Time-of-Day Neural Style Transfer for Architectural Photographs – Supplementary Material –

Yingshu Chen, Tuan-Anh Vu, Ka-Chun Shum, Binh-Son Hua, and Sai-Kit Yeung

Abstract—In this supplemental document, we first provide more details about the dataset and our image translation network implementation (Section 1 and Section 2). Then, we provide perceptual user study details (Section 3). Finally, we show more results (Section 4) including the complete quantitative results, and more visual results and comparisons in terms of style transfer and blending. Particularly, we display all baseline comparisons in an interactive html viewer.

1 Dataset Details

Annotation Explanation. Here depicts the definition of the four labels of magic times of the day for the dataset (i.e., daytime, golden hours, blue hours and night time), following the same annotation settings as Shih *et al.*'s work [1].

Officially speaking, the golden hours occur just after sunrise over the horizon or before sunset when the sun falls closing to the horizon, creating the magical warm glow. The blue hours appear shortly before sunrise and after sunset with the sun's position just below the horizon and produce the cooler tones. In addition to these four time slots, there are some hours worth photographing, such as civil, nautical, and astronomical hours [2], which arrive between the blue hour and nighttime. To covers all enchanting hours in a day for style transfer, we simply merge blue, civil, nautical, and hours as blue moments, astronomical and nighttime as night hours.

Dataset Statistics and Visual Examples. Table 1 illustrates the detailed statistics of our time-lapse architectural dataset. The evaluation set is a separate unseen set for the training set, consisting of high-fidelity real-world photos from public domains [3], [4], [5].

Some data and segmentation examples are shown in Figure 1 and 2. From day to night, illumination sources keep changing. In the daytime, the illumination of buildings comes from the ambient environment, mainly the sky. When the sun is falling and the sky is becoming dark, buildings derive lighting sources from interior and exterior lights such as buildings and street lights. Daytime images can always provide sharp and detailed geometry of the scenes. In contrast, images in golden, blue, and night hours depict pleasing and artistic joy in a day with mysterious chrominance and texture variation. For instance, at dusk and dawn (golden hours), the sky becomes yellowish-orange, and buildings are coated with warm daylight; at night, buildings are lightened with colorful glows or have inner gleaming lights on.

- *Yingshu Chen, Tuan-Anh Vu, Ka-Chun Shum, and Sai-Kit Yeung are with the Hong Kong University of Science and Technology.*
- *Binh-Son Hua is with VinAI Research, Vietnam.*

source / label	day	golden	blue	night	total
training set	7,382	5,488	3,397	4,024	21,291
unpaired imgs	6,463	3,942	3,567	2,936	16,908
video frames	919	1,546	830	1,088	4,383
evaluation set	384	275	145	199	1,003

TABLE 1: Data statistics.

Fig. 1: Examples of photos in different time slots in a day: daytime, golden hour, blue hour, and nighttime. Photos by Unsplash users `lisanto_12`, `bartmynameisbart`, `lanceanderson`, `christopher__burns`.

2 Network Implementation Details

This section describes details about the network architecture and training settings in the image translation module.

2.1 Network Architecture

Our style transfer network consists of a content encoder E^c, a content specific domain mapping M, a style encoder E^s and a generator G for each domain. The generator can transfer style from source domain to target domain given source content and target style representations. For two domains X_1 and X_2, we train both transfer directions simultaneously, i.e., from X_1 to X_2 and from X_2 to X_1.

The content encoder consists of three convolutional layers for down-sampling, four residual blocks, one shared residual block for both domains. A de-convolutional layer plus a convolutional layer is used for domain mapping. Style encoder is designed with five convolutional layers followed by a global average pooling and a fully-connected layer. The length of style latent code is 8. Our generator contains four residual blocks, and 5 layers of upsample and convolutional

Fig. 2: Examples of segmentation of an outdoor architectural scene. Photos by Unsplash users `jose_maria_sava`, `michael75`

layers. Our multi-scale discriminator has four convolution layers and one fully-connected layer on each scale. By default, we set it on three scales. We use Leaky ReLU activation for discriminator, the shared residual block in the content encoder, and use ReLU for style encoder, content encoder(except for the shared residual block), and generator except for the last de-convolutional layer to which tangent activation function is applied. We apply Instance Normalization (IN) to the content encoder and all residual blocks, Adaptive Instance Normalization (AdaIN) [6] to residual blocks in the generator, and Layer Normalization to convolutional layers in the generator.

Both foreground and background models use the same network architecture.

We use similar annotation to [7], c7s1-64 stands for 7×7 convolutional block with 64 filters and stride 1, uk denotes a 2 nearest-neighbor upsampling layer followed by a 5×5 convolutional block with k iterations and stride 1, rb, dc, GAP, fc stands for residual block, de-convolutional layer, global average pooling layer, and fully-connected layer.

Generation architecture details:

- Content encoder E^c: c7s1-64, c4s2-128, c4s2-128, rb3s1-128 × 4, rb3s1-128 (shared)
- Domain mapping M: dc3s2-128, c4s2-128
- Style encoder E^s: c7s1-64, c4s2-128, c4s2-256×3, GAP, fc8
- Decoder G: rb3s1-128×4, u128, c5s1-2, u64, c5s1-2, c7s1-3

Discriminator D architecture details: c4s2-64, c4s2-128, c4s2-256, c4s2-512, fc1

3 Perceptual Study Details

Two surveys were conducted for perceptual study, in terms of image photorealism, and structure similarity and style consistency. Photorealism indicates how much the image looks real as a photo. Structure similarity indicates how much the scene in the generated image looks the same as the scene in the input image. Style consistency indicates how accurate the color styles are transferred semantic accordingly (i.e. static foreground style to foreground, dynamic background style to background).

There are 73 participants in our user study. The participants are general audience with ages between 18 and 30. Before the participants started to fill out the questionnaires, they are required to read the brief introduction on style transfer with examples.

In the first questionnaire, we prepared 3 images in different target time slots (i.e., golden, blue hours, nighttime) generated from each baseline and our approach, and from the real world . In total there were (Ours) × 3 + 10 (baselines) × 3 + (real-world) × 3 = 36 questions. As shown in Fig. 3, each question contains one image, and participants are asked if the displayed image is as realistic as a photograph. For each question, the participants need to select an option among *"Yes"*, *"No"* and *"Not Sure"* for the same question:

Q: "Does this image look real?"

We accumulated the total number of each option and calculated percentages among all options for each method or the real-world group as the photorealism score. From result of photorealism scores in the main paper, our generated results look more realistic than other baselines.

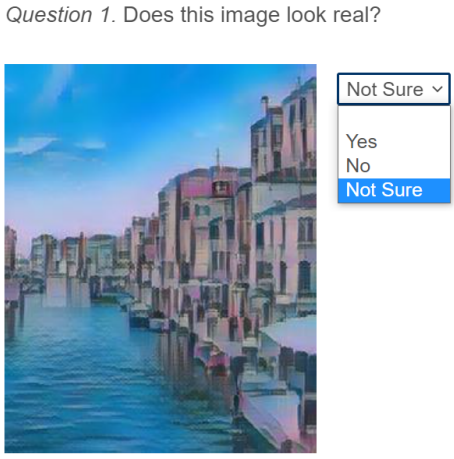

Fig. 3: Example of Photorealism Questions.

In the second questionnaire, we conducted a perceptual study via pairwise comparisons between all baselines and ours in terms of structural preservation (i.e., generated images have similar structure as the input image) and color matching (i.e., generated images have similar style to the reference image). For each baseline, we prepared 6 comparison pairs comprised of 3 time slots by 2 different scenes. In total there were 10 (baselines) × 6 (comparison pairs) = 60 pair-wise questions in this questionaire. As shown in Fig. 4, each question contains an input image, a style image, and two results from a baseline and our method respectively. The participants are asked to select a better result (baseline's or ours) or an option of *"Not sure"* for the question:

Q: "Which image looks better for you in terms of structural preservation (has similar structure as the input image), color correctness (has similar style as the style image)?"

As can be seen in pairwise results in main paper, our method outperforms the baselines, producing more natural matched color transfer and better structure preservation.

After taking the surveys, participants were asked to give reasons and feedback about their choices. We randomly picked unselected results in second survey and asked the participants to give reasons why they do not prefer these images.

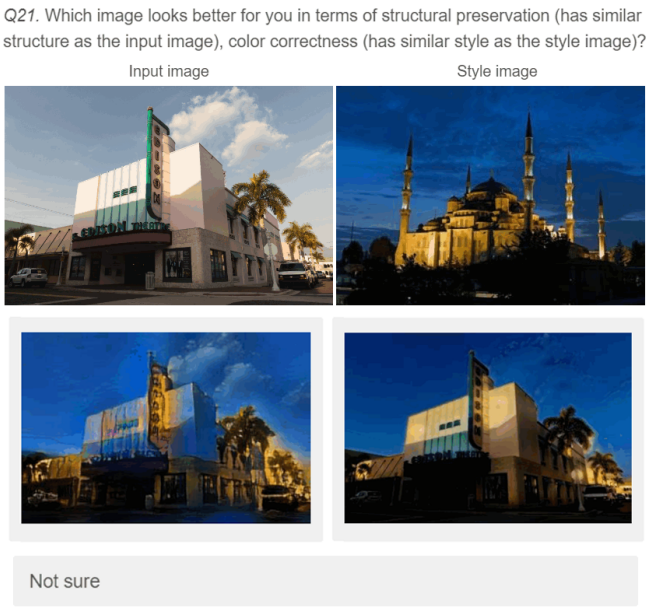

Fig. 4: Example of Pairwise Comparisons. Comparison between each baseline and our method in terms of structure similarity and style consistency.

The major comments we received on deficient image quality in the user study are summarized as follows:

- Some edges in the images were broken or distorted.
- Some smooth image areas were destroyed or removed.
- There is wrong sunlight direction (e.g., the sun is at the backside of the building but the light reflection is at the front side of the building).

From these feedbacks, humans concern much edge or contour information and clear appearance for a photorealistic image. People can easily perceive stereo geometry from an image and illumination effect in real 3D world, which 2D-based image approach is hard to realize. Taking lighting direction into account for time-of-day style transfer can be an interesting future work.

4 Additional Results

4.1 Metric Details

We describe the implementation of our metrics in the following paragraphs. All metrics excluding IS are computed for three types of style transfers, i.e., daytime to golden hours, blue hours, and nighttime, and get average score. In the main manuscript, we only report the mean score of three style transfers. We supplement all results in Section 4.2.

Translation accuracy. InceptionV3 [16] classifier can classify images to different classes (domains). Similar to [10], we trained an InceptionV3 classification model with our dataset for three classes (golden, blue and nighttime). Before training, we extended the data scale by random cropping to increase the accuracy. Accuracy rates of top-1 prediction by the trained classifier for each translation are computed. High accuracy indicates good style transfer to the target domain.

Quality Diversity. Inception score (IS) [17] measures how realistic the generation is and how much variety of output in an objective computation way. The trained InceptionV3 classifier with our dataset is used to calculate the IS score for all generations (with three styles) by each method.

Geometry Similarity. To evaluate geometry similarity, the typical differentiable structural similarity (SSIM) index [18] estimates the structural similarity in terms of luminance, contrast and structure between two images. Thus large luminance difference (e.g. day to night style transfer) downgrades SSIM measurement of geometry preservation. Instead of directly using SSIM, we calculate *Edge Conditioned SSIM* (edge-SSIM). edge-SSIM computes image structural similarity (SSIM) between Canny edge detected maps [19] of images. It can well alleviates luminance influence on geometry (e.g., comparison in Tab. 4). To get edge-SSIM, we first obtain the edge map using Canny edge detection function in OpenCV, and then calculate the SSIM between input (daytime domain) and output (target domain) edge maps using the image processing toolbox scikit-image in Python.

Semantic Segmentation Accuracy. For segmentation, IoU is the area of overlap divided by the area of union between the predicted segmentation and the ground truth. We compute IoU using the same pretrained segmentation model mentioned in the main manuscript between all generations and ground-truth daytime inputs. Then we get average score for each style transfer. High IoU means that generated images preserve good structure and recognizable real-world style for architectural photos.

4.2 More Quantitative Results

Complete Quantitative Results

We supplement complete quantitative results for all style translations, e.g., daytime to golden, blue and nighttime. The complete metric evaluation results among baselines and ours are displayed in Tables 2, 3, 4, complete evaluation results of ablation study on geometry losses are shown in Table 6. **Bold** texts are best results, underlined texts are second best results. Table 5 illustrates the accuracy and IS of the evaluation set for reference.

The traditional SSIM has unreasonable numbers in some cases such as day-to-night style transfer with dramatic luminance change (see SSIM in Fig. 4). Our edge-SSIM is more robust and stresses the edge information for structure similarity.

Overall, among image-to-image translation methods, MUNIT can retain primal input appearance (e.g., high edge-SSIM and IoU) but cannot transfer sufficient style in generated images (e.g., low accuracy, low IS). DSMAP accomplishes correct (high accuracy) and diverse style transfer (high IS), but dramatically destroys original input geometry (low edge-SSIM, low IoU and unrecognizable visual results). StarGANv2 completely destroys original appearance (please refer to the supplementary html viewer) with relatively low edge-SSIM and IoU.

Neural style transfer approaches perform better on golden-style image generation (high classification accuracy for golden class), but fail to generate blue and night stylized images (low accuracy). Particularly, the state-of-the-art AdaAttN preserves more high-frequency geometry information (high edge-SSIM) than other neural style transfer methods but therefore somehow weakens its capability of

	DRIT++ [8], [9]	MUNIT [7]	FUNIT [10]	DSMAP [11]	StarGANv2 [12]	AdaIN [6]	SANet [13]	AdaAttN [14]	LST [15]	Ours	Ours-opt
SSIM↑	0.6093	0.5224	0.4373	0.4552	0.3645	0.4730	0.5150	0.6538	0.4913	0.6371	**0.7531**
e-SSIM↑	0.5563	0.5061	0.4934	0.4779	0.4794	0.4979	0.4988	0.5411	0.4938	0.6314	**0.8200**
Acc↑	92.88%	87.62%	83.30%	94.61%	73.22%	83.07%	92.78%	80.39%	89.69%	**96.27%**	93.93%
IoU↑	0.7182	0.7533	0.5047	0.4936	0.3591	0.6676	0.7278	0.6543	0.6170	0.7362	**0.7911**

TABLE 2: Evaluation results of Daytime to Golden Hour translation. **Bold** text indicates the best result; underlined text indicates the 2nd best results.

	DRIT++ [8], [9]	MUNIT [7]	FUNIT [10]	DSMAP [11]	StarGANv2 [12]	AdaIN [6]	SANet [13]	AdaAttN [14]	LST [15]	Ours	Ours-opt
SSIM↑	0.4211	0.3188	0.3905	0.4484	0.3349	0.4381	0.4894	0.6324	0.4651	0.5735	**0.6886**
e-SSIM↑	0.4881	0.5297	0.5010	0.5010	0.4767	0.5040	0.4924	0.5287	0.4974	0.6309	**0.8106**
Acc↑	82.12%	84.13%	65.82%	82.96%	0.8378	64.05%	61.26%	46.80%	57.16%	**91.13%**	83.81%
IoU↑	0.7011	0.7236	0.5470	0.5344	0.3794	0.6797	0.7254	0.6733	0.6473	0.7374	**0.7915**

TABLE 3: Evaluation results of Daytime to Blue Hour translation. **Bold** text indicates the best result; underlined text indicates the 2nd best results.

	DRIT++ [8], [9]	MUNIT [7]	FUNIT [10]	DSMAP [11]	StarGANv2 [12]	AdaIN [6]	SANet [13]	AdaAttN [14]	LST [15]	Ours	Ours-opt
SSIM↑	0.0312	0.4598	0.2302	0.0919	0.1962	0.2761	0.4004	0.5063	0.3251	0.4027	**0.4806**
e-SSIM↑	0.5198	0.6600	0.4932	0.4581	0.4774	0.4866	0.4649	0.4883	0.4796	0.6453	**0.7975**
Acc↑	92.10%	88.60%	82.29%	95.61%	90.19%	83.27%	41.45%	53.70%	74.57%	**97.16%**	92.46%
IoU↑	0.6553	**0.7378**	0.5902	0.4647	0.4915	0.6452	0.7017	0.6321	0.6150	0.7034	0.7318

TABLE 4: Evaluation results of Daytime to Nighttime translation. **Bold** text indicates the best result; underlined text indicates the 2nd best results.

	Acc-golden	Acc-blue	Acc-night	Acc-mean	IS
Eval	99.64%	98.62%	100%	99.42%	2.8340

TABLE 5: Accuracy and IS of evaluation as reference

	w/o $\mathcal{L}_{gd} + \mathcal{L}_{kl}$	w/o $\mathcal{L}_{kl}$	w/o $\mathcal{L}_{gd}$	$\mathcal{L}_{total}$	$\mathcal{L}_{total}$(opt)
golden	0.4626	0.5502	0.5110	0.6314	**0.8200**
blue	0.4753	0.5394	0.5201	0.6309	**0.8106**
night	0.5020	0.5720	0.5165	0.6453	**0.7975**
mean	0.4800	0.5539	0.5159	0.6359	**0.8094**

TABLE 6: Ablation study with edge-SSIM metric (↑) on geometry losses.

accurate style transfer (low accuracy). AdaIN has better style transfer ability (high IS and high accuracy) but is bad at content preservation. Only our method can generate outputs with both high similarity of structure and appearance (high edge-SSIM), and semantically correct and sufficient style transfer (high IoU, accuracy, IS).

Running Time

Under training condition described in the main paper (i.e., same workstation), training time of *MUNIT*, *DRIT++*, *DSMAP*, *StarGANv2* or *Ours* takes 2 to 3 days per model, and *FUNIT* takes over 4 days. For neural style transfer approaches, baseline models were trained for around 1 to 2 days.

To infer an image of 256×, all baselines and ours take about 100ms to 200ms per image, and our image blending optimization takes around 300ms per image with 1 or 2 iterations. Some WCT^2 results are shown in the interactive viewer and we use pretrained WCT^2 model. By contrast, WCT^2 takes several seconds to predict an image of 256× or doubles the time for an image of 512×.

4.3 More Qualitative Results

We show a complete visual comparison among different baselines. Please refer to the supplementary interactive html files to view the results.

Comparisons with I2I Translation Baselines

In the interactive viewer, we show comparison results among DRIT++ ([8], [9]), MUNIT [7], FUNIT [10], DSMAP [11], and StarGANv2 [12]. FUNIT, DSMAP and StarGANv2 largely distort the building structure and appearance while DRIT++, MUNIT can somehow preserve the geometry, but do not always have correct corresponding semantic style mapping. Ours keeps geometry information, also transfers sky style and texture, and transfers correct foreground color similar to reference style. Our blending optimized images recover much geometry detail with little color loss.

Comparisons with Neural Style Transfer Baselines

State-of-the-art speedy neural style transfer methods (AdaIN [6], SANet [13], LST [20], AdaAttN [14]) tend to produce artistic effects with non-photorealistic texture and strokes even if we trained them with higher content (or other related) loss weight. Our results always tend to generate more photorealistic stylized images.

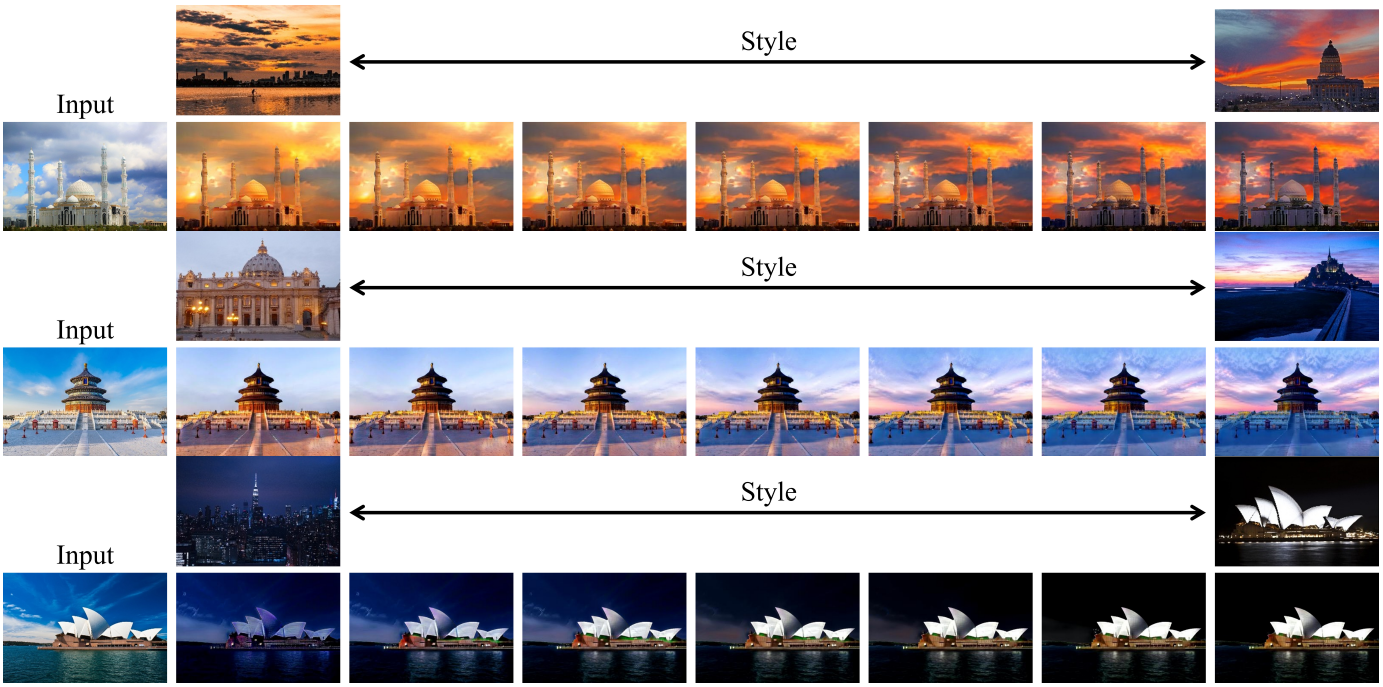

Fig. 5: Results of style interpolation.

WCT^2 [21] takes much longer time at inference time (a few seconds) compared to other baselines and ours. Since it takes segmentation as input too, we use the same foreground and background mask for both WCT^2 and our method to achieve fair comparison. From results, WCT^2 perfectly preserves both foreground and background geometry but tends to transfer smooth color style for both segments. In general, the transferred style of WCT^2 is not as impressive as ours or baselines'.

Stylization Diversity

Our models support style interpolation. We interpolate the outputs between two given style references by interpolating the style latent codes. The interpolation can generate smooth style transition as shown in the Fig.5.

Fig.7 illustrates diverse style transfers given different style references. Our method support stylization across different architectural images and styles.

Comparisons of Blending and Harmonization Techniques

To validate the effectiveness of our blending optimization, we show visual comparison of simple foreground and background addition *Copy and Paste*, our blending optimized result, *Deep Image Blending* (DIB) [22] and *Deep Image Harmonization* (DIH) [23] results in Figure 6. We used (a) *Copy and Paste* result as target, foreground or background as source for DIB and DIH (results in (c-d) and (e-f)). DIB tends to blend target style to source to make whole style consistent, which thus destroys original geometry or style. DIH changes building color or sky color according to the background and impair original transferred colors style (e.g., it brightens the building in (e) or darkens the sky in (f)). Our optimization approach (b) is able to refine geometry with original input gradient and preserve transferred colors. Besides, DIB is slow and our method is about $5\times$ faster than DIB. On average DIB takes 2m14s to blend an image while ours takes 0.27s. Our blending optimization supports high resolution restoration unlike other blending or harmonization approaches.

Fig. 6: Comparison to Blending and Harmonization Results. Deep Image Blending (DIB) uses (a) as target, (c) uses background image as source, (d) uses foreground as source. (e) and (f) are two Deep Image Harmonization (DIH) results. DIH is applied to *Copy and Paste* composite images shown in top-right insets with according foreground and background segmentation masks. Best view with zoom.

REFERENCES

[1] Y. Shih, S. Paris, F. Durand, and W. T. Freeman, "Data-driven hallucination of different times of day from a single outdoor photo," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 200, 2013. 1

[2] G. Marquès, "Understanding golden hour, blue hour and twilights," accessed November 28, 2020). [Online]. Available: https://www.photopills.com/articles/understanding-golden-hour-blue-hour-and-twilights 1

[3] "Pexels," accessed November, 2020. [Online]. Available: https://www.pexels.com/ 1

[4] "Pikwizard," accessed November, 2020. [Online]. Available: https://pikwizard.com/ 1

[5] "Unsplash," accessed November, 2020. [Online]. Available: https://unsplash.com/ 1

[6] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1501–1510. 2, 4

[7] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 172–189. 2, 4

[8] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 35–51. 4

[9] H.-Y. Lee, H.-Y. Tseng, Q. Mao, J.-B. Huang, Y.-D. Lu, M. Singh, and M.-H. Yang, "Drit++: Diverse image-to-image translation via disentangled representations," *International Journal of Computer Vision*, vol. 128, no. 10, pp. 2402–2417, 2020. 4

[10] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz, "Few-shot unsupervised image-to-image translation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 10 551–10 560. 3, 4

[11] H.-Y. Chang, Z. Wang, and Y.-Y. Chuang, "Domain-specific mappings for generative adversarial style transfer," *arXiv preprint arXiv:2008.02198*, 2020. 4

[12] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8188–8197. 4

[13] D. Y. Park and K. H. Lee, "Arbitrary style transfer with style-attentional networks," in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5880–5888. 4

[14] S. Liu, T. Lin, D. He, F. Li, M. Wang, X. Li, Z. Sun, Q. Li, and E. Ding, "Adaattn: Revisit attention mechanism in arbitrary neural style transfer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6649–6658. 4

[15] X. Li, S. Liu, J. Kautz, and M.-H. Yang, "Learning linear transformations for fast arbitrary style transfer," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 4

[16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826. 3

[17] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, pp. 2234–2242, 2016. 3

[18] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004. 3

[19] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986. 3

[20] A. Liu, S. Ginosar, T. Zhou, A. A. Efros, and N. Snavely, "Learning to factorize and relight a city," in *European Conference on Computer Vision*. Springer, 2020. 4

[21] J. Yoo, Y. Uh, S. Chun, B. Kang, and J.-W. Ha, "Photorealistic style transfer via wavelet transforms," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9036–9045. 5

[22] L. Zhang, T. Wen, and J. Shi, "Deep image blending," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 231–240. 5

[23] Y.-H. Tsai, X. Shen, Z. Lin, K. Sunkavalli, X. Lu, and M.-H. Yang, "Deep image harmonization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3789–3797. 5

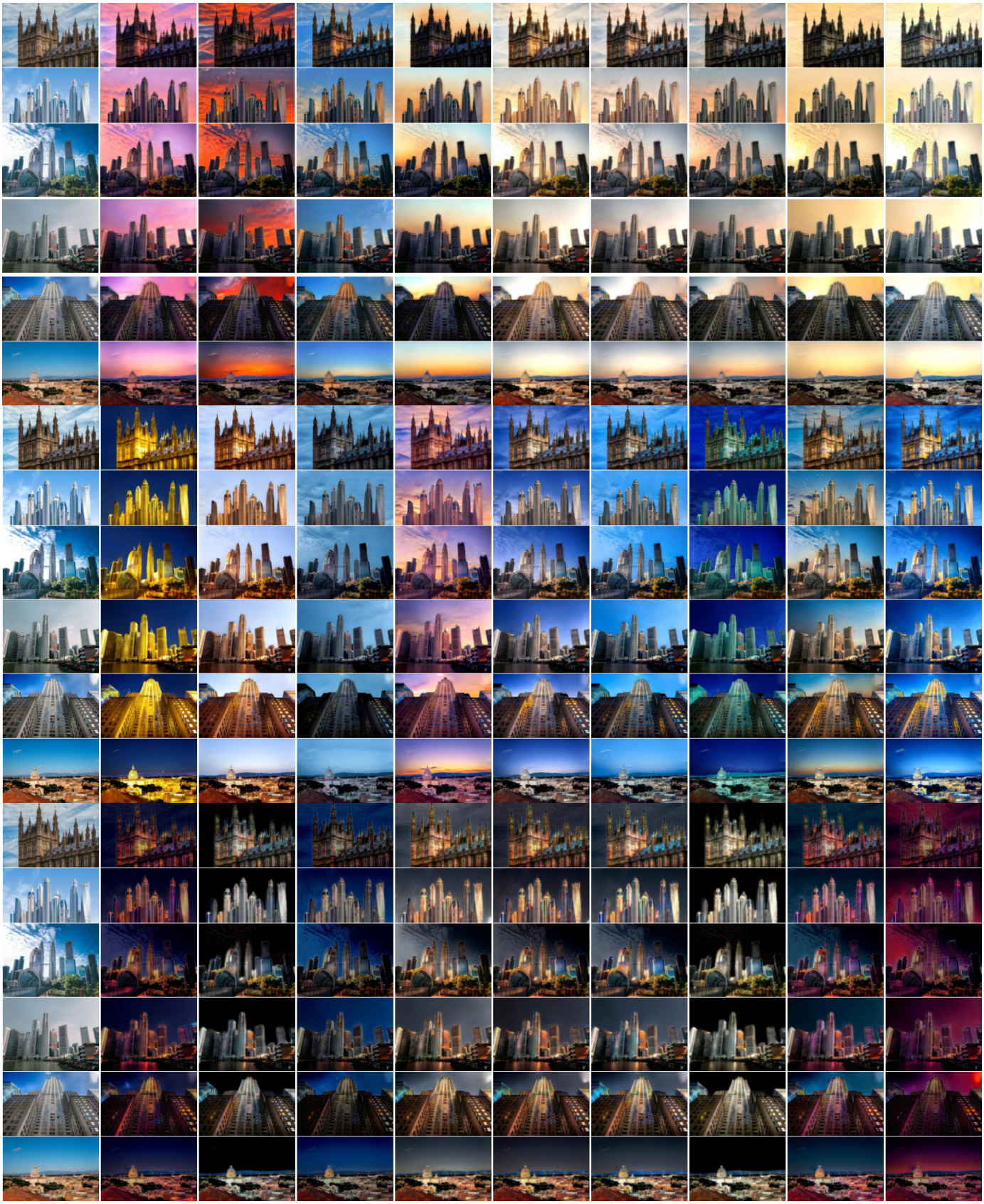

Fig. 7: Results of diverse styles under same scenes. Input in first column.