Learning Blind Video Temporal Consistency Supplementary Material

Wei-Sheng Lai[1] Jia-Bin Huang[2] Oliver Wang[3] Eli Shechtman[3]
Ersin Yumer[4] Ming-Hsuan Yang[1,5]

[1]UC Merced [2]Virginia Tech [3]Adobe Research [4]Argo AI [5]Google Cloud

1 Overview

In this supplementary document, we present additional results to complement the paper. We first provide the implementation and training details of the proposed model. We then analyze the performance contribution of several key designs in the proposed model. More visual comparisons with the state-of-the-art method are provided on our project website http://vllab.ucmerced.edu/wlai24/video_consistency.

2 Implementation details

We implement our model using PyTorch [10]. We use a kernel of size 7×7 for the first and the last convolutional layers and 3×3 for all other convolutional and transposed convolutional layers. The number of filters is 32 and is multiplied by 2 when the feature maps are downsampled. All the convolutional and transposed convolutional layers (except the last layers) are followed by the instance normalization [13] and leaky ReLUs (LReLU) [9] with a negative slope of 0.2. There are 5 residual blocks between the encoder and decoder. At the end of the decoder, we use a Tanh layer to constrain the range of the output into $[-1, 1]$.

During training, we use a batch size of 4 (i.e., 4 sequences). For each sequence, we sample 10 consecutive frames, which means that the long-term temporal coherence is enforced over a maximum of 10 frames. We run the forward pass of all 10 frames before updating the network parameters. We randomly crop video frames to 192×192 and apply the data augmentation of random scaling between $[1, 2]\times$, random rotation for 90°, 180° or 270°, and horizontal flipping. The same geometric transform is applied to all the frames in the same video. We also adopt a temporal augmentation by reversing the order of sequences. The initial learning rate is set to $1e-4$ and decreased by a factor of 2 for every 20,000 iterations. We train our model with the ADAM solver [6] for 100,000 iterations. During the training phase, only the image transformation network is updated while the FlowNet and VGG are fixed.

3 Additional Analysis

We conduct experiments to provide more understanding on the effect of the temporal loss, perceptual loss, and the ConvLSTM layer. We also analyze the

effect of L_1 and L_2 norm and compare the results of multi-task and single-task training. Finally, we show a failure case of the proposed method.

3.1 Effect of temporal and perceptual losses

Our training objective function is a combination of the content perceptual loss L_p, short-term temporal L_{st}, and long-term temporal losses L_{lt}. To further analyze the effect of each loss function, we train three models by setting the weights of each loss term, λ_p, λ_{st}, and λ_{lt}, to 0, respectively. We evaluate the performance of the variants using the WCT method [8] on the DAVIS test set [12] and provide quantitative comparisons in Fig. 1.

Without perceptual loss. The model trained without the perceptual loss generates blurry results. While a blurry video tends to have a low temporal warping error, the perceptual distance is large, indicating that the model cannot preserve the content of the processed video well.

Without short-term temporal loss. Without the short-term temporal loss, the model cannot reduce the temporal flickering well. The temporal warping error is close to that of the processed video in Fig. 1.

Without long-term temporal loss. When training without the long-term temporal loss, the model does not capture the long-term temporal coherence well and thus is prone to error propagation and occlusion. As shown in Fig. 2(e), the blue regions on the ground suddenly change into different colors after a man passing by. On the contrary, the model trained with all the losses produces stable results without temporal flickering.

Method	E_{warp}	$D_{\text{perceptual}}$
V_p	0.054	0
Bonneel et al. [2]	0.0312	0.0977
Ours w/o L_p	0.0222	0.1850
Ours w/o L_{st}	0.0518	0.0063
Ours w/o L_{lt}	0.0427	0.0132
Our full model	0.0348	0.0194

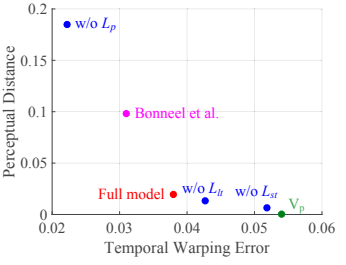


Fig. 1: **Analysis on loss functions**. (Left) We analyze the contribution of each loss by setting the weight of each term to 0, respectively. (Right) The trade off between perceptual similarity and temporal warping with different loss functions, as compared to Bonneel et al. [2], and the original processed video, V_p.

(a) Input video (b) Stylized video

(c) Without perceptual loss (d) Without short-term temporal loss

(e) Without long-term temporal loss (f) Ours

Fig. 2: **Effect of loss functions.** Without the perceptual content loss, the results are overly smooth and have a low perceptual similarity with the processed video. While the short-term temporal loss is crucial to remove the high-frequency flickering, the long-term temporal loss further reduces low-frequency jitter and avoids error propagation (e.g., the lower-right corner in (e)). This figure contains *animated videos*, which are best viewed using Adobe Acrobat.

3.2 Effect of LSTM

To analyze the effect of the ConvLSTM layer, we train an image transformation network without the ConvLSTM layer. To use the same amount of network parameters, we increase the number of residual blocks from 5 to 9 in this model. We show an example of stabilizing the results of a colorization method [4] on the VIDEVO dataset in Fig. 3. The model without the ConvLSTM layer produces propagation errors (as shown on the ground of Fig. 3(c)). Our model with the ConvLSTM layer successfully captures the spatio-temporal correlation of the original input video and produces more visually pleasing videos.

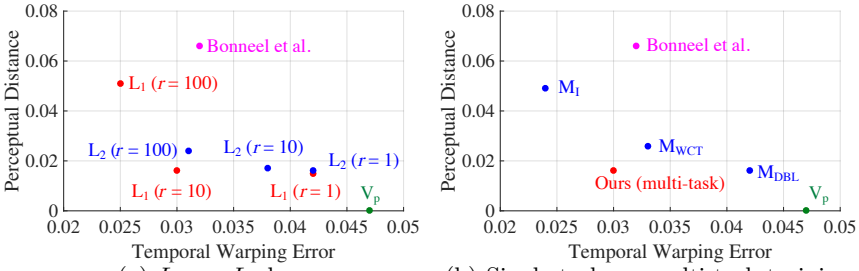
(a) Input video (b) Colorized video (frame-by-frame)

(c) Without ConvLSTM (d) With ConvLSTM

Fig. 3: **Effect of ConvLSTM layer.** The model trained without the ConvLSTM layer produces propagation errors, while our full model generates more visually pleasing videos. This figure contains *animated videos*, which are best viewed using Adobe Acrobat.

3.3 L_2 norm v.s. L_1 norm

We choose to use the L_1 loss as it is a robust loss function commonly used in several vision tasks, e.g., super-resolution [7] and inpainting [11]. However, we find that the choice of the loss function is not crucial in the proposed model. Here we train our model using the L_2 loss for computing the content and temporal losses and show the trade-off curve in Fig. 4(a). When setting $r = 100$, the model using the L_2 loss performs similarly to that using the L_1 loss function.



(a) L_1 vs. L_2 loss (b) Single-task vs. multi-task training

Fig. 4: **Analysis on loss function and multi-task training.**

The model optimized with the L_2 loss can achieve comparable performance as our current model with a proper weights setting, i.e., adjusting $r = \lambda_t/\lambda_p$.

3.4 Multi-task vs. single-task training

We train three single-task models using one style image for the WCT [8] (denoted by M_{WCT}), one enhancement model of the DBL [2] (denoted by M_{DBL}), and the shading layer of the intrinsic decomposition algorithm [1] (denoted by M_{I}), respectively. We evaluate the temporal warping error and perceptual distance on the DAVIS test set and present the detailed results in Table 1. We also show the trade-off curve between the average warping error and perceptual distance in Fig. 4(b). It is interesting that the single-task models do not always achieve the lowest temporal warping error and perceptual distance on the same task used in training. As the single-task training is susceptible to overfitting for the specific task, the single-task models may generate more artifacts and do not generalize well to multiple tasks. In contrast, the multi-task model maintains small temporal warping error and has the lowest perceptual distance.

3.5 Failure case

While our experimental results show that the proposed recurrent network performs well on a variety of videos and generalizes well to multiple tasks, we do observe some failure cases as shown in Fig. 5, where the brown color in the mountain region is wrongly propagated to the sky. Using more videos and tasks for training might be able to make our model more robust and reduce the error propagation.

Table 1: **Comparison of single-task and multi-task training.** The parentheses indicate that the task is used in training. We note that all the test videos are unseen during the training phase.

Task	Temporal warping error				Perceptual distance			
	M_{WCT}	M_{DBL}	M_{I}	Ours	M_{WCT}	M_{DBL}	M_{I}	Ours
WCT [8]/antimono	(0.037)	0.050	**0.028**	(0.035)	(0.029)	**0.012**	0.048	(0.019)
WCT [8]/asheville	0.061	0.080	**0.038**	0.055	0.023	**0.011**	0.055	0.019
WCT [8]/candy	0.047	0.063	**0.034**	(0.045)	0.030	**0.013**	0.065	(0.023)
WCT [8]/feathers	0.033	0.045	**0.025**	0.029	0.025	**0.015**	0.047	0.016
WCT [8]/sketch	0.031	0.040	0.024	(**0.023**)	0.026	**0.016**	0.036	(0.021)
WCT [8]/wave	0.032	0.043	**0.019**	0.027	0.024	**0.014**	0.055	0.015
Fast-neural-style [5]/princess	0.049	0.064	**0.039**	0.047	0.046	0.031	0.076	**0.029**
Fast-neural-style [5]/udnie	0.043	0.060	**0.034**	0.042	0.028	**0.015**	0.039	0.017
DBL [3]/expertA	0.030	(0.035)	**0.026**	(0.028)	0.022	(0.016)	0.037	(**0.011**)
DBL [3]/expertB	0.028	0.031	**0.022**	0.025	0.021	0.015	0.040	**0.011**
Intrinsic [1]/reflectance	0.017	0.021	**0.015**	**0.015**	0.022	0.015	0.051	**0.013**
Intrinsic [1]/shading	**0.010**	0.015	(**0.009**)	(0.011)	0.032	0.021	(0.030)	(**0.017**)
CycleGAN [15]/photo2ukiyoe	0.028	0.033	**0.017**	0.026	0.021	0.015	0.052	**0.012**
CycleGAN [15]/photo2vangogh	0.031	0.036	**0.019**	0.029	0.024	0.017	0.067	**0.016**
Colorization [14]	0.024	0.028	**0.019**	0.024	0.022	0.015	0.047	**0.013**
Colorization [4]	0.024	0.027	**0.018**	(0.023)	0.021	0.015	0.044	(**0.011**)
Average	0.033	0.042	**0.024**	0.030	0.026	**0.016**	0.049	0.017

(a) Input frames (b) Processed frames

(c) Bonneel et al. [2] (d) Ours

Fig. 5: **Failure case.** The brown color in the mountain region is wrongly propagated to the sky. This figure contains animated videos, which are best viewed using Adobe Acrobat.

References

1. Bell, S., Bala, K., Snavely, N.: Intrinsic images in the wild. ACM TOG (2014)
2. Bonneel, N., Tompkin, J., Sunkavalli, K., Sun, D., Paris, S., Pfister, H.: Blind video temporal consistency. ACM TOG (2015)
3. Gharbi, M., Chen, J., Barron, J.T., Hasinoff, S.W., Durand, F.: Deep bilateral learning for real-time image enhancement. ACM TOG (2017)
4. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Let there be color!: Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. ACM TOG (2016)
5. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV (2016)
6. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
7. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: CVPR (2017)
8. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Universal style transfer via feature transforms. In: NIPS (2017)
9. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: ICML (2013)
10. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
11. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: CVPR (2016)
12. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR (2016)
13. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In: CVPR (2017)
14. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV (2016)
15. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV (2017)