Photorealistic Style Transfer via Wavelet Transforms

Jaejun Yoo* Youngjung Uh* Sanghyuk Chun* Byeongkyu Kang Jung-Woo Ha
Clova AI Research, NAVER Corp.
{jaejun.yoo,youngjung.uh,sanghyuk.c,bk.kang,jungwoo.ha}@navercorp.com

(a) Inputs (b) WCT (c) PhotoWCT (d) Ours (WCT^2)

Figure 1: Photorealistic stylization results. Given (a) an input pair (top: content, bottom: style), the results of (b) WCT [20], (c) PhotoWCT [21], and (d) our model are shown. Every result is produced *without* any post-processing. While WCT and PhotoWCT suffer from spatial distortions, our model successfully transfers the style and preserves the fine details.

Abstract

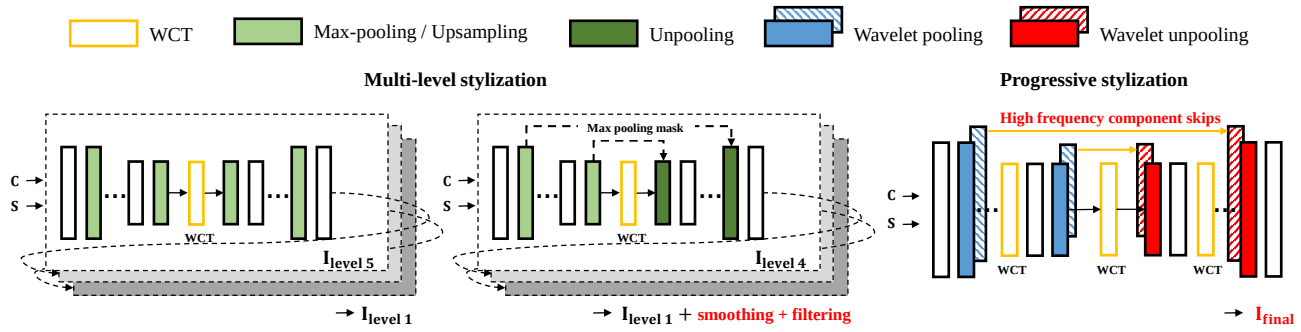
Recent style transfer models have provided promising artistic results. However, given a photograph as a reference style, existing methods are limited by spatial distortions or unrealistic artifacts, which should not happen in real photographs. We introduce a theoretically sound correction to the network architecture that remarkably enhances photorealism and faithfully transfers the style. The key ingredient of our method is wavelet transforms that naturally fits in deep networks. We propose a wavelet corrected transfer based on whitening and coloring transforms (WCT^2) that allows features to preserve their structural information and statistical properties of VGG feature space during stylization. ***This is the first and the only end-to-end model that can stylize a 1024×1024 resolution image in 4.7 seconds, giving a pleasing and photorealistic quality without any post-processing****. Last but not least, our model provides a stable video stylization without temporal constraints. Our code, generated images, and pre-trained models are all available at ClovaAI/WCT2.*

1. Introduction

Photorealistic style transfer has to satisfy contradictory objectives. To be photorealistic, a model should apply the reference style on the scene without hurting the details of an image. In Figure 1, for example, the general style (color and tone) of sky and sea should change, while the fine structures of the ship and the bridge remain intact. However, artistic style transfer methods (*e.g.*, whitening and coloring transforms, WCT [20]) generally suffer from severe distortions due to their strong abstraction ability, which is not favored in the photorealistic stylization (Figure 1b). (Please refer to our supplementary materials for more failure cases.)

Luan *et al*. [24] introduced a regularizer for photorealism on the traditional optimization-based method [9]. However, solving the optimization problem requires heavy computational costs, which limits their use in practice. To overcome this issue, Li *et al*. [21] recently proposed a photorealistic variant of WCT (PhotoWCT) that replaced the upsampling components of the VGG decoder with unpooling. By

* indicates equal contribution. Corresponding author: jaejun.yoo88@gmail.com



(a) WCT (artistic) (b) PhotoWCT (c) Ours (WCT^2)

Figure 2: Comparison between previous style transfer models and our proposed model architecture (WCT^2). Unlike WCT [20] and PhotoWCT [21] that use max-pooling and recursively stylize from coarse (level 5) to fine (level 1), WCT^2 replaces lossy operations (green) with wavelet pooling (blue) and unpooling (red), and employs the progressive stylization strategy in a single pass. Note that given the content (**c**) and style (**s**), WCT^2 outputs the final image (I_{final}) while the PhotoWCT output (I_{level1}) needs further post-processing steps (smoothing and filtering).

providing a max-pooling mask, PhotoWCT is designed to compensate for information loss during the encoding step and suppress the spatial distortion. Although their approach was valid, the introduction of the mask was not able to resolve the information loss that comes from the max-pooling of VGG network (Figure 1c). To fix the remaining artifacts, they had to perform a series of post-processing steps, which require the original image to patch up the result. Not only do these post-processing steps retuire cumbersome computation and time but they entail another unfavorable blurry artifact and hyper-parameters to manually set.

Instead of providing partial amendments, we address the fundamental problem by introducing a theoretically sound correction on the downsampling and upsampling operations. We propose a *wavelet corrected transfer* based on whitening and coloring transforms (WCT^2) that substitutes the pooling and unpooling operations in the VGG encoder and decoder with wavelet pooling and unpooling. Our motivation is that the learned function by the network should have its inverse operation to enable exact signal recovery, and accordingly, photorealistic stylization. (We provide theoretical details in our supplementary materials.) It allows WCT^2 to fully reconstruct the signal without any post-processing steps, thanks to the favorable properties of wavelets providing minimal information loss [34, 35]. The decomposed wavelet features provide interesting interpretations on the feature space as well, such as *component-wise stylization* and *why average pooling is known to give better stylization than max-pooling* (Section 4.1).

In addition, we propose *progressive stylization* instead of following the multi-level strategy that is used in WCT [20] and PhotoWCT [21] (Figure 2). To maximize the stylization effect, WCT and PhotoWCT recursively transformed features in a multi-level manner from coarse to fine. In contrast, we progressively transform features during a single pass. This allows two significant advantages over the others. First, our model is simple and efficient since we only have a single decoder during training as well as in the inference time. On the other hand, the multi-level strategy requires to train a decoder for each level without sharing parameters, which is inefficient in terms of the number of parameters and training procedure. This overhead remains in the inference time as well because the model requires to pass multiple encoder and decoder pairs to stylize an image. Second, by recursively encoding and decoding the signal with the lossy VGG networks, artifacts are amplified during the multi-level stylization. Because of wavelet operations and progressive stylization, our model does not have such a problem, and even more, it shows little error amplification when the multi-level strategy is employed (Figure 6).

Our contributions are summarized as follows. We first show that the spatial distortions come from the network operations that cannot satisfy the reconstruction condition (Section 3). By employing the wavelet corrected transfer and progressive stylization, we propose **the first end-to-end photorealistic style transfer model** that allows to remove the additional post-processing steps. Our model can **process a high resolution image (1024×1024) in 4.7 seconds**, which is **830 times faster** than the state-of-the-art models, where PhotoWCT fails due to an out-of-memory issue and Deep Photo Style Transfer (DPST) [24] takes 3887.8 seconds. Our experimental results show quantitatively **better visual quality** in both SSIM and Gram loss (Figure 9), and qualitatively **being preferred by 62.21%** in the user study (Table 2). In addition, our model has **three times fewer** parameters than PhotoWCT and provides **temporally stable stylization** enabling video applications without additional constraints, such as optical flow (Figure 8).

2. Related Work

Style transfer. Starting from the seminal work of Gatys *et al*. [9], many artistic style transfer studies have been proposed to synthesize stylized images through either iterative optimization [10], finding dense correspondence [22, 30, 11] or manipulating features in pre-trained networks [14, 20]. However, due to their powerful ability to abstract the features, they cannot be used in the photorealistic style transfer as they are.

Compared to artistic style transfer, photorealistic transfer has been overlooked. Classical methods mostly match the color and tone [1, 26, 27] of the images, which are restricted to specific usage. Luan *et al*. [24] proposed deep photo style transfer (DPST), which augments the neural style algorithm [9] with an additional photorealism regularization term and a semantic segmentation mask. However, DPST requires heavy computation to solve the regularized optimization problem.

Recently, Li *et al*. [21] proposed a photorealistic variant of WCT (PhotoWCT), which replaces the upsampling of the VGG decoder with unpooling. PhotoWCT showed that the spatial distortion could be relaxed by providing max-pooling masks to the decoder. Because the visual quality of the raw outputs of PhotoWCT was not satisfactory, the authors had to employ additional post-processing, such as *smoothing* and *filtering*. However, not only do these increase runtime exponentially to the image resolution, but blur final outputs.

Different from the existing methods, our method can preserve the fine structures of an image with little spatial distortion in an end-to-end manner, and thus removes the necessity of additional post-processing steps.

Signal reconstruction using wavelets. Signal reconstruction using wavelets has been an extensive research topic in applied mathematics community due to its favorable characteristics such as proven convergence and compact representation of an arbitrary signal [6, 19]. There have been several attempts to incorporate both classical signal processing and deep learning approaches, including feature reduction [18], network compression [12, 18], super-resolution [2], classification [3, 8, 25, 29, 32] and image denoising [15]. Similarly, our approach augments wavelets as a component of the network architecture and provides an interpretable module that can enhance the photorealism of a style transfer model.

One closest related work [32] recently proposed to use wavelets as an alternative to traditional neighborhood pooling. However, their goal is to reduce feature dimensions by discarding the first-level sub-bands, while we exploits all sub-bands. In addition, we utilize both wavelet decomposition and reconstruction together to exactly recover the spatial information with minimal noise amplification.

3. WCT^2

To achieve photorealism, a model should recover the structural information of a given content image while it stylizes the image faithfully at the same time. To address this issue, we propose a Wavelet Corrected Transfer based on Whitening and Coloring Transforms, dubbed WCT^2. More specifically, we handle the first objective by employing wavelet pooling and unpooling, which preserve information of the content to the transfer network. We use progressive stylization within a single forward pass to tackle the second issue.

3.1. Wavelet corrected transfer

Haar wavelet pooling and unpooling. We first explain the main components of our model using Haar wavelets, which we call wavelet pooling and unpooling. Haar wavelet pooling has four kernels, $\{\mathrm{LL}^\top\ \mathrm{LH}^\top\ \mathrm{HL}^\top\ \mathrm{HH}^\top\}$, where the low (L) and high (H) pass filters are

$$L^\top = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \end{bmatrix}, \quad H^\top = \frac{1}{\sqrt{2}}\begin{bmatrix} -1 & 1 \end{bmatrix}. \tag{1}$$

Thus, unlike common pooling operations, the output of the Haar wavelet pooling has four channels. Here, the low-pass filter captures smooth surface and texture while the high-pass filters extract vertical, horizontal, and diagonal edge-like information. For simplicity, we denote the output of each kernel as LL, LH, HL, and HH, respectively.

One important property of our wavelet pooling is that the original signal can be exactly reconstructed by mirroring its operation; *i.e*., wavelet unpooling. In detail, wavelet unpooling fully recovers the original signal by performing a component-wise transposed-convolution, followed by a summation. (Please see our supplementary materials for more details.) Thanks to this favorable property, our proposed model can stylize an image with minimal information loss and noise amplification. On the other hand, max-pooling does not have its exact inverse so that the encoder-decoder structured networks used in the WCT [20] and PhotoWCT [21] cannot fully restore the signal.

Note that Haar wavelet pooling and unpooling is not the only operation which can fully recover the original signal. However, we choose Haar wavelet because it splits the original signal into channels that capture different components, which leads to better stylization.

Model architecture. To fully utilize the encoded information, we replace every max-pooling and unpooling of PhotoWCT with the wavelet pooling and unpooling (Figure 2). Specifically, we use the ImageNet [5] pre-trained VGG-19 network [31] from `conv1_1` layer to `conv4_1` layer as the encoder. The max-pooling layers are replaced with wavelet pooling where the high frequency components

(LH, HL, HH) are skipped to the decoder directly. Thus, only the low frequency component (LL) is passed to the next encoding layer. The decoder has a mirror structure of the encoder, and the wavelet unpooling aggregates the components. (Please refer to our supplementary materials for more details about the proposed network architecture)

3.2. Stylization

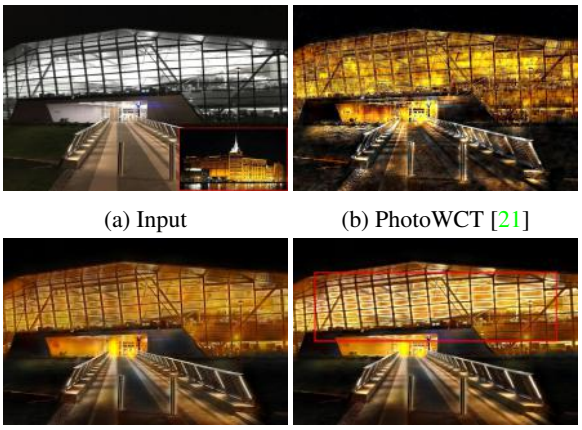
Whitening and coloring transforms (WCT). Since our method is built upon WCT [20][1], we first recap WCT briefly. WCT can perform style transfer with arbitrary styles by directly matching the correlation between content and style in the VGG feature domain. It projects the content features to the eigenspace of style features by calculating singular value decomposition (SVD). The final stylized image is obtained by feeding the transferred features into the decoder. To provide better artistic style transfer, the authors employed a multi-level stylization framework by applying WCT to multiple encoder-decoder pairs (Figure 2b).

Progressive stylization. Instead of using the multi-level stylization used in WCT and PhotoWCT, we progressively transform features within a single forward-pass as illustrated in Figure 2. We sequentially apply WCT at each scale (`conv1_X`, `conv2_X`, `conv3_X` and `conv4_X`) within a single encoder-decoder network. Note that the number of SVD computations of our model remains the same. We can add more WCTs on skip-connections and decoding layers to further strengthen the stylizing effect at the cost of time consumption. This will be covered in more detail in Section 4.4. There are several advantages of our proposed progressive stylization against the multi-level one. First, the multi-level strategy requires to train a decoder for each level without sharing parameters, which is inefficient. On the other hand, our training procedure is simple because we only have a single pair of encoder and decoder, which is advantageous in the inference time as well. Second, recursively encoding and decoding the signal with VGG network architecture amplifies errors causing unrealistic artifacts in the output. In the later section, we show that our proposed progressive stylization technique suffers less from the error amplification than the multi-level strategy.

4. Analysis

4.1. Wavelet pooling

We first examine the effects of using the wavelet pooling instead of max-pooling. As shown in Figure 3b and 3c, PhotoWCT suffers from the loss of spatial information by max-pooling while ours preserves fine details. We recall that the low frequency component captures smooth surface and texture while the high frequency components detect edges. This enables our model to separately control the stylization effect by choosing a component. More specifically, it implies that applying WCT to LL of the encoder affects overall texture or surface while applying WCT to the high frequency components (*i.e.*, LH, HL, HH) stylize edges. Indeed, when we stylize all components (Figure 3c), our model transfers the given style to the entire building. In contrast, if we do not perform WCT on the high frequency components, the boundaries of windows remain unchanged (Figure 3d).



(a) Input (b) PhotoWCT [21]

(c) Ours (d) Ours (LL only)

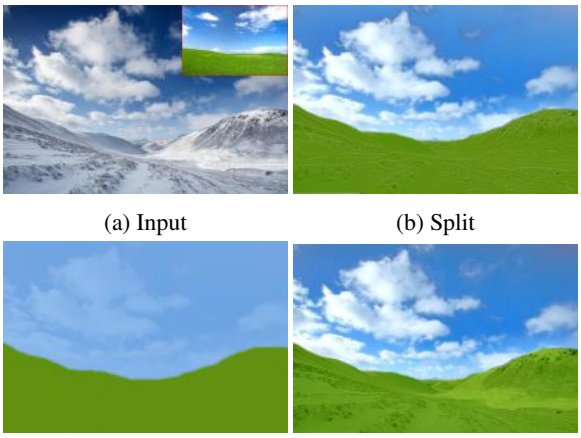
Figure 3: Comparison between max-pooling and wavelet pooling. Given (a) an input pair (inset: style), we compare the results of (b) PhotoWCT without post-processing, (c) ours and (d) ours but stylize only the LL component. Note that the edges are left unstylized (inside the red box).

Note that using only the LL component of our wavelet pooling is equivalent to using the average pooling. Interestingly, since Gatys *et al*. [9], many studies have consistently reported that replacing the max-pooling operation with average pooling yields slightly more appealing results. This can be explained in our framework that the model is using only the partial information (LL) of the wavelet decomposed feature domain. In addition, because each frequency component of the content feature is transformed into its corresponding component of style feature, we can obtain a similar advantage as we do by using spatial correspondences.

4.2. Ablation study

To show that our model indeed benefits from the wavelet pooling, we compare the stylization results using other pooling variants. We unpool the features similar to the way we do for the wavelet unpooling; *i.e.*, transposed-convolution and summation.

[1]Note that our wavelet corrected transfer is not limited to a specific stylization method. Here, we simply used WCT for better stylization. For example, at the expanse of slight image quality degradation, our model can be integrated with AdaIN [14], which further accelerates the model by avoiding SVD calculation. (Please refer to the supplemntary materials.)



(a) Input (b) Split

(c) Learnable (d) Ours (WCT^2)

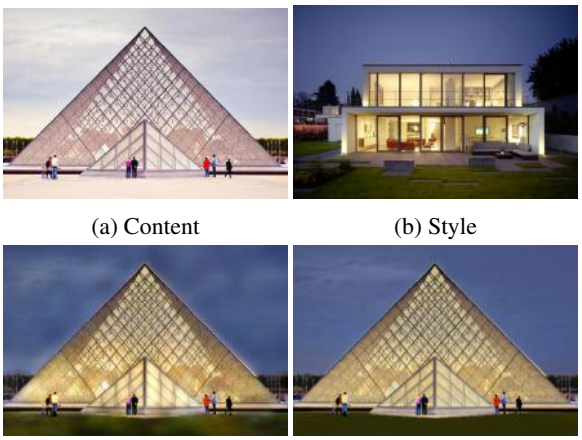
Figure 4: Ablation study on pooling methods. While split and learnable poolings suffer from the lack of representation power or altered feature statistics, wavelet pooling benefits from the compact representation of wavelets and retains the original VGG feature property intact.

Split pooling. Split pooling has 2×2 filters with fixed weights, *i.e.*, [1 0 0 0], [0 1 0 0], [0 0 1 0], and [0 0 0 1]. Split pooling has a similar property to wavelet pooling in that it can carry whole information. Here, we can see a similar effect but degradation in fine details, *e.g.*, the grass (Figure 4b). We suspect that this is due to the lack of representation power.

Learnable pooling. Learnable pooling is a trainable `conv` layer with a stride of two. As shown in Figure 4c, it does not preserve the content nor faithfully transfer the style. We suppose that this happens because the learnable pooling brings too much flexibility to the network. This ruins the original feature properties of VGG networks [31], which is known to be good at extracting styles [9].

4.3. Unpooling options

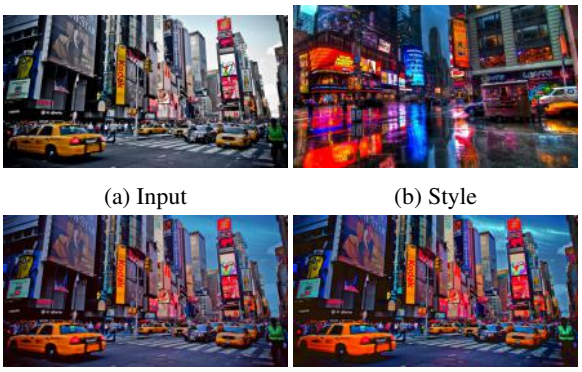
To achieve better reconstruction, we adopted concatenation instead of summation for unpooling, similar to U-Net structure [13, 28, 36]. This enables the network to learn the weighted sum of components at the expense of interpretability and theoretical correctness. Specifically, our wavelet unpooling now performs channel-wise concatenation of four feature components from the corresponding scale plus feature output before the wavelet pooling. Therefore, the number of parameters increases at the `conv` layer that comes right after the wavelet unpooling. This increases the total number of parameters to be $1.80\times$ of the sum-version of WCT^2 while PhotoWCT has $3.06\times$ parameters. As shown in Figure 5, spatial details are further improved. The sum-version generally produces a more stylized output



(a) Content (b) Style

(c) Sum-version (d) Concatenated-version

Figure 5: Variation of the unpooling options (Section 4.3).



(a) Input (b) Style

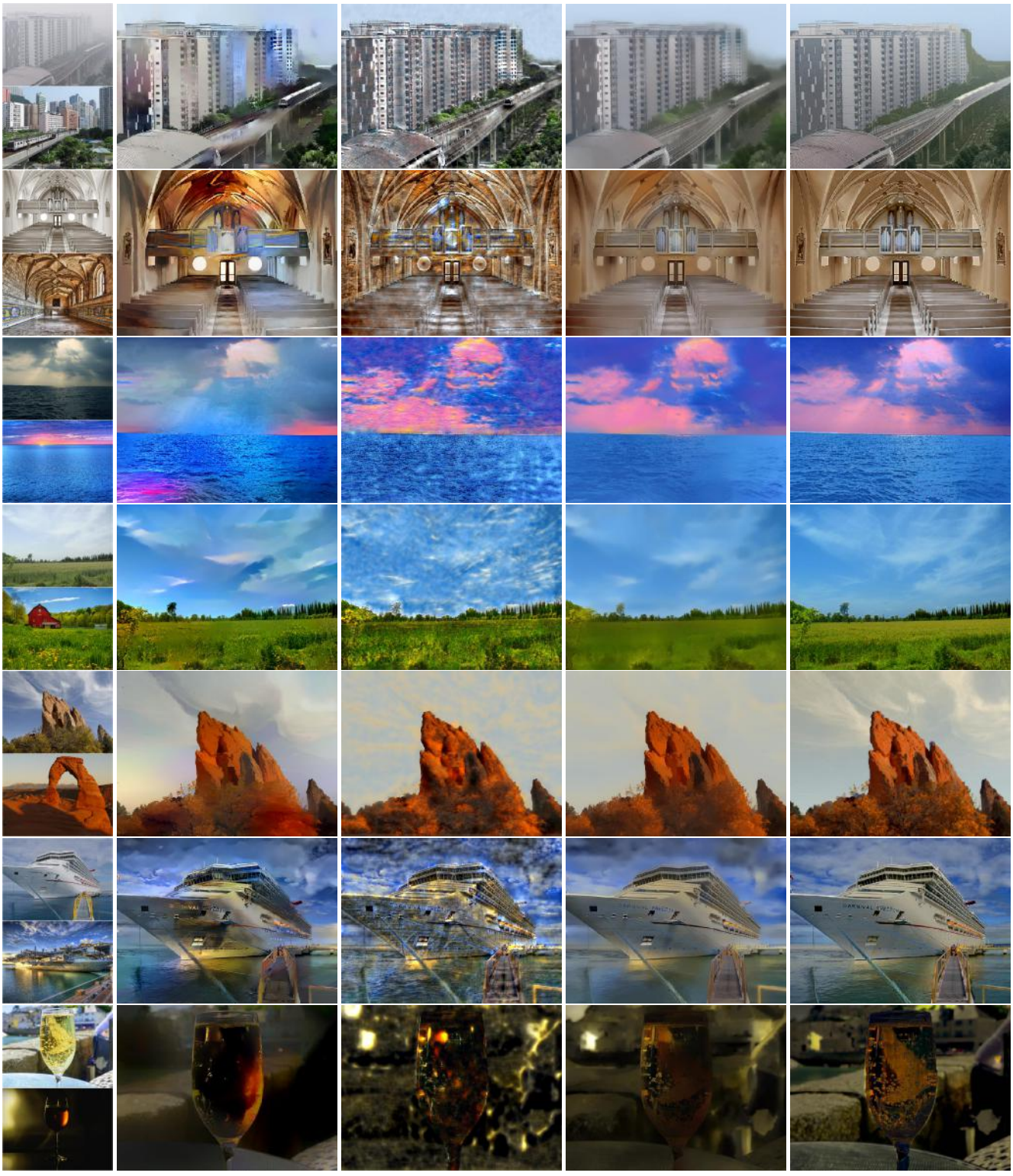
(c) Single-pass (WCT^2) (d) Multi-level

Figure 6: Stylization strength with more whitening and coloring transforms. Single-pass is our baseline (Section 4.4).

while the concatenated-version produces a clearer image. (Please see our supplementary materials for more results.)

4.4. Progressive vs. multi-level strategy

Owing to the exact reconstruction property of wavelet pooling, our model can adopt the multi-level strategy to increase the contrast in the transferred style with minimal noise amplification. As shown in Figure 6d, adopting the multi-level approach in addition to WCT^2 leads to more vivid results. Note that it maintains photorealism while PhotoWCT produces spotty artifacts due to the noise amplification (Figure 7c). In addition, performing progressive stylization at the decoder as well, namely `conv3_2`, `conv2_2`, and `conv1_2`, further increases stylization effect. Still, strengthening the style comes at the cost of photorealism and multiple SVD computations. (Please refer to the supplementary materials for more results)



(a) Input (b) DPST [24] (c) PhotoWCT [21] (d) PhotoWCT (full) [21] (e) Ours (WCT^2)

Figure 7: Photorealistic stylization results. Given (a) an input pair (top: content, bottom: style), the results of (b) deep photo style transfer (DPST) [24], (c) and (d) PhotoWCT [21], and (e) ours (WCT^2) are shown. PhotoWCT (full) denotes the results after applying two post-processing steps proposed by the authors [21]. Note that WCT^2 **does not** need any post-processing.

Figure 8: Photorealistic video stylization results (from day-to-sunset). Given a style image and video frames (top), we show the results by (middle) WCT^2 and (bottom) PhotoWCT [21] without providing semantic segmentation maps and post-processing steps.

5. Experimental results

In this section, we show that our simple modification can remarkably enhance the performance of photorealistic style transfer. Here, every result is reported based on the concatenated version of our model. For a fair comparison and time-efficiency, we only perform whitening and coloring on LL components (*e.g.*, `convX_1` outputs of the encoder) progressively. Thus, the number of whitening and coloring procedure of our model matches with PhotoWCT.

5.1. Implementation details

We use the encoder-decoder architecture with fixed VGG encoder weights. The decoder is trained on Microsoft COCO dataset [23], minimizing the L2 reconstruction loss and the additional feature Gram matching loss with the encoder. The training is done with NAVER Smart Machine Learning (NSML) platform [16]. We use ADAM optimizer [17] with a learning rate of 10^{-3}. Finally, similar to PhotoWCT and DPST, we utilize the semantic map to match the styles of corresponding image regions. The code and pre-trained models are available at ClovaAI/WCT2.

5.2. Qualitative evaluation

Figure 7 shows the results of DPST, PhotoWCT and ours (WCT^2). DPST often generates "staircasing" or "cartoon" artifacts [4] with an unrealistic color transfer, which severely hurts photorealism (Figure 7b). PhotoWCT better reconstructs the details of the content image, while it shows spotty artifacts over entire images (Figure 7c). Such artifacts can be removed by employing additional post-processing steps (Figure 7d). However, it has three disadvantages that 1) optimization is slow, 2) hyper-parameters should be carefully tuned to trade-off between smoothness and fine details, and 3) the final image becomes blurry at the expense of removed artifacts. In contrast, our proposed method shows fewer artifacts while faithfully transferring the reference styles (Figure 7e). Note that we do not apply any post-processing after the network output.

Video stylization. To emphasize consistent feature representation of the wavelet pooling and unpooling, we separately stylize every video frame to target style without any semantic segmentation. Figure 8 shows that WCT^2 performs stable video style transfer without any temporal consistency regularization such as optical flow. On the other hand, PhotoWCT generates spotty and varying artifacts over frames, which harms the photorealism. (The link to the full video can be found in our project page.)

5.3. Quantitative evaluation

Statistics. To measure photorealism, we employ two surrogate metrics for spatial recovery and stylization. We calculate the structural similarity (SSIM) index between edge responses [33] of original contents and stylized images. Following WCT [20], we report the covariance matrix difference (VGG style loss [9]) between the style image and the outputs of each model. Figure 9 shows SSIM (X-axis) against style loss (Y-axis). Our proposed model (WCT^2) remarkably outperforms other methods.

Note that WCT^2 and its variants are located at the top-right corner, superior to PhotoWCT (full) and DPST that perform post-processing. Here, DPST has strength on the Gram-based score because it directly optimizes the style loss. Still, it is far from being practical due to its heavy optimization procedure (Table 1). As expected, when we compare the results of our variants, the multi-level approach adds more style (smaller Gram-based loss) at the expense of noise amplification (larger SSIM index), which is even better than the direct optimization (DPST).

In addition, by comparing the gap before and after the post-processing steps (Figure 9, dashed lines), we can clearly see that **the final visual qualities of PhotoWCT majorly come from the powerful post-processing, especially the smoothing step, not the network itself**. The original WCT with smoothing already shows a comparable result to that of PhotoWCT. This demonstrates that the unpooling substitution of PhotoWCT did not fully address the information loss but the post-processing did.

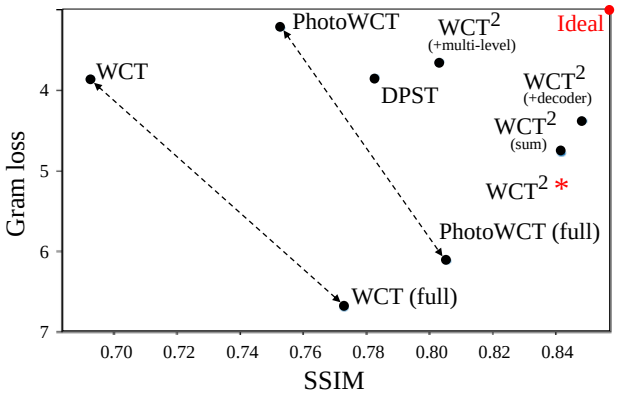


Figure 9: SSIM index (higher is better) versus Style loss (lower is better). Ideal case is the top-right corner (red dot). Dashed lines depict the gap before and after the post-processing steps, *i.e.*, smoothing. The baseline WCT^2 with concatenation is denoted by the red asterisk.

Image Size	DPST	PhotoWCT (full) (WCT + post)	Ours
256×256	306.9	$3.2 + 9.2$	**3.2**
512×512	1020.7	$3.6 + 40.2$	**3.8**
896×896	2988.6	$3.8 + \text{OOM}$	**4.4**
1024×1024	3887.8	$3.9 + \text{OOM}$	**4.7**

Table 1: Runtime comparision of DPST, PhotoWCT (full) and ours in seconds. OOM denotes out-of-memory error.

Runtime & memory. Table 1 shows the runtime comparison of DPST, PhotoWCT, and WCT^2. For PhotoWCT, we separately measured WCT and post-processing steps to better compare with ours. The reported runtime for each model is an average of ten-rounds run on a single NVIDIA P40 GPU. As expected, our model inherits the computational time of the original WCT. Note that the concatenation in unpooling hardly increases the runtime of WCT^2. Because our model can remove the cumbersome post-processing steps, WCT^2 can deal with high resolution images, such as 1024×1024, maintaining a high quality of photorealistic style transfer. Compared to DPST, WCT^2 achieves a speed-up of about **830 times** in runtime. In addition, WCT^2 uses only 51% GPU-memory of PhotoWCT, which uses a multi-level stylization requiring four encoder-decoder models (Section 4.4) since WCT^2 progressively stylize an image using a single encoder-decoder.

User study. We conducted a user study to further evaluate the methods in terms of fewer artifacts, faithfulness to the style input, and overall qualities. Our benchmark dataset consists of content and style pairs provided by Luan *et al.* [24]. Total 40 sets of questions were asked to 41 subjects, in which subjects had to choose one among three stylized images from each model. The results are shown in random order with content and style images. Table 2 shows the percentage of model outputs that are chosen out of 1640 ($= 40 \times 41$) responses. Our method is preferred by human subjects against the other state-of-the-art methods by a large margin in all aspects. Note that we compare our results with PhotoWCT (full) that applies two post-processing steps proposed by the authors [21] while we do not perform any post-processing for WCT^2. (Please see our supplementary materials for the images that are used for the user study)

	DPSP	PhotoWCT (full)	Ours
Fewest artifacts	21.34%	9.33%	**69.33%**
Best stylization	30.49%	12.74%	**56.77%**
Most preferred	24.63%	11.16%	**62.21%**

Table 2: User study results. The percentage indicates the preferred model outputs out of 1640 responses.

Failure cases. Many photorealistic models [24, 21] including ours require the semantic map and its accuracy is important for better stylization results. In fact, this phenomenon is more prominent in our model because WCT^2 retains every fine detail unlike the others (Supplementary materials). The effect of misaligned map is visible in our result while PhotoWCT smooths it out unintentionally. Resolving the dependency on the semantic label map is an interesting future research direction.

6. Conclusion

In this paper, we proposed the first end-to-end photorealistic style transfer method, WCT^2. Based on the theoretical analysis, we specifically designed our model to satisfy the reconstruction condition. The exact recovery of the wavelet transforms allows our model to preserve structural information while providing stable stylization without any constraints. By employing progressive stylization, we achieved better results with less noise amplification. Compared to the other state-of-the-arts, our analysis and experimental results showed that WCT^2 is scalable, lighter, faster and achieves better photorealism quantitatively and qualitatively. Our results were preferred by human subjects in every aspect with a significant margin. Future study will include removing the necessity of semantic labels, which should be accurate for a flawless result so far.

7. Acknowledgement

We would like to thank Clova AI Research team and advisory members, especially Yunjey Choi, Sangdoo Yun, Dongyoon Han, Youngjoon Yoo, and Jun-Yan Zhu for their helpful feedback and discussion.

References

[1] Soonmin Bae, Sylvain Paris, and Frédo Durand. Two-scale tone management for photographic look. *ACM Transactions on Graphics (TOG)*, 25(3):637–645, 2006. 3

[2] Woong Bae, Jaejun Yoo, and Jong Chul Ye. Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification. In *CVPR Workshops*, pages 1141–1149, 2017. 3

[3] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013. 3

[4] Tony F Chan and Jianhong Jackie Shen. *Image processing and analysis: variational, PDE, wavelet, and stochastic methods*, volume 94. Siam, 2005. 7, 11

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009. 3

[6] Bin Dong, Qingtang Jiang, and Zuowei Shen. Image restoration: wavelet frame shrinkage, nonlinear evolution PDEs, and beyond. *Multiscale Modeling & Simulation*, 15(1):606–660, 2017. 3

[7] Richard J Duffin and Albert C Schaeffer. A class of nonharmonic Fourier series. *Transactions of the American Mathematical Society*, 72(2):341–366, 1952. 11

[8] Shin Fujieda, Kohei Takayama, and Toshiya Hachisuka. Wavelet convolutional neural networks for texture classification. *arXiv preprint arXiv:1707.07394*, 2017. 3

[9] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423. IEEE, 2016. 1, 3, 4, 5, 7, 13, 14

[10] Leon A Gatys, Alexander S Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling perceptual factors in neural style transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[11] Shuyang Gu, Congliang Chen, Jing Liao, and Lu Yuan. Arbitrary style transfer with deep feature reshuffle. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8222–8231, 2018. 3

[12] Lionel Gueguen, Alex Sergeev, Rosanne Liu, and Jason Yosinski. Faster neural networks straight from jpeg. In *ICLR*, 2018. 3

[13] Yoseob Han and Jong Chul Ye. Framing u-net via deep convolutional framelets: Application to sparse-view ct. *IEEE transactions on medical imaging*, 37(6):1418–1429, 2018. 5

[14] Xun Huang and Serge J Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, pages 1510–1519, 2017. 3, 4, 13, 14

[15] Eunhee Kang, Won Chang, Jaejun Yoo, and Jong Chul Ye. Deep convolutional framelet denosing for low-dose ct via wavelet residual network. *IEEE transactions on medical imaging*, 37(6):1358–1369, 2018. 3

[16] Hanjoo Kim, Minkyu Kim, Dongjoo Seo, Jinwoong Kim, Heungseok Park, Soeun Park, Hyunwoo Jo, KyungHyun Kim, Youngil Yang, Youngkwan Kim, et al. NSML: Meet the mlaas platform with a real-world case study. *arXiv preprint arXiv:1810.09957*, 2018. 7

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[18] A Levinskis. Convolutional neural network feature reduction using wavelet transform. *Elektronika ir Elektrotechnika*, 19(3):61–64, 2013. 3

[19] Bing Li and Xuefeng Chen. Wavelet-based numerical analysis: a review and classification. *Finite Elements in Analysis and Design*, 81:14–31, 2014. 3

[20] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *Advances in Neural Information Processing Systems*, pages 386–396, 2017. 1, 2, 3, 4, 7, 12, 13, 14

[21] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. *arXiv preprint arXiv:1802.06474*, 2018. 1, 2, 3, 4, 6, 7, 8, 12, 15, 16, 17

[22] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. Visual attribute transfer through deep image analogy. *ACM Transactions on Graphics (TOG)*, 36(4):120, 2017. 3

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 7

[24] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. 1, 2, 3, 6, 8, 13, 15, 16

[25] Edouard Oyallon, Eugene Belilovsky, and Sergey Zagoruyko. Scaling the scattering transform: Deep hybrid networks. In *International Conference on Computer Vision (ICCV)*, 2017. 3

[26] Francois Pitie, Anil C Kokaram, and Rozenn Dahyot. N-dimensional probablility density function transfer and its application to colour transfer. In *null*, pages 1434–1439. IEEE, 2005. 3

[27] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001. 3

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 5, 12

[29] Jongbin Ryu, Ming-Hsuan Yang, and Jongwoo Lim. Dft-based transformation invariant pooling layer for visual classification. In *The European Conference on Computer Vision (ECCV)*, September 2018. 3

[30] Lu Sheng, Ziyi Lin, Jing Shao, and Xiaogang Wang. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3

[31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 3, 5

[32] Travis Williams and Robert Li. Wavelet pooling for convolutional neural networks. In *ICLR*, 2018. 3

[33] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015. 7

[34] Jong Chul Ye, Yoseob Han, and Eunju Cha. Deep convolutional framelets: A general deep learning framework for inverse problems. *SIAM Journal on Imaging Sciences*, 11(2):991–1048, 2018. 2, 11, 12

[35] Rujie Yin, Tingran Gao, Yue M Lu, and Ingrid Daubechies. A tale of two bases: Local-nonlocal regularization on image patches with convolution framelets. *SIAM Journal on Imaging Sciences*, 10(2):711–750, 2017. 2, 11, 12

[36] Jaejun Yoo, Abdul Wahab, and Jong Chul Ye. A mathematical framework for deep learning in elastic source imaging. *SIAM Journal on Applied Mathematics*, 78(5):2791—-2818, 2018. 5

A. Frame-based signal reconstruction

Our proposed model WCT^2 is inspired by the recent theoretical advancement of frame-based signal reconstruction approaches [34, 35]. To make the paper self-contained, we provide a brief introduction to the frame theory (Section A.1), tightness of Haar wavelets (Section A.2) and our theoretical motivation (Section A.3).

A.1. Perfect reconstruction condition

Consider an *analysis operator* $\Phi = \begin{bmatrix}\phi_1 & \cdots & \phi_m\end{bmatrix} \in \mathbb{R}^{n\times m}$, where $\{\phi_k\}_{k=1}^m$ is a family of functions in a Hilbert space H. Then, $\{\phi_k\}_{k=1}^m$ is called a *frame* if it satisfies the following inequality [7]:

$$\alpha\|f\|^2 \leq \|\Phi^\top f\|^2 \leq \beta\|f\|^2, \quad \forall f \in H, \tag{2}$$

where $f \in \mathbb{R}^n$ is an input signal and $\alpha, \beta > 0$ are called the frame bounds.

The original signal f can be exactly recovered from the frame coefficient $z = \Phi f$ when there is the *dual frame* $\tilde{\Phi}$ (*i.e.*, *synthesis operator*) satisfying *the perfect reconstruction (PR) condition*: $\tilde{\Phi}\Phi^\top = I$, since $f = \tilde{\Phi}z = \tilde{\Phi}\Phi^\top f = f$. Here, we call such frame *tight* (*i.e.*, $\alpha = \beta$ in (2)) which is equivalent to $\tilde{\Phi} = \Phi$ or $\Phi\Phi^\top = I$. Note that a tight frame does not amplify the power of the input and thus it has the minimum noise amplification factor. To achieve the best reconstruction performance, frame bases should satisfy another property, called energy compaction. This is particularly important to parametric models, which have to adaptively deal with varying amounts of information with a fixed number of parameters, *e.g.*, deep neural networks (DNNs). For example, singular value decomposition (SVD) provides both tight and energy compact bases given an arbitrary signal. However, SVD is data-dependent, which makes it hard to use for a large dataset.

A.2. Wavelet frames

Wavelets are known to compactly represent signals while maintaining important information such as edges, thus resulting in a good energy compaction [4]. Therefore, by using a tight wavelet filter-bank, we can improve the reconstruction performance of encoder-decoder type of networks with minimal noise amplification. Specifically, the non-local basis Φ^T is now composed of a filter bank:

$$\Phi = [T_1 \cdots T_L], \tag{3}$$

where T_k denotes the k-th subband operator and the filter bank is tight, *i.e.*

$$\Phi\Phi^T = \sum_{k=1}^{L} T_k T_k^T = I. \tag{4}$$

In this paper, we use Haar wavelets which is one of the simplest tight filter bank frames with low and high sub-band decomposition. Here, $T_1 \in \mathbb{R}^{\frac{n}{2}\times n}$ is the low-pass subband. This is equivalent to the average pooling:

$$T_1^\top = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 1 & \cdots & 0 & 0 \\ & \vdots & & & & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 1 \end{bmatrix}. \tag{5}$$

Then, T_2 is the high pass filtering given by

$$T_2^\top = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & -1 & \cdots & 0 & 0 \\ & \vdots & & & & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & -1 \end{bmatrix} \tag{6}$$

and we can easily see that

$$T_1 T_1^T + T_2 T_2^T = I, \tag{7}$$

so the Haar wavelet frame is tight.

A.3. Theoretical motivation

In the perspective of the frame-based signal reconstruction, the commonly used encoder-decoder convolution structure of deep neural networks (DNNs), such as U-net [28], can be interpreted as the data-driven way of learning the local bases Ψ (*e.g.*, convolution filters) with hand-crafted global bases Φ (*e.g.*, max-pooling) [34]. Recently, Ye *et al*. [34] interpreted training DNNs as finding a multi-layer realization of the convolution framelets [35]:

$$Z = \Phi^T(f \circledast \Psi) \tag{8}$$

$$f = \left(\tilde{\Phi} Z\right) \circledast \tilde{\Psi}, \tag{9}$$

where $\Phi = [\phi_1, \cdots, \phi_n]$ and $\tilde{\Phi} = [\tilde{\phi}_1, \cdots, \tilde{\phi}_n] \in \mathbb{R}^{n \times n}$ (resp. $\Psi = [\psi_1, \cdots, \psi_q]$ and $\tilde{\Psi} = [\tilde{\psi}_1, \cdots, \tilde{\psi}_q] \in \mathbb{R}^{d \times q}$) are frames and their duals. Here, $\circledast$ stands for the convolution operation.

Therefore, the convolutional layers of the encoder learns the signal representation with a global pooling operation. We refer to Φ as global bases because it observes the entire image dimension n while Ψ learns local features from the data by $d \times d$ convolution kernels of q channels. When these frames satisfy the PR condition:

$$\tilde{\Phi}\Phi^\top = I_{n \times n}, \quad \Psi\tilde{\Psi}^\top = I_{d \times d}, \tag{10}$$

the input signal f can be exactly recovered from the learned representations. Note that the encoder-decoder architectures of WCT [20] and PhotoWCT [21] cannot satisfy the perfect reconstruction condition because of the max-pooling, which does not have its exact inverse (*i.e.*, not a frame). On the other hand, our model WCT^2 can fully exploit the information from the encoder due to the favorable property of the wavelet decomposition and reconstruction, *i.e.*, Haar wavelet pooling and unpooling.

A.4. Proposed network architecture

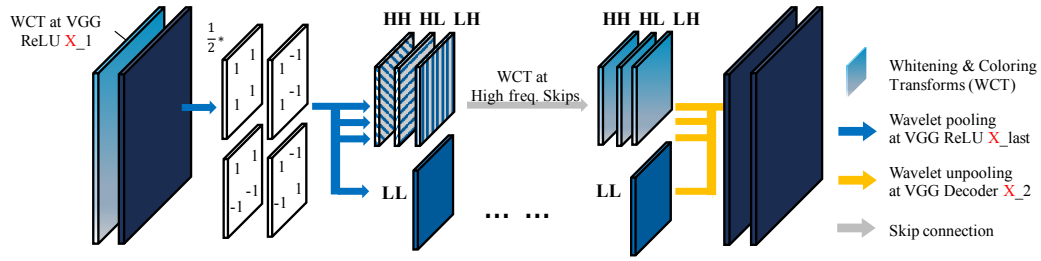


Figure 10: The proposed module using Haar wavelet pooling and unpooling.

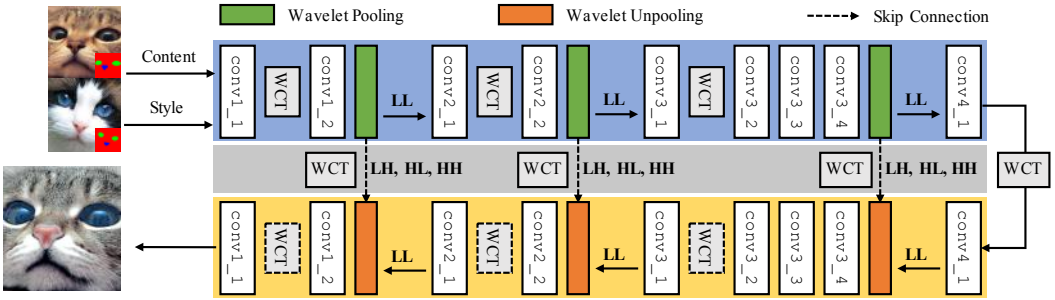
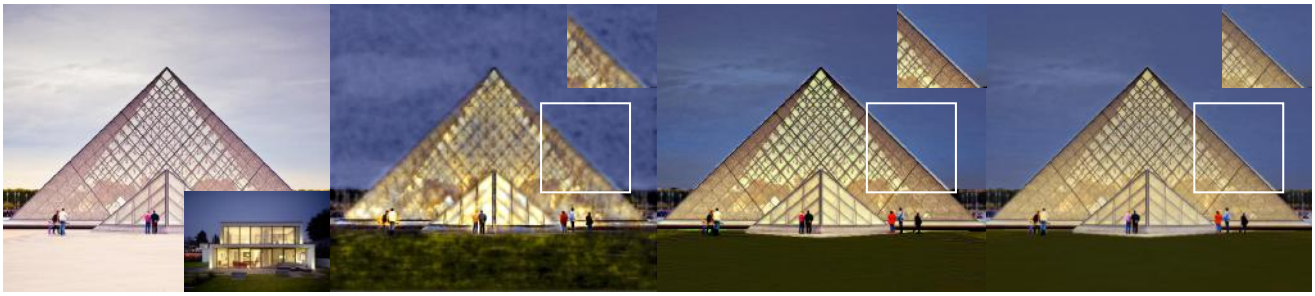


Figure 11: Overview of the proposed progressive stylization. For the encoder, we perform WCT on the output of `convX_1` layer and skip connections. For the decoder, we apply WCT on the output of `convX_2` layer, which is optional.

In Figure 10, a pair of encoder and decoder at same scale are shown. WCT is performed on the output of VGG `convX_1` layer followed by subsequent VGG layers and wavelet pooling. Only the low component passes to the next layer and the high frequency components are directly skipped to the corresponding decoding layer. At the decoder, the components are aggregated by the wavelet unpooling.

A.5. Differences to PhotoWCT and Wavelet corrected transfer based on AdaIN (WCT-AdaIN)

Our method shares the motivation with PhotoWCT but the way we posit the problem and reach to its solution is fundamentally different from PhotoWCT: *i*) We showed that the reason why PhotoWCT fails in preserving spatial information is because **pooling and unpooling operations cannot satisfy the frame condition** *ii*) Based on this theoretical analysis, our model architecture is **specifically designed to perfectly preserve spatial structure**, which is **proved effective in theory and practice.** This removes the necessity of post-processing, thus making our model far more practical and powerful than the previous methods. *iii*) The wavelet corrected model is **by no means limited to a specific stylization method**. It can serve as a **general architecture for photorealistic style transfer**, which is compatible with various methods, *e.g.*, AdaIN (Figure 12 (c)). Currently, our method (WCT^2) can process 1k resolution image in 4.7 seconds and this can be accelerated further ($\sim$1 second) by employing adaptive instance normalization (AdaIN) instead of time-consuming SVD procedure.



(a) Inputs **(b) PhotoWCT** **(c) Ours (AdaIN)** **(d) Ours (WCT)**

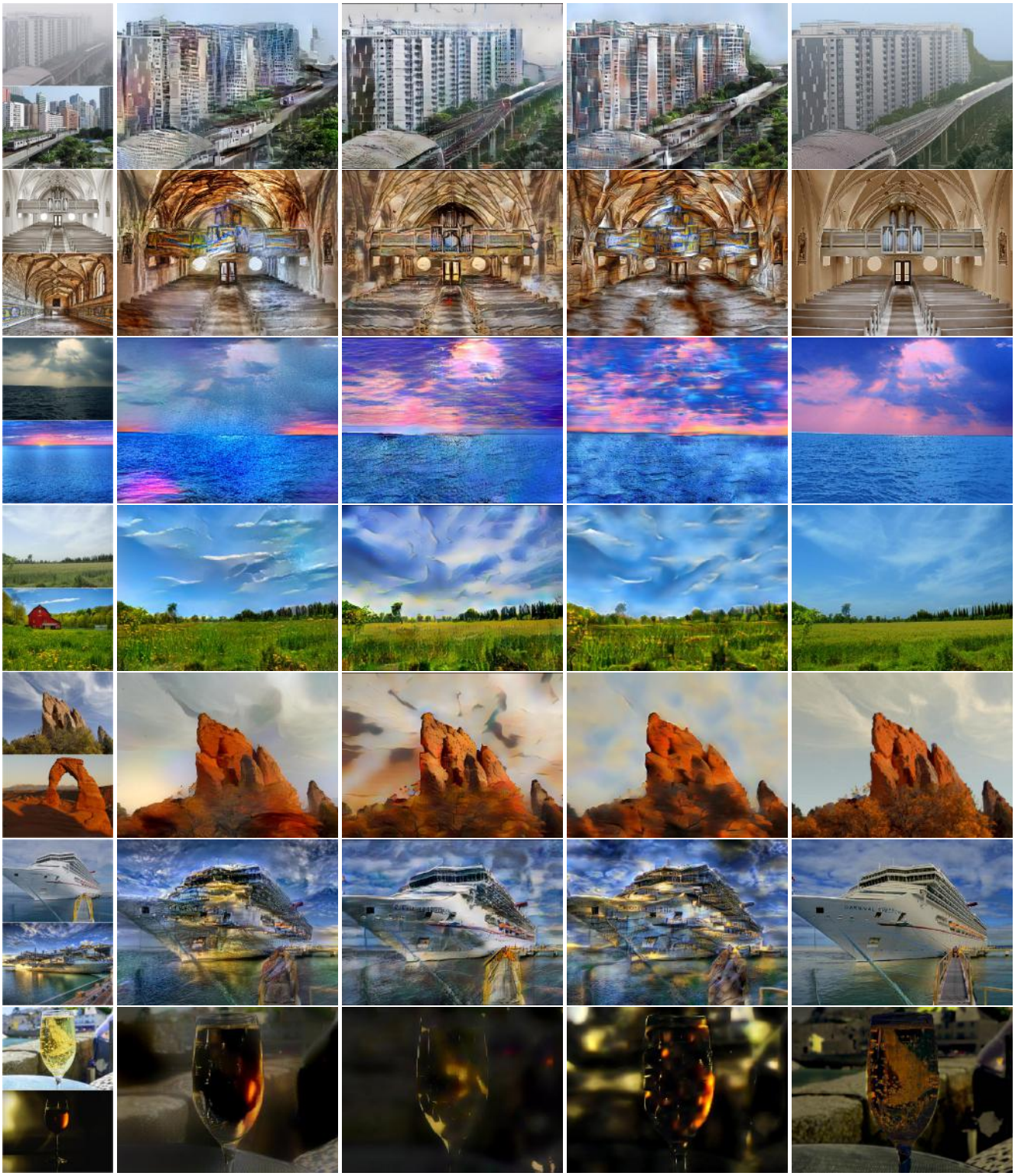
Figure 12: Photorealistic style transfer results of (a) input pairs using (b) PhotoWCT, (c) Ours (AdaIN) and (d) Ours (WCT). (c) is the results using our model architecture combined with AdaIN as the stylization method, and (d) is WCT^2 (proposed).

A.6. Qualitative comparison with artistic style transfer results

We compare our proposed WCT^2 with popular artistic style transfer methods including NeuralStyle [9], AdaIN [14] and WCT [20] in Figure 13. To apply semantic segmentation map to the artistic style transfer methods, we followed the spatial control techniques proposed by the authors [24, 14, 20] respectively. In the figure, artistic style transfer methods generate undesired distortions and artifacts and often fail to maintain the structural information despite the spatial control with segmentation maps. In comparison, because of the proposed wavelet corrected transfer, our proposed WCT^2 prevents unrealistic artifacts and preserve the structure information such as edges.

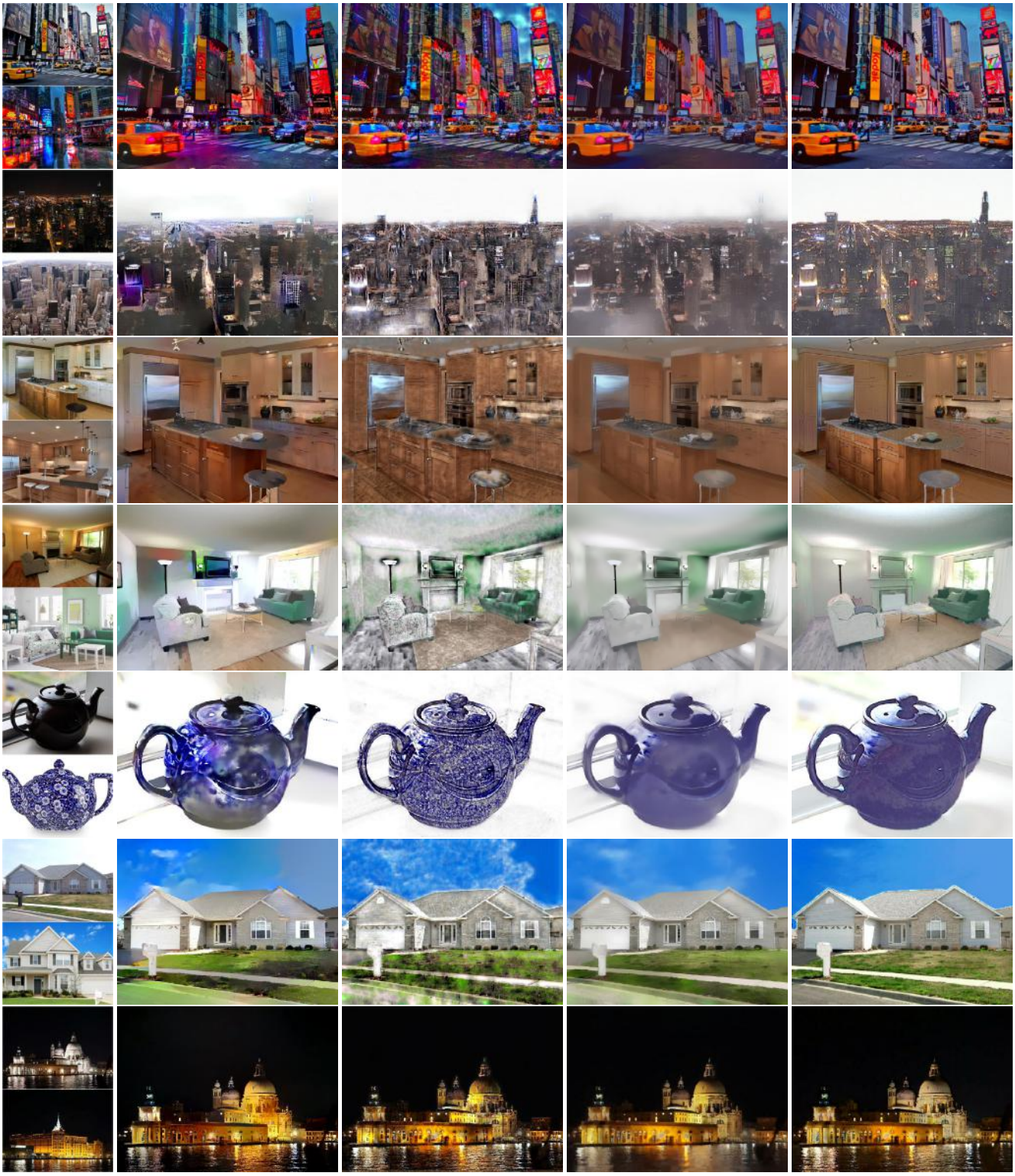
A.7. Additional Qualitative comparison with photorealistic style transfer

Additional qualitative results using WCT^2 and its variants are shown in Figure 14, Figure 15 and Figure 16. The video stylization results can be found in one of the other supplementary materials.



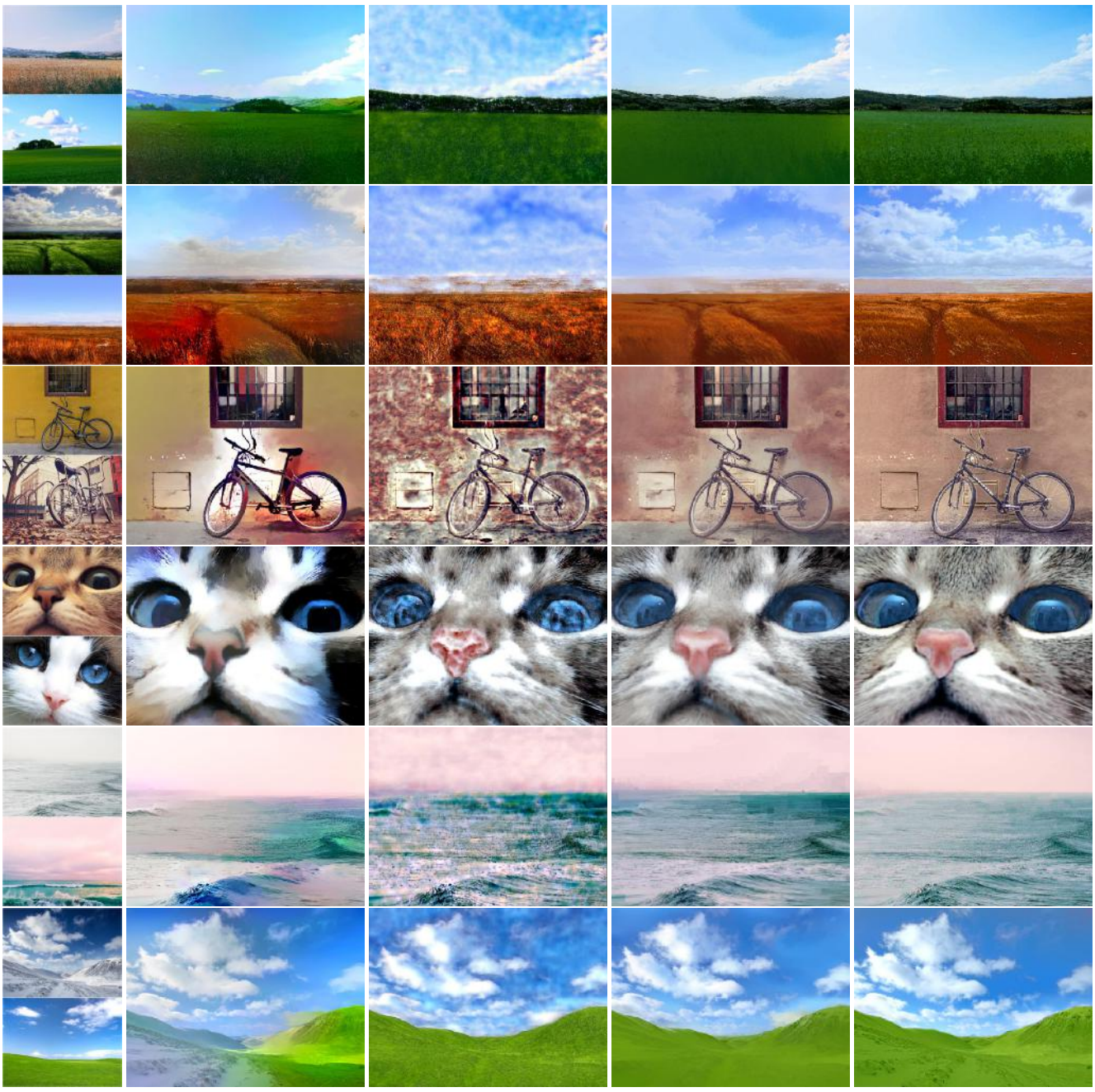
(a) Input (b) NeuralStyle [9] (c) AdaIN [14] (d) WCT [20] (e) Ours (WCT^2)

Figure 13: Qualitative comparison with artistic style transfer results. Given (a) an input pair (top: content, bottom: style), we compare the results of (b) NeuralStyle [9], (c) AdaIN [14] (d) WCT [20] and (e) ours (WCT^2).



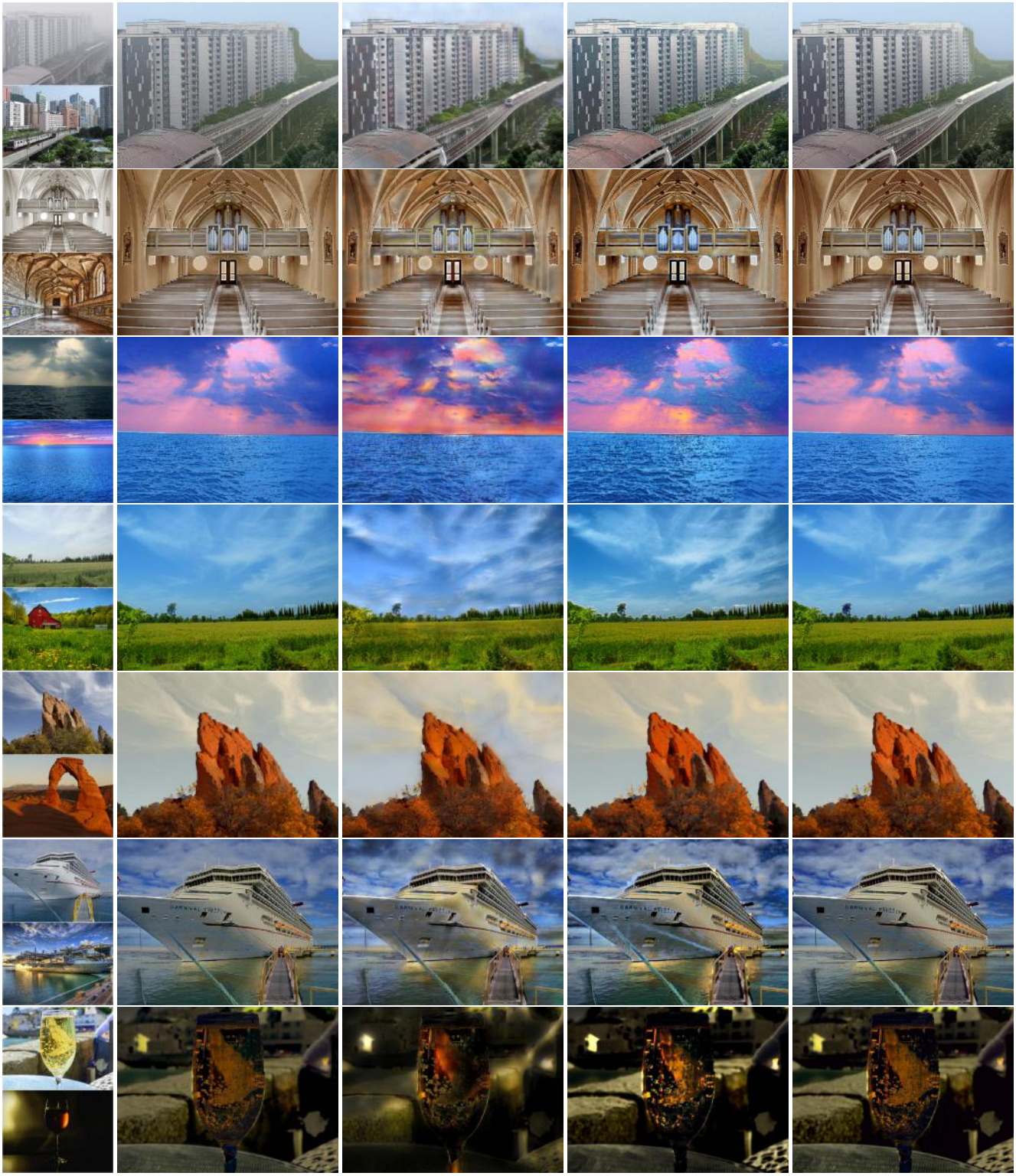
(a) Input (b) DPST [24] (c) PhotoWCT [21] (d) PhotoWCT (full) [21] (e) Ours (WCT^2)

Figure 14: Photorealistic style transfer results. Given (a) an input pair (top: content, bottom: style), we compare the results of (b) deep photo style transfer (DPST) [24], (c) and (d) PhotoWCT [21] and (e) ours (WCT^2). (c) is the results of PhotoWCT without any post-processing and (d) shows the results after applying two post-processing steps proposed by the authors [21].



(a) Input (b) DPST [24] (c) PhotoWCT [21] (d) PhotoWCT (full) [21] (e) Ours (WCT^2)

Figure 15: Photorealistic style transfer results. Given (a) an input pair (top: content, bottom: style), we compare the results of (b) deep photo style transfer (DPST) [24], (c) and (d) PhotoWCT [21] and (e) ours (WCT^2). (c) is the results of PhotoWCT without any post-processing and (d) shows the results after applying two post-processing steps proposed by the authors [21].



(a) Input (b) WCT^2 (c) WCT^2 (sum) [21] (d) WCT^2 (+multi-level) [21] (e) WCT^2 (+decoder)

Figure 16: Photorealistic style transfer results. Given (a) an input pair (top: content, bottom: style), we compare the results of WCT^2 and its variants, *i.e.*, (b) WCT^2, (c) WCT^2 (sum) (d) WCT^2 (+multi-level) and (e) WCT^2 (+decoder).