CCPL: Contrastive Coherence Preserving Loss for Versatile Style Transfer Supplemental File

Zijie Wu[1*], Zhen Zhu[2*], Junping Du[3], and Xiang Bai[1†]

[1] Huazhong University of Science and Technology, China
[2] University of Illinois at Urbana-Champaign, USA
[3] Beijing University of Posts and Telecommunications, China
{zijiewu,xbai}@hust.edu.cn, zhenzhu4@illinois.edu, junpingdu@126.com

1 Details of CCPL

We summarize the detailed execution of the proposed CCPL in Alg. 1. Please refer to our code in https://github.com/JarrentWu1031/CCPL for further reference.

Algorithm 1 Contrastive Coherence Preserving Loss (CCPL)

Input:

f_g	feature map encoded by the generated image (spatial size : $H \times W$)
f_c	feature map encoded by the content image (spatial size : $H \times W$)
N	number of sampled anchor vectors per layer

Output:

L_{ccp}	CCPL

$G_a,\ W_a,\ H_a = Random\_Sample(f_g, W_a \in [2, W-1], H_a \in [2, H-1], N)$;
// randomly sample N anchor vectors from f_g

$C_a = Sample(f_c, W_a, H_a)$; // sample f_c at the same locations

$G_n = Neighbor\_Sample(G_a),\ C_n = Neighbor\_Sample(C_a)$;
// sample all the eight nearest neighboring vectors

$d_g^{x,y} = G_a^x - G_n^{x,y},\ d_c^{x,y} = C_a^x - C_n^{x,y}$; // compute the difference vectors

$d_g = MLP(d_g),\ d_c = MLP(d_c)$; // MLP is a two − layer perceptron

$d_g = l_2\_norm(d_g),\ d_c = l_2\_norm(d_c)$; // l_2 normalization

$L_{\text{ccp}} = \sum_{m=1}^{8\times N} -\log[\frac{\exp(d_g^m \cdot d_c^m/\tau)}{\exp(d_g^m \cdot d_c^m/\tau) + \sum_{n=1, n\neq m}^{8\times N} \exp(d_g^m \cdot d_c^n/\tau)}]$.
// compute InfoNCE [7] loss

∗ Equal contribution † Corresponding author

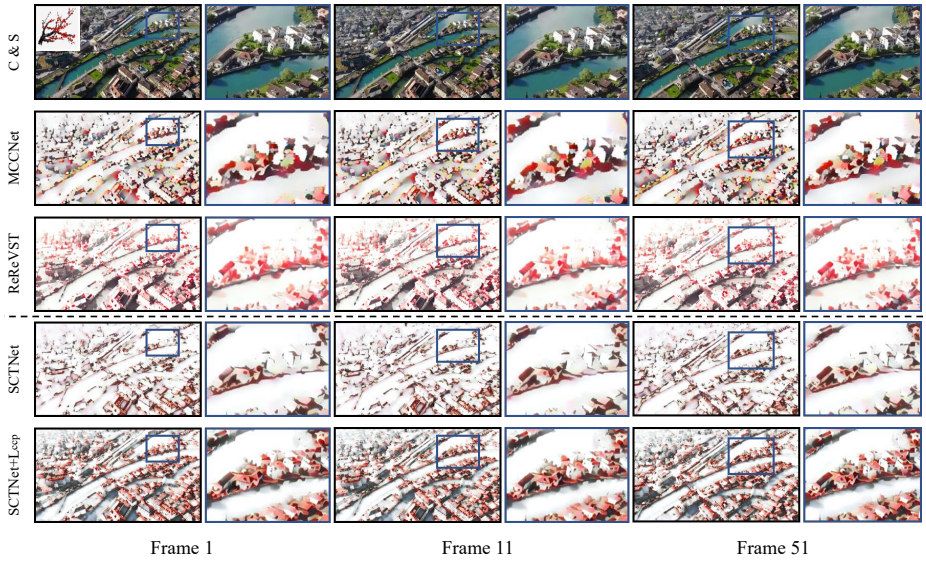

Fig. 1. Qualitative comparison of long-term temporal consistency. We compare our method with the SOTA single-frame (MCCNet [2]) and multi-frames (ReReVST [10]) methods. At the bottom of the figure, we present the frame indices of the 24FPS video.

2 Additional Qualitative Results

Image style transfer. Additional results for artistic and photo-realistic image style transfer are shown in Fig. 6 and Fig. 7. We compare our method with the same algorithms presented in the main paper. Both figures suggest the significant visual quality improvements brought by CCPL and the superior stylization ability of our full model, which are consistent with the conclusions drawn in the main paper.

3 Additional Quantitative Results

Photo-realistic style transfer. We add excution speed comparisons with the SOTA photo-realistic methods, shown in Tab. 1. Our method surpasses these SOTAs in inference speed, indicating the potential for real-world applications.

Video style transfer. As for video style transfer, we compare the long-term temporal consistency of our results with the recent SOTA single-frame method (MCCNet [2]) and multi-frames method (ReReVST [10]). The results in Fig. 1 show better long-term temporal consistency of video results when trained with CCPL. And our full model generates results better than the SOTA single-frame method (MCCNet [2]) and is on par with the most advanced multi-frames method (ReReVST [10]) in long-term temporal consistency with single-frame content sources. Moreover, the results generated by MCCNet [2] and

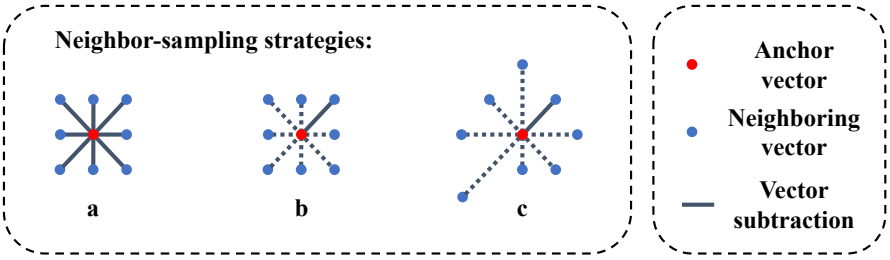

Fig. 2. Three different neighbor-sampling strategies of CCPL. In the main paper, we adopt strategy a as the default setting.

Fig. 3. Results with different neighbor-sampling strategies of CCPL.

ReReVST [10] suffer from unreasonable colorization (green color in Fig. 1b) and detail loss (vanished house contours in Fig. 1b, c). For further demonstration, we provide a video demo to show the effect of CCPL on improving temporal consistency. The name of the video demo is CCPL_TC.mp4. We attach it in the folder: *Video demo*. Note that we also release the questionnaire of our user study in folder *user study*. The questionnaire will better understand how we conducted the user study.

4 Additional ablation studies

4.1 Neighbor-sampling strategies

We show three different neighbor-sampling strategies in Fig. 2. Here, a represents our default sampling strategy which samples all 8 neighbors around the anchor vector. Instead of taking all neighbors, b randomly sample one neighbor out of eight for the anchor vector. We think of strategy b since it will show whether we can further decrease the training budget. Different from b, c randomly sample one neighboring vector where its vertical and horizontal shifts about the anchor vector lie within the integer range of $[0, R]$. R is a natural number. Actually, b can be regarded as a special case of c when R equals 1. Verifying the effect of c can show if a more distant neighbor is beneficial to the outcome. To compare the above neighbor-sampling strategies of CCPL, we sample 64 anchor-neighboring vector pairs from each layer ($relu2\_1$, $relu3\_1$, $relu4\_1$) using these strategies, respectively. To be noted, since strategy a samples every nearest neighboring

Table 1. Execution speed comparison (unit: FPS). We use a single 12GB Titan XP GPU for all the execution time testing. OOM denotes the Out-Of-Memory error.

Photo-realistic	LST [5]	WCT^2 [11]	SNAS [1]	DSTN [3]	Ours
512×256	26.3	4.6	5.6	10.0	**31.3**
1024×512	8.3	2.3	2.3	2.7	**9.5**
2048×1024	3.6	OOM	OOM	0.6	**3.7**

Fig. 4. Qualitative results of ablation studies on weight ratio of CCPL with the style loss.

vector (8 vectors) of the anchor vector, we only sample 8 anchor vectors each layer for fairness. For b and c, the numbers of anchor vectors are both 64. And the neighbor-sampling ranges R for c are set to {4, 2, 1} for {$relu2\_1$, $relu3\_1$, $relu4\_1$}, respectively.

As shown in Fig. 3, the body color of the bus is scattered in b and c. In contrast, the result from strategy a shows consistent transferred color, which suggests its superior ability to maintain the coherence of the content source. We attribute the reason to more informative negative examples when calculating CCPL. For strategy a, the difference vectors generated by the anchor vector and its eight nearest neighboring vectors are closer related than other random difference vectors, which give more information of the neighbor contrast, thus leading to better-preserved coherence of the content source. Similar viewpoints are found in [9]. Therefore we choose a as the default neighbor-sampling strategy for CCPL.

4.2 Qualitative comparisons of CCPL

In the main paper, we only show the quantitative comparisons of CCPL with different settings of three factors: 1) layers to apply the loss; 2) the number of difference vectors sampled each layer; 3) the loss weight ratio with the style loss. Here we also provide the qualitative results to show how we find the default settings, which are demonstrated in Fig. 4 and Fig. 5.

As depicted in Fig. 4, the generated image is cleaner with a higher CCPL/SL (SL: the style loss) weight ratio, while the colorization remains vivid. As the loss weight ratio increases to a certain extent (1.0), the visual quality of the generated image is close to the original content image with barely local style patterns. To balance the visual quality and stylization effects, we choose 0.5 as the default setting of the CCPL/SL weight ratio.

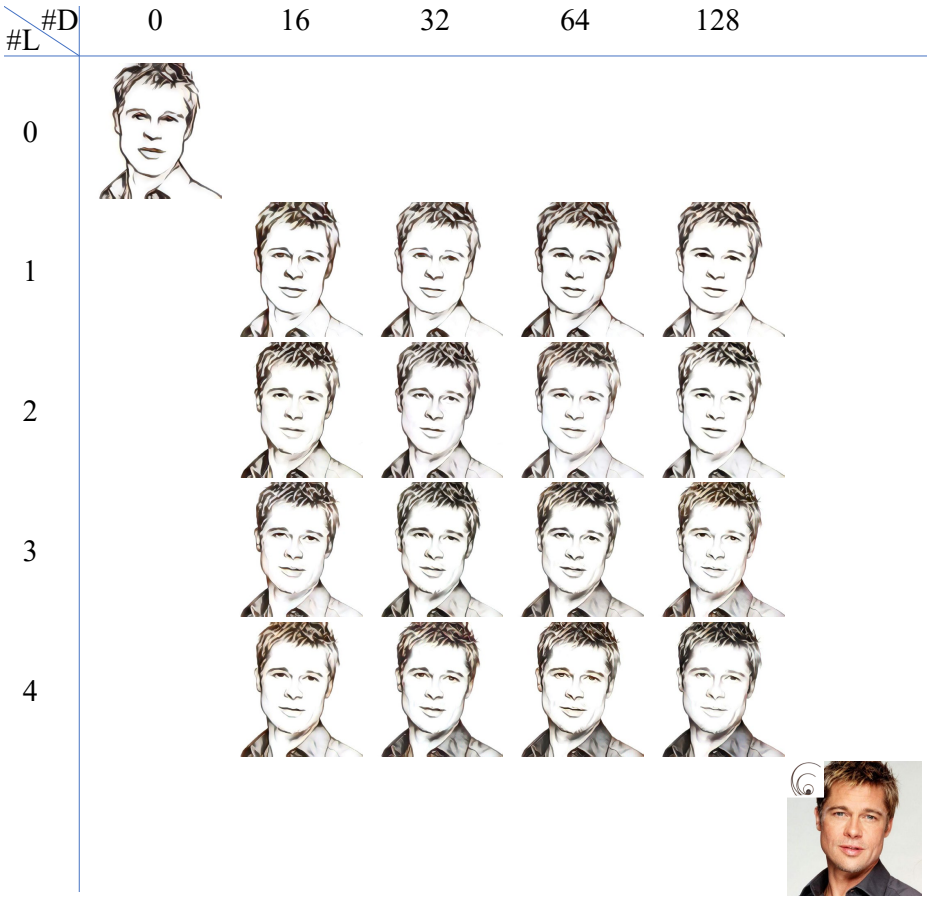

Fig. 5. Qualitative results of ablation studies on two factors of CCPL: 1) #L: the number of layers to apply CCPL; 2) #D: the number of difference vectors sampled for each layer. We present the content and style image in the lower right corner.

Once the CCPL/SL weight ratio is settled, we conduct several experiments by enumerating the number of CCPL layers from 0 to 4 (start from the deepest layer) and choosing from [16, 32, 64, 128] as the number of sampled combinations to show the impacts of the first two factors, as we described in the main paper. From Fig. 5 we can see that, as the number of layers to apply CCPL increases, the visual quality also gets better, with better fidelity to the content source. The conclusion also holds for the number of sampled difference vectors. However, when the number of CCPL layers turns to 4, the generated image is too close to the content input that it fails to demonstrate the bold style of the style source well. Therefore we calculate CCPL utilizing feature maps from $\{relu2\_1, relu3\_1, relu4\_1\}$ during training. As for the number of sampled difference vectors, the performances of 64, 128 are close while being better than 32, 16 in visual quality.

Considering the computation burden, we choose 64 as the number of difference vectors sampled for each layer.

5 Additional Applications

We apply the CCPL on several existing methods: AdaIN [4], SANet [8], Linear [5] (also illustrated in the main paper), and show more visual results in Fig. 8. Besides, we further demonstrate one more application of our full model: *high-resolution photo-realistic style transfer*.

Super-resolution photo-realistic style transfer. Since our SCTNet is light and fast while being able to transfer photo-realistic styles when trained with CCPL, our full model is appropriate for performing super-resolution photo-realistic style transfer. We use the same graphic card: Titan XP to synthesize all the examples below. The results are shown in Fig. 9 - Fig. 10, with the image resolution nearly scaling up to 4K.

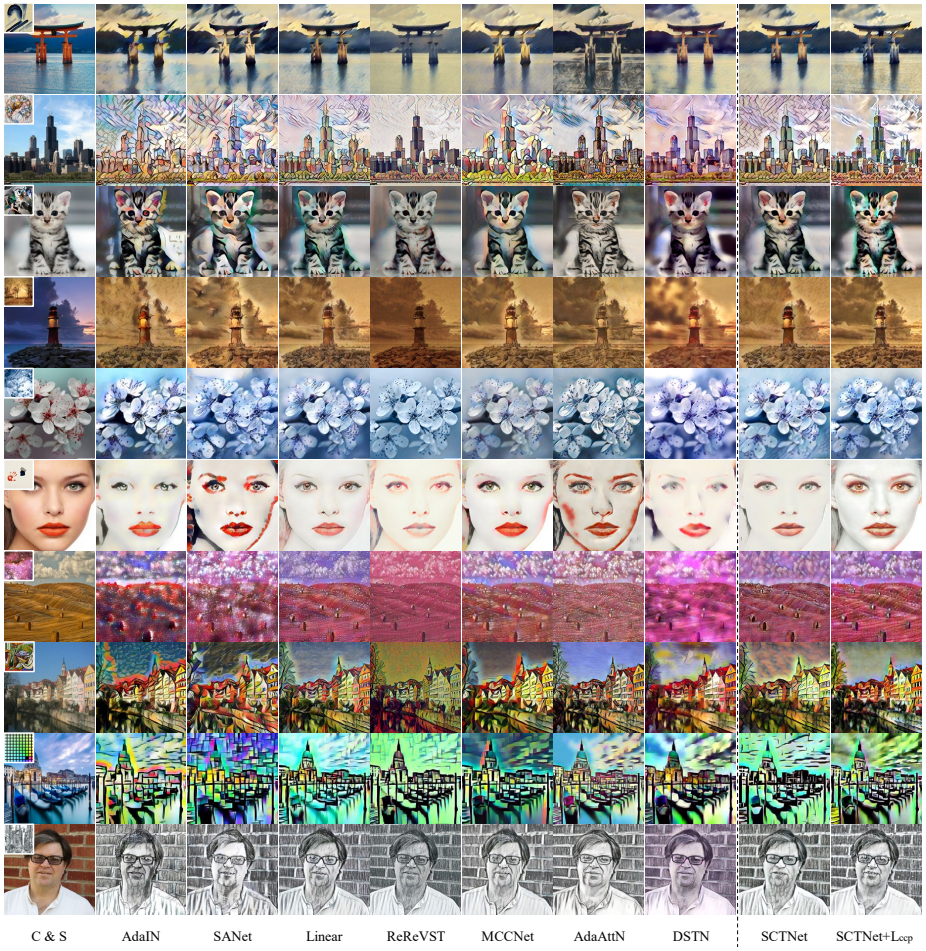

Fig. 6. Additional qualitative comparison of artistic style transfer. We compare our method with seven SOTAs: AdaIN [4], SANet [8], Linear [5], ReReVST [10], MCC-Net [2], AdaAttN [6], DSTN [3].

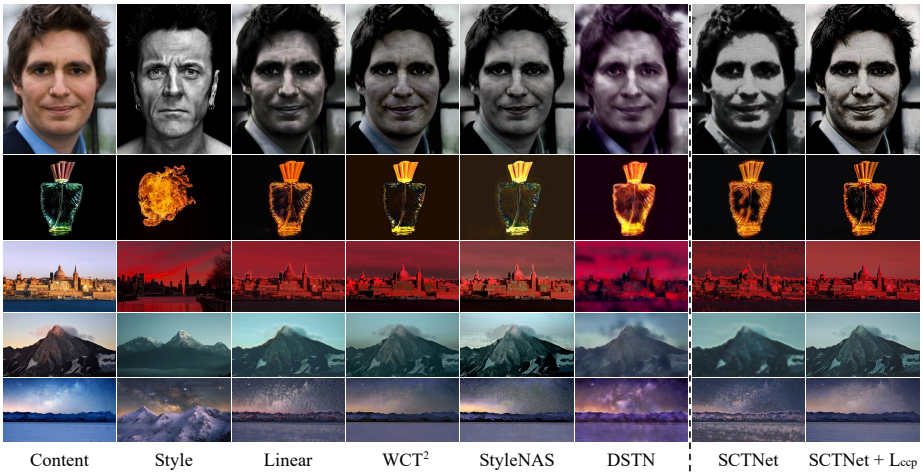

Fig. 7. Additional qualitative comparison of photo-realistic style transfer. We compare our method with four SOTAs: Linear [5], WCT^2 [11], StyleNAS [1], DSTN [3]. Zoom in for a better view.

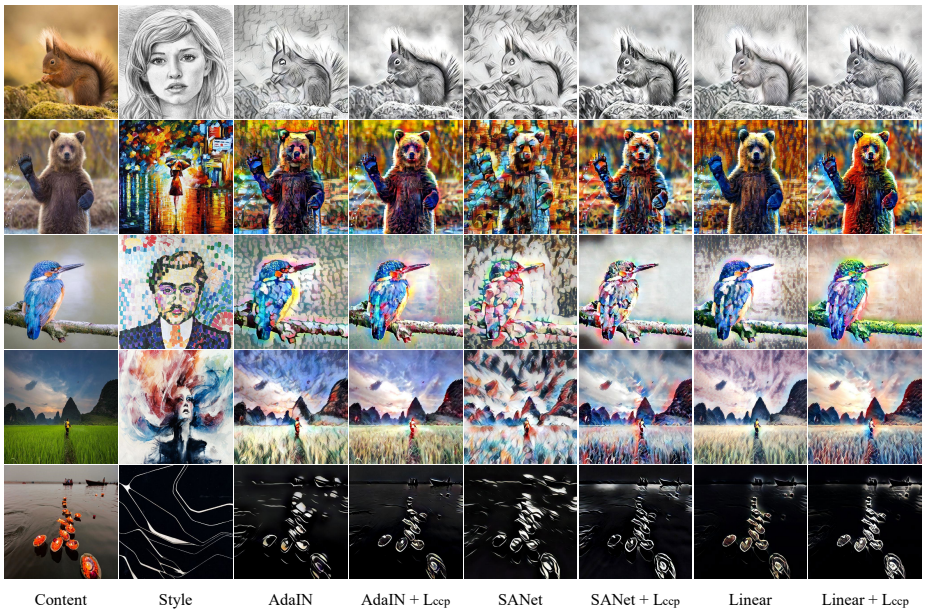

Fig. 8. Additional comparison of different results generated by models (AdaIN [4], SANet [8], Linear [5]) trained with or without CCPL.

Fig. 9. An example of High-resolution photo-realistic style transfer generated by SCT-Net trained with CCPL (resolution: 3640×2048).

Fig. 10. An example of High-resolution photo-realistic style transfer generated by SCT-Net trained with CCPL (resolution: 3276×2048).

References

1. An, J., Xiong, H., Ma, J., Luo, J., Huan, J.: Stylenas: An empirical study of neural architecture search to uncover surprisingly fast end-to-end universal style transfer networks. arXiv preprint arXiv:1906.02470 (2019)
2. Deng, Y., Tang, F., Dong, W., Huang, H., Ma, C., Xu, C.: Arbitrary video style transfer via multi-channel correlation. arXiv preprint arXiv:2009.08003 (2020)
3. Hong, K., Jeon, S., Yang, H., Fu, J., Byun, H.: Domain-aware universal style transfer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14609–14617 (2021)
4. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1501–1510 (2017)
5. Li, X., Liu, S., Kautz, J., Yang, M.H.: Learning linear transformations for fast image and video style transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3809–3817 (2019)
6. Liu, S., Lin, T., He, D., Li, F., Wang, M., Li, X., Sun, Z., Li, Q., Ding, E.: Adaattn: Revisit attention mechanism in arbitrary neural style transfer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6649–6658 (2021)
7. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
8. Park, D.Y., Lee, K.H.: Arbitrary style transfer with style-attentional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5880–5888 (2019)
9. Wang, F., Liu, H.: Understanding the behaviour of contrastive loss. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2495–2504 (2021)
10. Wang, W., Yang, S., Xu, J., Liu, J.: Consistent video style transfer via relaxation and regularization. IEEE Transactions on Image Processing **29**, 9125–9139 (2020)
11. Yoo, J., Uh, Y., Chun, S., Kang, B., Ha, J.W.: Photorealistic style transfer via wavelet transforms. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9036–9045 (2019)