High-Resolution Daytime Translation Without Domain Labels

Supplementary Materials

1. Model description

The proposed HiDT model consists of a content encoder E_c, a style encoder E_s and a decoder G. During training, we also use two discriminators: a general discriminator D and a conditional discriminator D_s. Our training pipeline contains three branches:

Autoencoding branch. We decompose the original image into the content and style latent codes and reconstruct it afterwards

$$\mathbf{c} = E_c(\mathbf{x}),\ \mathbf{s} = E_s(\mathbf{x}),\ \tilde{\mathbf{x}} = G(\mathbf{c}, \mathbf{s})_x,$$

where $G(\cdot,\cdot)_x$ means that we take only the image from the decoder output and omit the predicted segmentation mask. L_1 distance estimates the discrepancy between the original and reconstructed images

$$\mathcal{L}_{\text{rec}} = \|\tilde{\mathbf{x}} - \mathbf{x}\|_1.$$

Noise branch. The original image $\mathbf{x}$ is translated to the random style $\mathbf{s}_r \sim p^*(\mathbf{s})$, sampled from the prior distribution. After that, the obtained image $\mathbf{x}_r$ is fed to the autoencoder

$$(\mathbf{x}_r, \mathbf{m}_r) = G(\mathbf{c}, \mathbf{s}_r),\ \tilde{\mathbf{c}}_r = E_c(\mathbf{x}_r),$$
$$\tilde{\mathbf{s}}_r = E_s(\mathbf{x}_r),\ \tilde{\mathbf{x}}_r = G(\tilde{\mathbf{c}}_r, \tilde{\mathbf{s}}_r)_x.$$

Adversarial losses enforce the plausibility of the generated image $\mathbf{x}_r$ and the dependency between $\mathbf{x}_r$ and $\mathbf{s}_r$.

$$\mathcal{L}^{D,r}_{\text{adv}} = \mathcal{L}^{D}_{\text{LS}}(D(\mathbf{x}), D(\mathbf{x}_r)) + \mathcal{L}^{D}_{\text{LS}}(D_{\text{s}}(\mathbf{x} \mid \mathbf{s}), D_{\text{s}}(\mathbf{x}_r \mid \mathbf{s}_r)),$$
$$\mathcal{L}^{G,r}_{\text{adv}} = \mathcal{L}^{G}_{\text{LS}}(D(\mathbf{x}_r)) + \mathcal{L}^{G}_{\text{LS}}(D_{\text{s}}(\mathbf{x}_r \mid \mathbf{s}_r)),$$

where $\mathcal{L}^{D}_{\text{LS}}$ and $\mathcal{L}^{G}_{\text{LS}}$ denote the least squares GAN [3] adversarial losses for the discriminator and the generator respectively. Latent reconstruction losses are applied to the extracted $\tilde{\mathbf{c}}_r$ and $\tilde{\mathbf{s}}_r$, while image reconstruction loss compares $\mathbf{x}_r$ and $\tilde{\mathbf{x}}_r$. The segmentation loss checks whether the

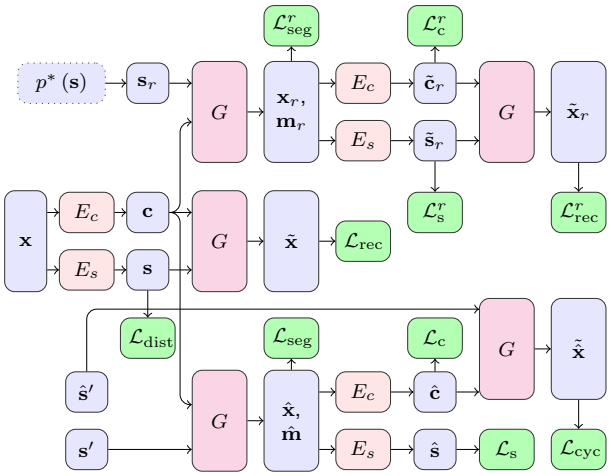


Figure 1: HiDT data flow. We show half of the (symmetric) architecture; $\mathbf{s}' = E_s(\mathbf{x}')$ is the style extracted from the other image $\mathbf{x}'$, and $\hat{\mathbf{s}}'$ is obtained similarly to $\hat{\mathbf{s}}$ with $\mathbf{x}$ and $\mathbf{x}'$ swapped. Light blue nodes denote data elements; light green, loss functions; others, functions (subnetworks). Functions with identical labels have shared weights.

obtained mask $\mathbf{m}_r$ mimics the externally provided [5] segmentation $\mathbf{m}$ of the original image $\mathbf{x}$

$$\mathcal{L}^r_{\text{s}} = \|\tilde{\mathbf{s}}_r - \mathbf{s}_r\|_1,\ \mathcal{L}^r_{\text{c}} = \|\tilde{\mathbf{c}}_r - \mathbf{c}\|_1,$$
$$\mathcal{L}^r_{\text{rec}} = \|\tilde{\mathbf{x}}_r - \mathbf{x}_r\|_1,\ \mathcal{L}^r_{\text{seg}} = \text{CE}(\mathbf{m}, \mathbf{m}_r),$$

where $\text{CE}(\mathbf{m}, \hat{\mathbf{m}})$ stands for the cross entropy between the original $\mathbf{m}$ and reconstructed $\hat{\mathbf{m}}$ segmentation masks

$$\text{CE}(\mathbf{m}, \hat{\mathbf{m}}) = -\sum_{(i,j)} m_{i,j} \log \hat{m}_{i,j}.$$

Swapping branch. This branch considers two real images $\mathbf{x}$ and $\mathbf{x}'$ that exchange the extracted styles between

each other.

$$\mathbf{s}' = E_s(\mathbf{x}'),\ (\hat{\mathbf{x}}, \hat{\mathbf{m}}) = G(\mathbf{c}, \mathbf{s}'),$$
$$\hat{\mathbf{c}} = E_c(\hat{\mathbf{x}}),\ \hat{\mathbf{s}} = E_s(\hat{\mathbf{x}}).$$

We apply the swapping twice to introduce the cross cycle consistency constraint below

$$\hat{\mathbf{s}}' = E_s(G(E_c(\mathbf{x}'), \mathbf{s})_x),$$
$$\tilde{\hat{\mathbf{x}}} = G(\hat{\mathbf{c}}, \hat{\mathbf{s}}').$$

The cross cycle consistency loss function intends to reconstruct the original image $\mathbf{x}$ after being transferred twice

$$\mathcal{L}_{\text{cyc}} = \left\| \tilde{\hat{\mathbf{x}}} - \mathbf{x} \right\|_1.$$

Other losses are similar to the noise branch, excluding the style reconstruction criterion $\mathcal{L}_{\text{s}}$: to avoid reducing the style encoder outputs $\hat{\mathbf{s}}$ and $\mathbf{s}'$ to zero, we apply a more robust objective than common L_1 distance

$$\mathcal{L}^D_{\text{adv}} = \mathcal{L}^D_{\text{LS}}(D(\mathbf{x}), D(\hat{\mathbf{x}})) + \mathcal{L}^D_{\text{LS}}(D_{\text{s}}(\mathbf{x} \mid \mathbf{s}'), D_{\text{s}}(\hat{\mathbf{x}} \mid \mathbf{s}')),$$
$$\mathcal{L}^G_{\text{adv}} = \mathcal{L}^G_{\text{LS}}(D(\hat{\mathbf{x}})) + \mathcal{L}^G_{\text{LS}}(D_{\text{s}}(\hat{\mathbf{x}} \mid \mathbf{s}')),$$
$$\mathcal{L}_{\text{s}} = \|\hat{\mathbf{s}} - \mathbf{s}'\|_1,\ \mathcal{L}_{\text{c}} = \|\hat{\mathbf{c}} - \mathbf{c}\|_1,$$
$$\mathcal{L}_{\text{cyc}} = \left\| \tilde{\hat{\mathbf{x}}} - \mathbf{x} \right\|_1,\ \mathcal{L}_{\text{seg}} = \text{CE}(\mathbf{m}, \hat{\mathbf{m}}).$$

In addition to the mentioned losses, our total objective also includes the style distribution loss function $\mathcal{L}_{\text{dist}}$, described in the main text, that aims to match the empirical distribution of extracted styles with the prior distribution $p^*(\mathbf{s})$.

2. Ablation Study

We conduct experiments to examine the influence of different parts of our model. As mentioned above, we use the noise and swapping branches to allow the model to perform both style swaps between images and random translation at the same time. Therefore, the main interest for us is to demonstrate the contribution of the individual parts of these branches.

Tab. 1 reports the metrics and the preference score of the full HiDT configuration against its ablated versions. The procedure of user study is the same as in the comparison with baselines in the main text. To check the statistical significance we test the hypothesis "User preference equals 0.5" against the alternative "User preference is less than 0.5".

The evaluation demonstrates that the usage of segmentation and style distribution loss may be redundant in some cases. However, the Fig. 5 of the main text provides one of the typical (though rare) examples, when segmentation loss could be profitable. The incorporating of style distribution loss could be acquitted by the desire to have a style space with predefined properties, though it does not bring benefits in terms of assessors' score.

Method	DIPD ↓ swapped	CIS ↑	IS ↑ swapped	User ↓ study (p-value)
HiDT	0.691	1.559	1.605	–
w/o Distribution Loss	0.585	1.618	1.596	0.48 (0.02)
w/o Segmentation	0.728	1.554	1.625	0.49 (0.09)
w/o AdaIN in Skip-Connections	0.77	1.601	1.557	0.53 (0.99)
w/o D_s	0.226	1.150	1.997	0.56 (0.99)
w/o Skip-Connections	0.867	1.531	1.566	0.59 (0.99)

Table 1: **Ablation Results:** We demonstrate the effect of different elements of our system. Note that model with the lowest DIPD score actually converged to a trivial (identity) solution. The results show novel Adaptive UNet architecture results in better quality while segmentation and style distribution losses are not necessary. Conditional discriminator D_s turns out in one of the key components of the HiDT pipeline.

3. Image translation results

We provide additional results of the HiDT model to illustrate the properties of the obtained style space and the plausibility of translated images. Fig. 2 shows the projection of our 3-dimensional style space, learned by the style encoder E_s, to the two-dimensional plane. Each style code is denoted with the thumbnail of an image it was extracted from.

Fig. 3 shows 2-dimensional style space of our model trained on datasets of facades [6], cityscapes [1], and maps vs aerial images merged with facades. The Figure demonstrates that our model is capable to separate styles from different discrete domains on these datasets.

Fig. 4 showcases the style swapping between two images. The trained model copes with exchanging the style while preserving the content.

We demonstrate the style interpolation by linearly interpolating between two style codes, extracted from different images, in Fig. 7, moving from cloudy to clear sky, and from dusk to sunset. Our model turns out to handle realistic artworks as well as photos. We show the results of translation for artworks in Fig. 8.

Translation of an image to styles, sampled from the prior distribution, for the model, trained on the Flowers dataset [4], is showcased in Fig. 5.

Additionally we train our model on a dataset of artworks from WikiArt. The architecture is the same but we use style vector of dimension 12 instead of 3. We train the network with batch 8 for 100000 iterations. The loss weights were set to $\lambda_1 = 5, \lambda_2 = 1, \lambda_3 = 0, \lambda_4 = 1, \lambda_5 = 0.1, \lambda_6 = 4, \lambda_7 = 6$. The rest training pipeline is the same (except the omission of segmentation masks). We show the results

of swapping style for model trained on images of artworks from WikiArt in Fig. 6.

4. Timelapse videos

The proposed approach may be used to generate static timelapses from a single image, using either an external video as a guidance or a predefined continuous trajectory from the latent style space. Note that modeling object motion (e.g. clouds) is out of scope for the present paper. Besides, we have no video consistency losses during the training process.

The attached supplementary video shows different timelapses, produced with the out of the model. We demonstrate sample frames with the illumination changes over time in Fig. 9 (translation networks outputs) and Fig. 10 (enhanced).

5. Enhancement

As was mentioned in the main text, we consider three upsampling schemes: guided image filtering [2], direct application of the trained fully convolutional translation network to the hi-res input, and our enhancement approach. All these methods have issues, showcased in Fig. 11, and overall the proposed method overcomes some limitations of its competitors.

References

[1] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

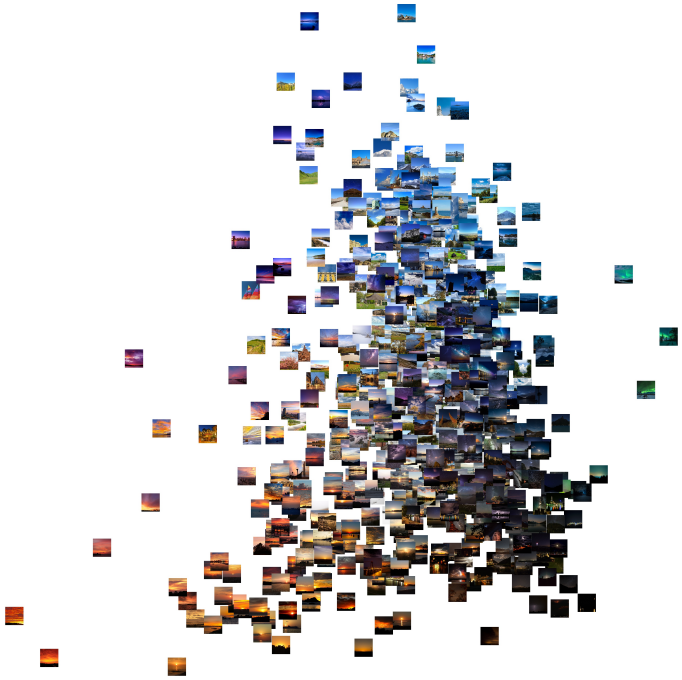
[2] K. He, J. Sun, and X. Tang. Guided Image Filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, June 2013. 3

[3] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2017. 1

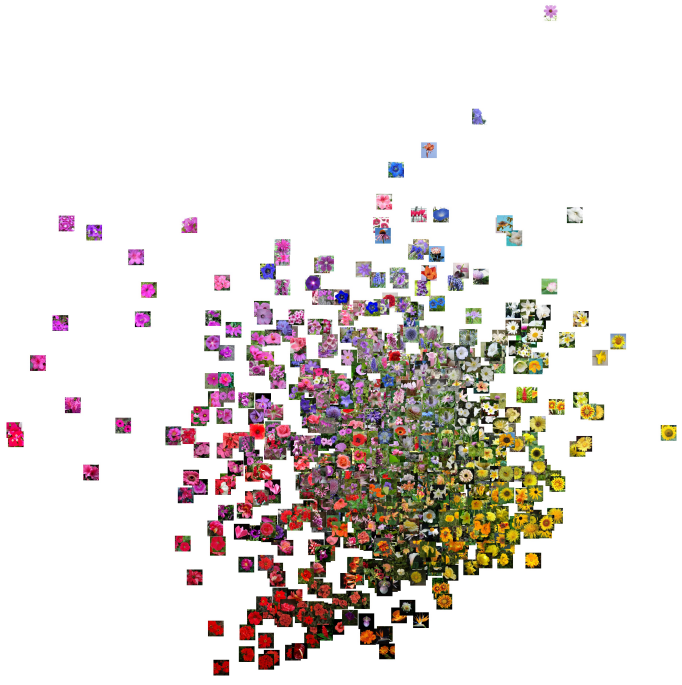
[4] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008. 2

[5] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[6] Radim Tyleček and Radim Šára. Spatial pattern templates for recognition of objects with regular structure. In *Proc. GCPR*, Saarbrucken, Germany, 2013. 2



(a)



(b)

Figure 2: Two-dimensional projection of the learned style space. Please zoom-in for details. (a) The main dataset of landscapes. (b) The flowers dataset. In both cases, images with similar type-of-lighting/color/time-of-the-day have been successfully grouped together in the style space.

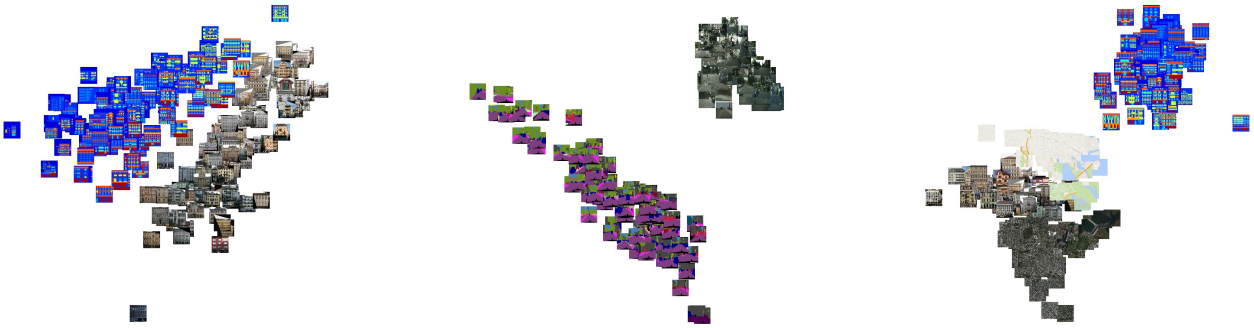


Figure 3: 2D style space for datasets of facades (left), cityscapes (middle), and maps vs aerial images merged with facades (right). A model separates styles from different discrete domains on these datasets. Please zoom in for details.

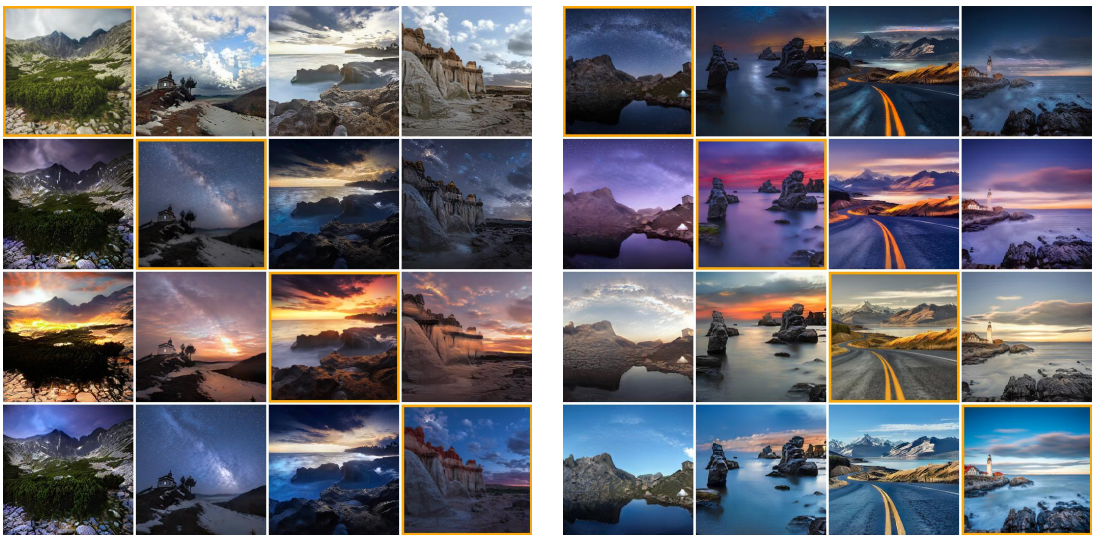


Figure 4: Swapping styles between two images. Original images are shown on the main diagonal, and off-diagonal images correspond to swaps. Swapping successfully combines both content and style from two input images into naturally-looking photographs.

Figure 5: To show the versatility of the approach, we apply it to the Oxford Flowers dataset. In each of three cases, a real image shown in the top-left is successfully translated to eight random "styles" (colormaps).

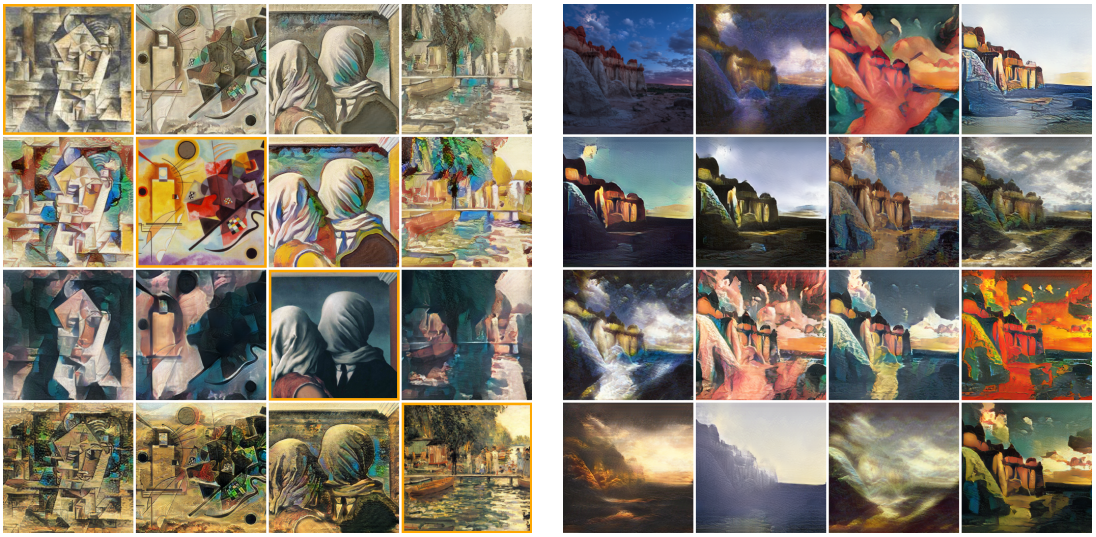
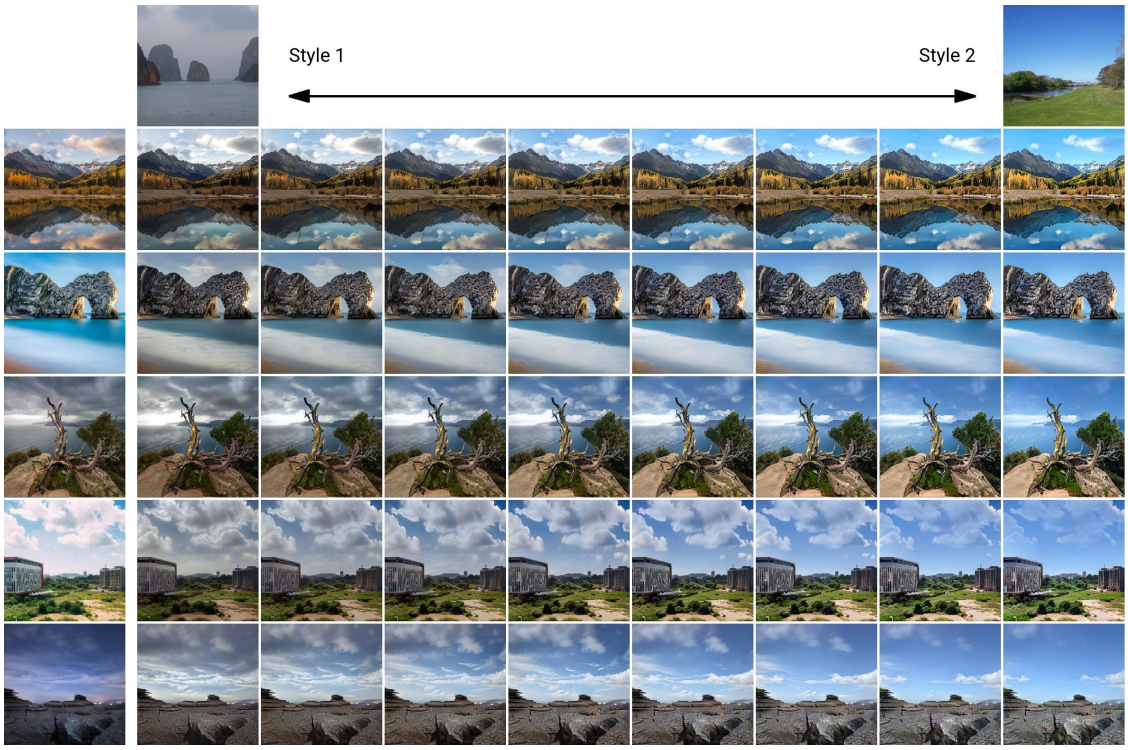
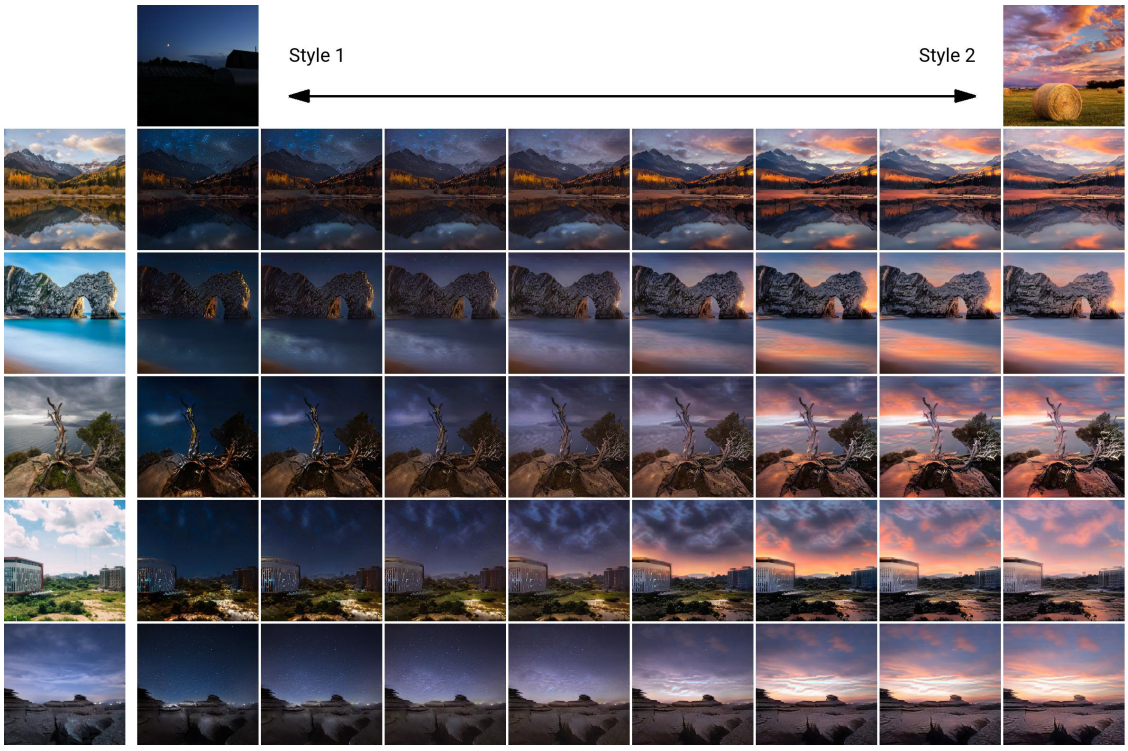


Figure 6: Style swapping and style sampling with the HiDT system trained on a paintings dataset. Left: Original images are shown on the main diagonal, and off-diagonal images correspond to swaps. Right: The original content image (top left), transferred to randomly sampled styles from prior distribution.

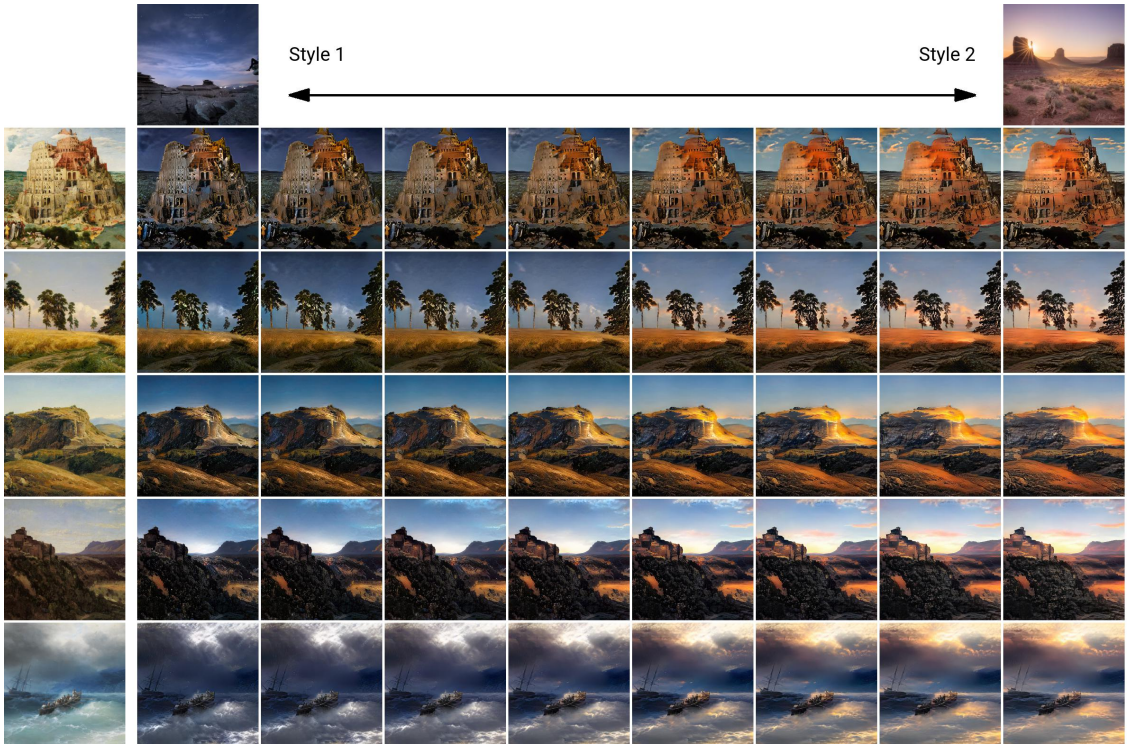


(a) Interpolation between cloudy and clear sky.

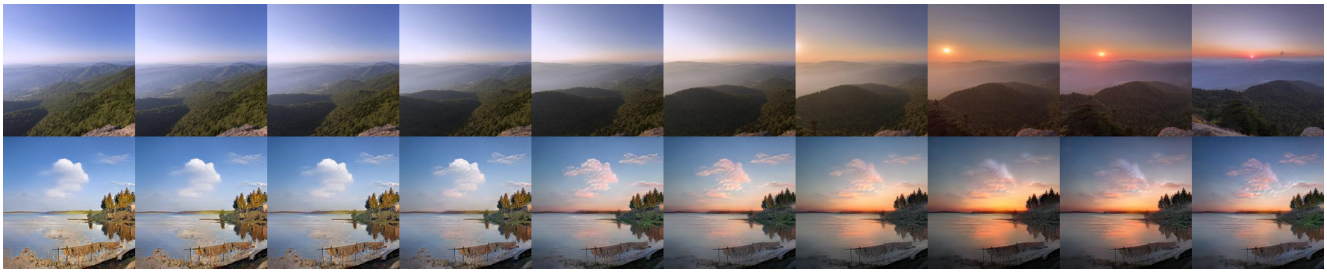


(b) Interpolation between dusk and sunset.

Figure 7: Linear interpolation between two extracted styles (lighting conditions) on our main dataset. The leftmost column contains original images while the topmost row contains the two images the endpoint styles were extracted from. Linear interpolation delivers smooth transition between styles.



(a) Linear interpolation between two extracted styles (lighting conditions) by the HiDT model, trained on our dataset of landscape photos. The leftmost column contains artworks in realism style.



(b) Our model is capable to generate timelapse from artworks. Top: the guidance real video. Bottom: the translated timelapse.

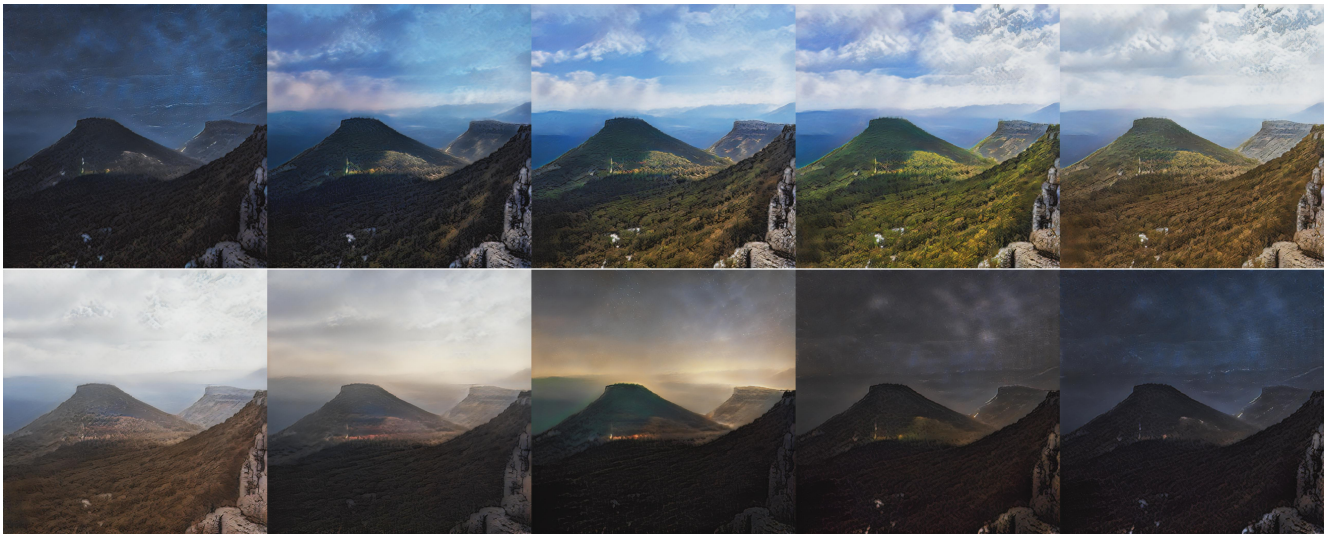
Figure 8: The result of our model, trained on the dataset of landscape photos, applied to the artworks from WikiArt. Our system can handle artworks despite not seeing paintings at train time.

(a)

(b)

(c)

Figure 9: Timelapse generation with a guidance video. Top: frames from the guidance video. Bottom: corresponding frames from the produced timelapse (translation network outputs). The original image is a “regular” landscape photo.



(a)

(b)

Figure 10: High-resolution timelapses obtained with a manually selected latent trajectory. The proposed enhancement scheme produces the resolution four times higher than the translation network output.

(a) Receptive field mismatch leads to the unrealistic glowing produced by the translation network, applied directly to hi-res input (third column).

(b) The guided filtering (second column) fails in highly detailed regions, resulting in blurry artifacts.

(c) While other methods affect the image contrast, our scheme (fourth column) enhances the output and produces detailed textures.

Figure 11: Comparison of different enhancement methods. Columns, left to right: lo-res translation network output; guided filter upsampling; the result of our translation network applied directly to the hi-res input; the result of our enhancement scheme. (a), (b) Top: the full image. Bottom: the selected crop.