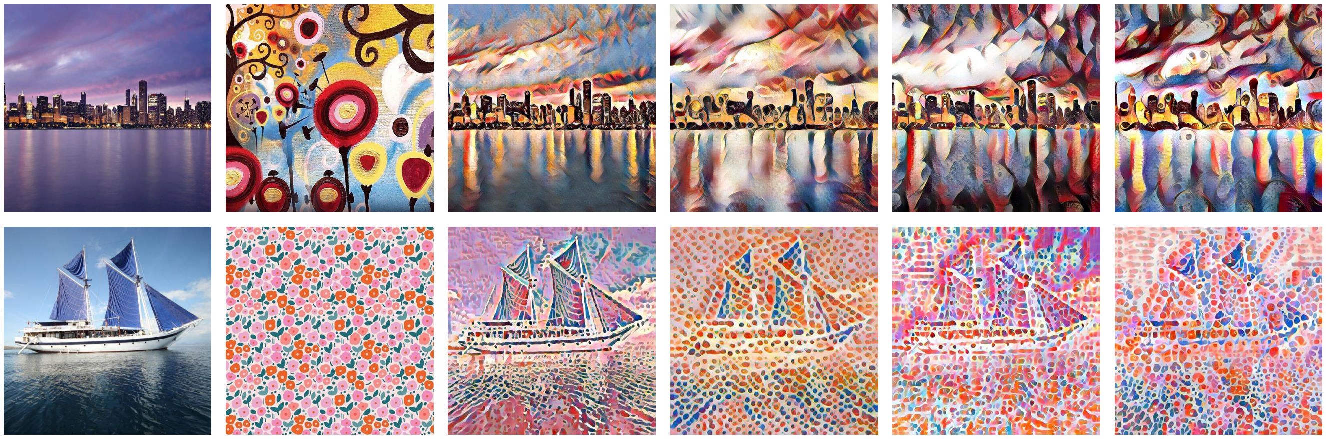

Content Style AdaAttN (Ours) Ours w/o shallow feature SANet with shallow feature SANet

Figure 2. More ablation study results.

Method	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
SANet	8.57	8.93	10.3	4.66	12.4	4.39	9.06	5.31	10.6	11.5	12.0	3.29	8.92	5.82	7.58	8.40	5.84	4.97	4.51	8.21
Linear	4.41	5.10	5.24	2.67	6.73	2.68	6.69	2.19	5.03	7.80	7.50	1.90	4.84	3.69	4.59	4.35	2.75	3.13	2.98	4.03
MCCNet	4.63	4.84	5.48	2.35	6.92	2.39	8.26	2.72	5.75	**6.70**	7.34	1.93	4.16	3.64	4.47	4.25	3.05	2.94	2.84	4.29
Ours	5.65	5.77	6.41	3.39	8.36	4.00	7.08	4.78	6.73	8.76	8.48	2.61	6.16	4.38	5.55	6.00	3.55	3.75	3.55	5.37
Ours + Cos	4.09	4.59	5.15	2.26	6.59	**2.24**	**5.97**	**2.06**	4.89	7.45	7.27	1.70	4.43	3.35	4.06	**3.94**	2.53	2.78	2.68	3.69
Ours + $\mathcal{L}_{is}$	5.51	5.31	6.26	3.31	7.96	4.56	6.84	5.03	6.37	8.90	8.55	2.62	6.15	4.82	5.30	6.29	3.66	4.01	3.63	5.05
Ours + Cos + $\mathcal{L}_{is}$	**3.70**	**4.46**	**4.49**	**2.14**	**6.06**	2.52	6.24	2.15	**4.55**	7.35	**7.11**	**1.60**	**3.86**	**3.27**	**3.85**	4.03	**2.31**	**2.54**	**2.44**	**3.43**

Table 2. Full results of optical flow error evaluation for 20 styles.

gions with closed borders (*e.g.*, Figure 1 (right)). Then, user-specified regions for content and style images can be generated by means of the classical region growing algorithm. To achieve use-controlled stylization, simply setting the attention scores between specified content regions and out of interest style regions as $-\infty$ before the Softmax operation in AdaAttN can work very well.

More Ablation. As discussed in our main paper, there are two factors leading to distorted stlylization of SANet: absence of low-level feature and failure in distribution alignment. To further illustrate impacts of these factors, we conduct more ablation studies under four settings: (1) AdaAttN, (2) AdaAttN without shallow features, (3) SANet with shallow features, and (4) SANet. As shown in Figure 2, both shallow features and feature distribution alignment prevent dirty textures to some extent. Combining them together, AdaAtN in this paper receives the best stylization results with the least distortion.

Pair-wise Combination between Content and Style. In order to demonstrate the robustness of our method on different contents and styles, we provide stylization results of pair-wise combinations between 8 content images and 6 style images in Figure 3. It can be seen that our AdaAttN can robustly achieve appealing style transfer results.

2.2. Video Style Transfer

Quantitative Results. Optical flow errors of the full 20 styles[1] used for video stylization are shown in Table 2, as supplement to Table 2 of our main paper.

Qualitative Results. We provide more video style transfer examples in Figure 4. Full animations can be found in the attachments.

---

[1] They are from official codebase of AdaIN.

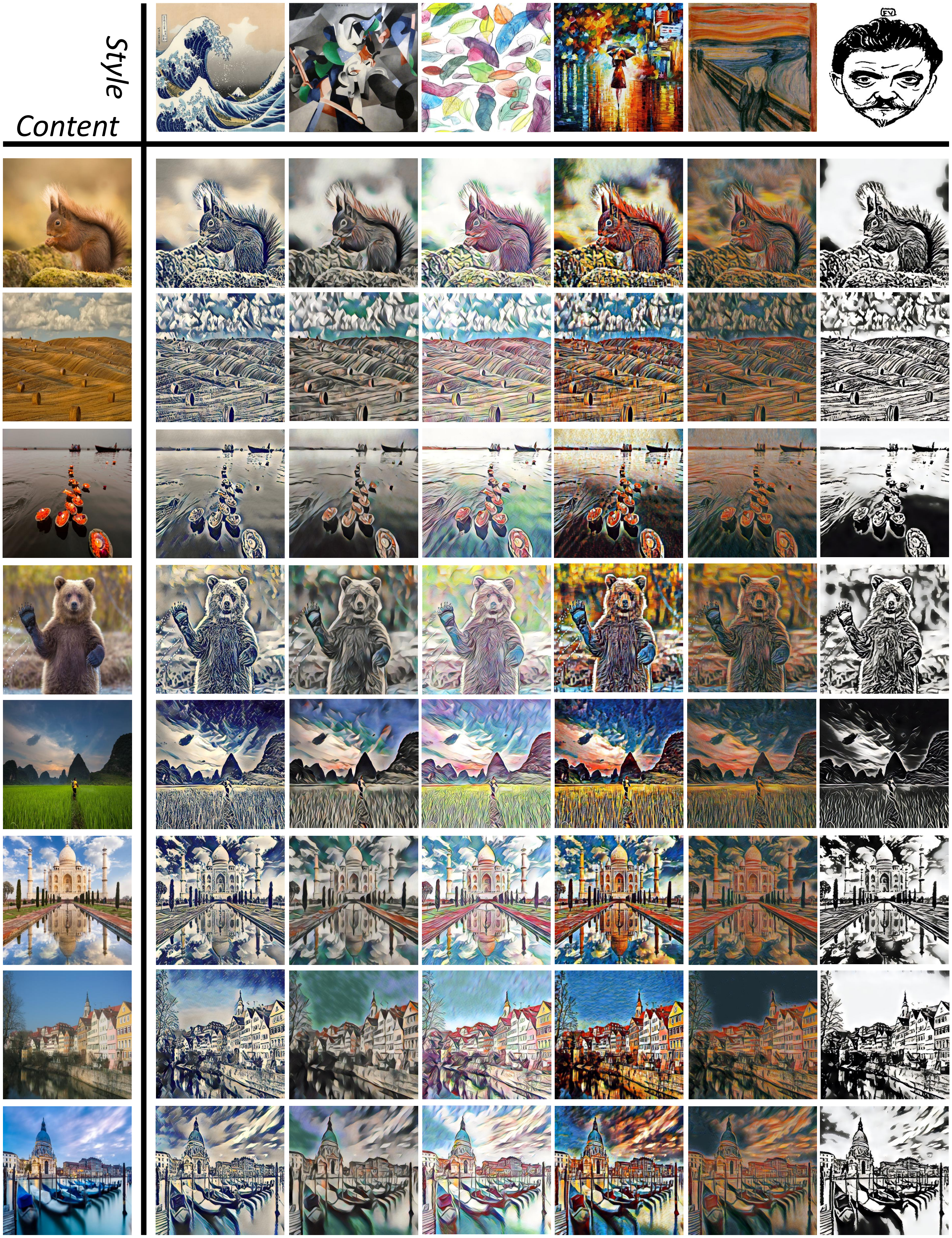


Figure 3. More image style transfer results.

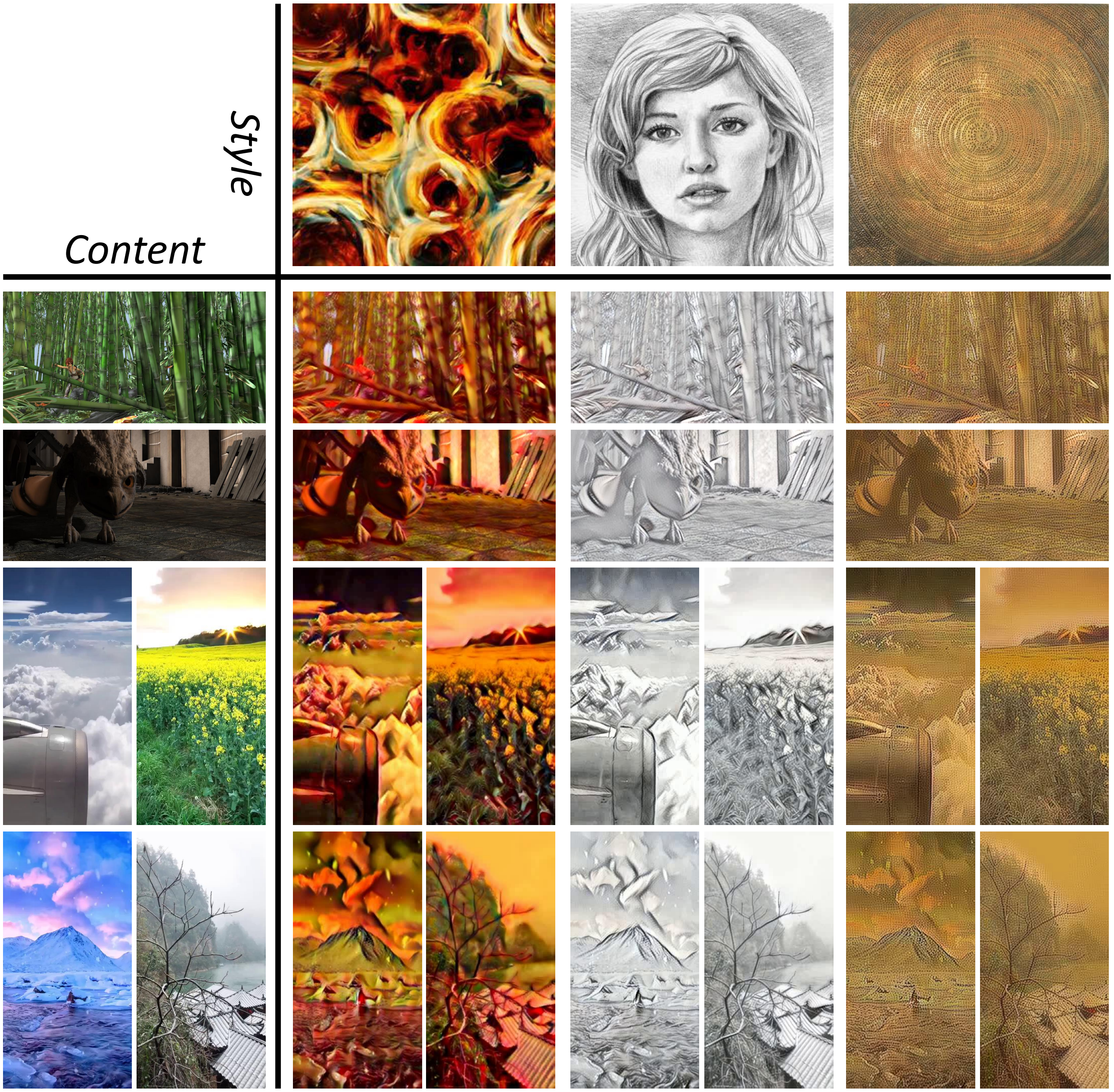


Figure 4. Snapshots of video stylization results. Full videos can be found in the attachments.