High-Resolution Daytime Translation Without Domain Labels

I. Anokhin[1,*] P. Solovev[1,*] D. Korzhenkov[1,*] A. Kharlamov[1,*]
T. Khakhulin[1,3], G. Sterkin[1], A. Silvestrov[1], S. Nikolenko[2,1], V. Lempitsky[1,3]

[1]Samsung AI Center, Moscow
[2]National Research University Higher School of Economics, St.-Petersburg
[3]Skolkovo Institute of Science and Technology, Moscow

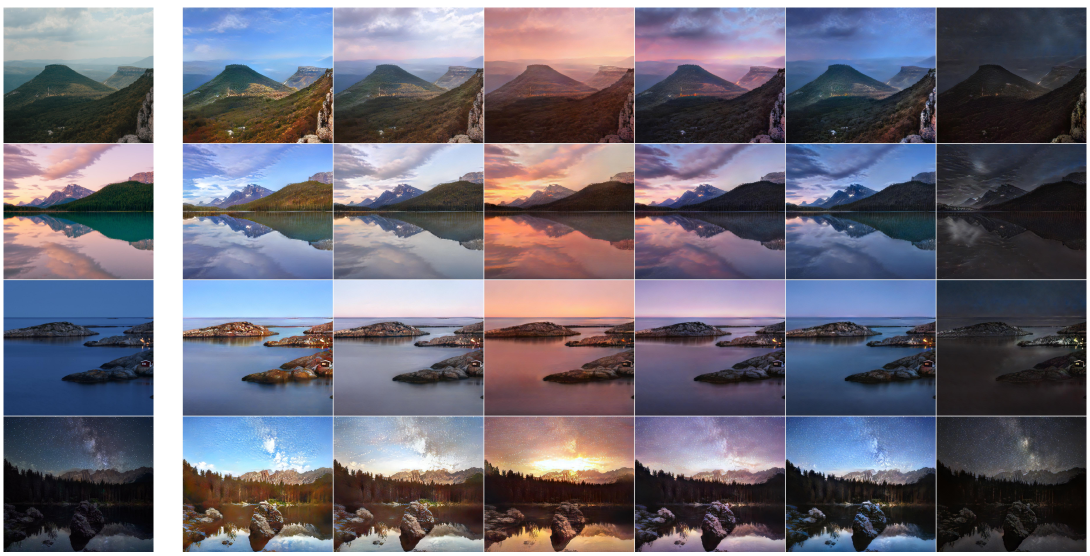


Figure 1: Daytime translation with HiDT: original images (left) and translated and enhanced images (one style per column).

Abstract

Modeling daytime changes in high resolution photographs, e.g., re-rendering the same scene under different illuminations typical for day, night, or dawn, is a challenging image manipulation task. We present the high-resolution daytime translation (HiDT) model for this task. HiDT combines a generative image-to-image model and a new upsampling scheme that allows to apply image translation at high resolution. The model demonstrates competitive results in terms of both commonly used GAN metrics and human evaluation. Uniquely, this good performance comes as a result of training on a dataset of still landscape images with no daytime labels available. Our results are available at `https://saic-mdal.github.io/HiDT/`.

[*] Equal contribution.

1. Introduction

Over the last years, the problem of image-to-image translation based on deep neural networks has evolved from translation between two predefined paired domains, as in the *pix2pix* framework [9], and unpaired approaches in CycleGAN [32] and others [16, 8], to the development of unified models for translation between multiple domains [2, 14, 13, 17]. Most classical approaches to image-to-image translation require domain labels. The recent FUNIT model [17] relaxes this constraint: to extract the style at inference time, it uses several images from the target domain as guidance for translation (known as the *few-shot* setting), but it still needs domain labels during training.

We arrive at the image-to-image translation problem from the practical task of generating daytime timelapse videos. Since obtaining a dataset of high-resolution diverse daytime timelapse videos is much harder than obtaining a dataset of various high-resolution images, we base our system on an image-to-image translation approach. Rather than collecting domain annotations, which are hard to define and hard to solicit from users in our case, we develop a method that uncovers the implicit domain structure of data without explicit domain supervision.

We thus train our model on a large dataset of unaligned images without domain labels. The only external (weak) supervision used by our approach are coarse segmentation maps estimated using an off-the-shelf semantic segmentation network. We demonstrate our results, in particular, on the task of photorealistic daytime altering for landscape images; prior works on this problem have always used paired or domain-labeled images for training.

Our contributions in this work are as follows. First, we present a method for image-to-image semantic preserving style transfer without knowledge of the domains represented in the dataset. We show that the internal bias of the collected dataset, the architectural bias, and a specially developed training procedure make it possible to learn style transformations even in this setting.

Second, to ensure fine detail preservation, we propose an architecture for image-to-image translation that combines the two well-known ideas: skip connections [23] and adaptive instance normalizations (AdaIN) [7]. We show that such a combination is feasible and leads to an architecture that preserves details much better than currently dominant AdaIN architectures without skip connections. We evaluate our system against several state-of-the-art baselines through objective measures as well as a user study. Apart from our main application, we show that our system can be used to learn multi-domain image stylization/recoloring, achieving quality on par with current state of the art.

Finally, since training a high-capacity image-to-image translation network directly at high resolution is computationally infeasible, we propose a new enhancement scheme that allows to apply the image-to-image translation network trained at medium resolution for high-resolution images.

The rest of the paper is organized as follows. Section 2 reviews related work. The main Section 3 presents the High-Resolution Daytime Translation (HiDT) model and the resolution-increasing enhancement model. Section 4 presents the results of a comprehensive experimental study, and Section 5 concludes the paper. Representative timelapse videos generated by our system are provided in the supplementary material.

2. Related work

Unpaired image-to-image translation. The task of image translation aims to transfer images from one domain to another (e.g., from summer to winter) or add/remove some image attributes (e.g., eyeglasses to a portrait). Many image translation models exploit generative adversarial networks (GAN) with conditional generators to inject information about the target attribute or domain [2]. Others [8, 13] split input images into content and style representations and subsequently edit the style to obtain the desired effect. In both cases, most works target the two-domain setting [32] or a setting with several discrete domains [2].

More closely related to our work, several recent approaches [8, 13, 11] split input images into content and style representations and subsequently edit the style to obtain the desired effect. The most common choice for generators uses adaptive instance normalization (AdaIN) in the encoder-decoder architecture [7]. Providing explicit domain labels is still mandatory for most multi-domain algorithms. The recently proposed FUNIT model [17] is designed for the case when those labels are used only by the conditional discriminator, while the generator is extracting some style code from given samples in the target domain. In this work, we take the next logical step in the evolution of GAN-based style transfer and do not use domain labels at all.

Timelapse generation. The generation of timelapses has attracted some attention from researchers, but most previous approaches use a dataset of timelapse videos for training. In particular, the work [25] used a bank of timelapse videos to find the scene most similar to a given image and then exploited the retrieved video as guidance for editing. Following them, the work [12] used a database of labeled images to create a library of transformations and apply them to image regions similar to input segments. Both methods rely on global affine transforms in the color space, which are often insufficient to model daytime appearance changes.

Unlike them, a recent paper [21] has introduced a neural generation approach. The authors leveraged two timelapses datasets: one with timestamp labels and another without them, both of different image quality and resolution. Finally, a very recent and parallel research [4] uses a dataset of diverse videos to solve the daytime appearance change modeling problems. Note that the method [4] also considers the problem of modeling short-term changes and rapid object motion, which we do not tackle in our pipeline.

Our approach is different from all previous works in that it needs neither timestamps nor paired datasets (such as, e.g., timelapse frames).

High-resolution translation. Modern generative models are often hard to scale to high-resolution input images due to memory constraints; most models are trained on either cropped parts or downscaled versions of images. Therefore, to generate a plausible image in high resolution one needs an additional enhancement step to upscale the translation output and remove artifacts. Such enhancement is closely related to the superresolution problem.

The work [15] compared photorealistic smoothing and image-guided filtering [5] and noted that the latter slightly degraded the performance as compared to the former, but led to a significant performance gain. Another way, proposed in [21], is to apply a different kind of guided upsampling via local color transfer [6]. However, unlike image-guided filtering, this method does not have a closed-form solution and requires an optimization procedure at inference time. In [4], the model predicts the parameters of a pixel-wise affine transformation of the downscaled image and then applies bilinear upsampling with these parameters to the full-resolution image. Unfortunately, both approaches often produce "glowing" artifacts near the edges.

The work most similar to ours in this regard, the *pix2pixHD* model [9], developed a separate refinement network. Our enhancement model is similar to their approach, as we also use the refinement procedure as a postprocessing step. But instead of training on the features, we use the output of low-resolution translation directly in a way inspired by classical multi-frame superresolution approaches [28].

3. Methods

Previous methods relied on some form of domain/attribute annotations to cope with the decomposition of the image into "content" and "style" that can be independently swapped between images. In particular, StarGAN [2] uses explicit discrete domain labels and trains dedicated encoders/decoders for each domain, FUNIT [17] utilizes a projection discriminator conditioned on the domain labels, and the content/style disentanglement model from [11] uses supervised metric learning losses in the style space to facilitate such a decomposition. Perhaps surprisingly, we have found that this decomposition can be achieved without domain labels, by using an appropriately chosen architecture and a training procedure. In what follows, we describe our loss function for HiDT and the training procedure in detail; graphically it is presented in detail in Fig. 2.

3.1. Components of the HiDT model

The HiDT model aims to extract independent encodings of content and style from an input image $\mathbf{x}$ by using its own architectural bias, with no explicit supervision from the training set, and then construct images with new content-style combinations. Similar to [8, 17], the problem is to generate an output image $\hat{\mathbf{x}}$ that takes the content from $\mathbf{x}$ and changes its style. However, we do not use conventional conditional GAN architectures [19] with categorical variables as conditions but rather define the task as transferring the style from a *style image* $\mathbf{x}'$ to a *content image* $\mathbf{x}$.

During training, the decoder is predicting not only the input image $\mathbf{x}$ but also a corresponding segmentation mask $\mathbf{m}$ (produced by an external pretrained network [27]). We do not aim to achieve state of the art segmentation as a byproduct, but segmentation in HiDT helps to control the style transfer and better preserve the semantic layout. Otherwise there is nothing preventing the network from repainting,

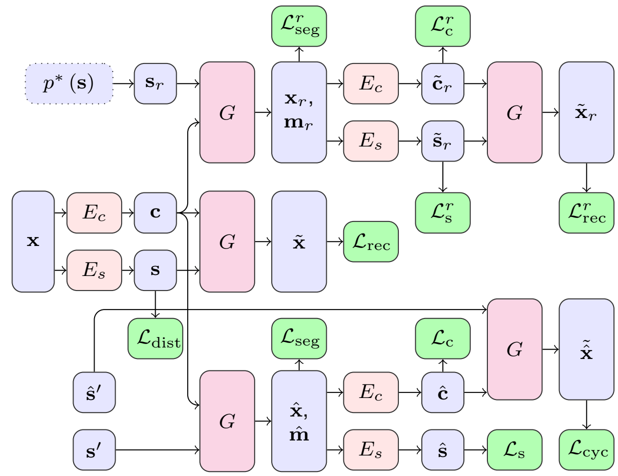


Figure 2: HiDT data flow. We show half of the (symmetric) architecture; $\mathbf{s}' = E_s(\mathbf{x}')$ is the style extracted from the other image $\mathbf{x}'$, and $\hat{\mathbf{s}}'$ is obtained similarly to $\hat{\mathbf{s}}$ with $\mathbf{x}$ and $\mathbf{x}'$ swapped. Light blue nodes denote data elements; light green, loss functions; others, functions (subnetworks). Functions with identical labels have shared weights.

say, grass into water or vice versa. Note that segmentation masks are *not* given as input to the networks, and are thus not needed at inference time.

Throughout this section, we denote the space of input images by $\mathcal{X}$, their segmentation masks by $\mathcal{M}$, and individual images with segmentation masks by $\mathbf{x}, \mathbf{m} \in \mathcal{X} \times \mathcal{M}$; the space of latent content codes $\mathbf{c}$ is $\mathbf{c} \in \mathcal{C}$, and the space of latent style codes $\mathbf{s}$ is $\mathbf{s} \in \mathcal{S}$ (as we will see below, $\mathcal{S} = \mathbb{R}^3$ while $\mathcal{C}$ has a more complex structure).

To extract $\mathbf{c}$ and $\mathbf{s}$ from an image $\mathbf{x}$, HiDT employs two encoders: $E_c : \mathcal{X} \to \mathcal{C}$ extracts the content representation $\mathbf{c}$ of the input image $\mathbf{x}$, and $E_s : \mathcal{X} \to \mathcal{S}$ extracts the style representation $\mathbf{s}$ of the input image $\mathbf{x}$. Given a latent code $\mathbf{c} \in \mathcal{C}$ and a style code $\mathbf{s} \in \mathcal{S}$, the decoder (generator) $G : \mathcal{C} \times \mathcal{S} \to \mathcal{X} \times \mathcal{M}$ produces a new image $\hat{\mathbf{x}}$ and the corresponding segmentation mask $\hat{\mathbf{m}}$. In particular, one can combine content from $\mathbf{x}$ and style from $\mathbf{x}'$ as $(\hat{\mathbf{x}}, \hat{\mathbf{m}}) = G\left(E_c(\mathbf{x}), E_s(\mathbf{x}')\right)$. HiDT operates by combining these three building blocks as follows, starting from two images $\mathbf{x}$ and $\mathbf{x}'$. See Figure 2 for a detailed illustration. The full description is given in the supplementary materials.

3.2. Loss functions in the HiDT model

Adversarial loss. First, we need to be able to generate realistic images, as defined in the usual adversarial fashion. To account for styles, we use two discriminators, unconditional $D : \mathcal{X} \to \mathbb{R}$ and conditional on the style vector $D_s : \mathcal{X} \times \mathcal{S} \to \mathbb{R}$. Both try to distinguish between real and translated images using the least squares GAN approach [18]. We define $\hat{\mathbf{x}} = G\left(E_c(\mathbf{x}), E_s(\mathbf{x}')\right)$ as a "fake" image produced from real content and style images

$\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, $\mathbf{s}' = E_s(\mathbf{x}')$. The same scheme is used for images produced with random style $\mathbf{s}_r \sim p^*(\mathbf{s})$.

We use the projection conditioning scheme [20] and detach the styles from the computational graph when feeding them to D_s during the generator update step. Real images use styles extracted from them, while generated images are coupled with styles they were translated to. Fig. 2 does not show adversarial losses.

Image reconstruction loss. The image reconstruction loss $\mathcal{L}_\mathrm{rec}$ is defined as the L_1-norm of the difference between original and reconstructed images. It is applied three times in the HiDT architecture: to the reconstruction $\tilde{\mathbf{x}}$ of the content image $\mathbf{x}$, $\mathcal{L}_\mathrm{rec} = \|\tilde{\mathbf{x}} - \mathbf{x}\|_1$, to the reconstruction $\tilde{\mathbf{x}}_r$ of the random style image $\mathbf{x}_r$, $\mathcal{L}^r_\mathrm{rec} = \|\tilde{\mathbf{x}}_r - \mathbf{x}_r\|_1$, and to the reconstruction $\tilde{\hat{\mathbf{x}}}$ of the image $\mathbf{x}$ from the content of the stylized image $\hat{\mathbf{x}}$ and the style of the stylized image $\hat{\mathbf{x}}'$ (cross cycle consistency): $\mathcal{L}_\mathrm{cyc} = \left\|\tilde{\hat{\mathbf{x}}} - \mathbf{x}\right\|_1$, where $\tilde{\hat{\mathbf{x}}} = G(\hat{\mathbf{c}}, \hat{\mathbf{s}}')$ (see Fig. 2).

Segmentation loss. The segmentation loss $\mathcal{L}_\mathrm{seg}$ is used together with the image reconstruction loss and defined as the cross entropy $\mathrm{CE}(\mathbf{m}, \hat{\mathbf{m}}) = -\sum_{(i,j)} m_{i,j} \log \hat{m}_{i,j}$ between the original $\mathbf{m}$ and reconstructed $\hat{\mathbf{m}}$ segmentation masks. It is applied twice: to the segmentation mask $\hat{\mathbf{m}}$ of the translated image, $\mathcal{L}_\mathrm{seg} = \mathrm{CE}(\mathbf{m}, \hat{\mathbf{m}})$, and to the mask $\mathbf{m}_r$ of the random style image, $\mathcal{L}^r_\mathrm{seg} = \mathrm{CE}(\mathbf{m}, \mathbf{m}_r)$.

Latent reconstruction loss. Two more reconstruction losses, $\mathcal{L}_\mathrm{s}$ and $\mathcal{L}_\mathrm{c}$, deal with the style and content codes; they are applied to the difference between original and reconstructed codes and used twice in the HiDT architecture. First, for the style $\tilde{\mathbf{s}}_r$ and content $\tilde{\mathbf{c}}_r$ of the random style image $\hat{\mathbf{x}}_r, \hat{\mathbf{m}}_r$, where the style should match $\mathbf{s}_r$ and the content should match $\mathbf{c}$: $\mathcal{L}^r_\mathrm{s} = \|\tilde{\mathbf{s}}_r - \mathbf{s}_r\|_1$, $\mathcal{L}^r_\mathrm{c} = \|\tilde{\mathbf{c}}_r - \mathbf{c}\|_1$. Second, for the style $\hat{\mathbf{s}}$ and content $\hat{\mathbf{c}}$ of the stylized image $\hat{\mathbf{x}}, \hat{\mathbf{m}}$, where the style should match $\mathbf{s}'$ and the content should match $\mathbf{c}$; we apply the L_1 loss to content, $\mathcal{L}_\mathrm{c} = \|\hat{\mathbf{c}} - \mathbf{c}\|_1$, and a more robust loss function to styles in order to avoid reducing them to zero: $\mathcal{L}_\mathrm{s} = \|\hat{\mathbf{s}} - \mathbf{s}'\|_1 + \frac{\|\hat{\mathbf{s}} - \mathbf{s}'\|_1}{\|\mathbf{s}'\|_1} + \mathrm{CosineDist}\left(\hat{\mathbf{s}}, \mathbf{s}'\right)$.

Style distribution loss. To enforce the structure of the space of extracted style codes, the style distribution loss inspired by the CORAL approach [26], is applied to a pool of styles collected from a number of previous training iterations. Namely, for a given pool size T we collect the styles $\{\mathbf{s}^{(1)}, \dots, \mathbf{s}^{(T)}\}$ from past minibatches with the *stop gradient* operation applied, add extracted styles $\mathbf{s}$ and $\mathbf{s}'$ (which are part of the current computational graph) to this pool, and calculate the mean vector $\hat{\boldsymbol{\mu}}_s$ and covariance matrix $\hat{\boldsymbol{\Sigma}}_s$ using the updated pool. Then the style distribution loss matches empirical moments of the resulting distribution to the theoretical moments of the random style vector generator $\mathcal{N}(\mathbf{0}, \mathbf{I})$: $\mathcal{L}_\mathrm{dist} = \|\hat{\boldsymbol{\mu}}_T\|_1 + \left\|\hat{\boldsymbol{\Sigma}}_T - \mathbf{I}\right\|_1 + \left\|diag(\hat{\boldsymbol{\Sigma}}_T) - \mathbf{1}\right\|_1$. Since the space $\mathcal{S} = \mathbb{R}^3$ is low-dimensional, and our target is the standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, this simplified approach suffices to enforce the structure in the space of latent codes. After computing the loss value, the oldest styles are removed from the pool to keep its size at T.

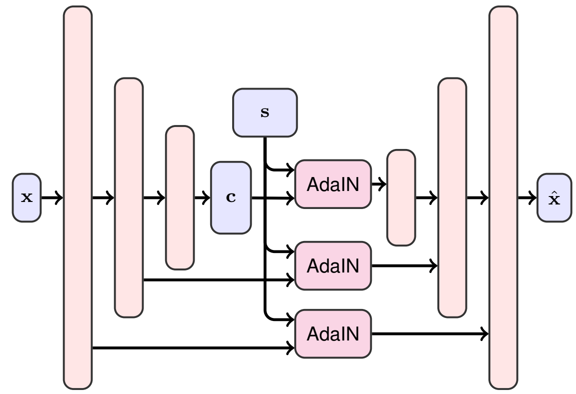


Figure 3: Diagram of the Adaptive U-Net architecture: an encoder-decoder network with dense skip-connections and content-style decomposition $(\mathbf{c}, \mathbf{s})$.

Total loss function. Thus, overall HiDT is jointly training the style encoder, content encoder, generator, and discriminator with the following objective:

$$\begin{aligned}\min_{E_c, E_s, G} \max_{D} \; &\mathcal{L}\left(E_c, E_s, G, D\right) = \lambda_1(\mathcal{L}_\mathrm{adv} + \mathcal{L}^r_\mathrm{adv}) + \\ &+ \lambda_2(\mathcal{L}_\mathrm{rec} + \mathcal{L}^r_\mathrm{rec} + \mathcal{L}_\mathrm{cyc}) + \lambda_3(\mathcal{L}_\mathrm{seg} + \mathcal{L}^r_\mathrm{seg}) + \\ &+ \lambda_4(\mathcal{L}_\mathrm{c} + \mathcal{L}^r_\mathrm{c}) + \lambda_5\mathcal{L}_\mathrm{s} + \lambda_6\mathcal{L}^r_\mathrm{s} + \lambda_7\mathcal{L}_\mathrm{dist}.\end{aligned}$$

Hyperparameters $\lambda_1, \dots, \lambda_7$ define the relative importance of the components in the overall loss function; they have been chosen by hand and will be shown below.

During our experiments, we have observed that the projection discriminator significantly improves the results, while removing the segmentation loss function sometimes leads to undesirable hallucinations in the generator (see Fig. 5 for an example). However, the model is still well trained without segmentation loss function and gets a comparable user preference score. Our experiments also demonstrate that the style distribution loss function is not necessary. We suggest this is due to the usage of both the projection discriminator and random styles during training. See supplementary materials for the ablation study.

3.3. Adaptive U-Net architecture

To create a plausible daytime landscape image, the model should preserve details from the original image. To satisfy this requirement, we enhance the FUNIT-inspired encoder-decoder architecture [17] with dense skip connections between the downsampling part of E_c and the upsam-

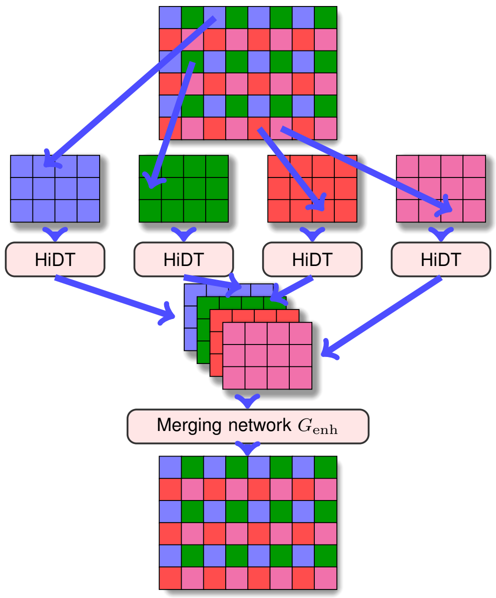


Figure 4: Enhancement scheme: the input is split into subimages (color-coded), translated individually by HiDT at medium resolution and then merged with G_{enh}. For illustration purposes, we show upsampling by a factor of two, but in the experiments we use a factor of four and also apply bilinear downsampling to prevent aliasing.

pling part of G. Unfortunately, regular skip connections would also preserve the style of the initial input. Therefore, we introduce an additional convolutional block with AdaIN [7] and apply it to the skip connections (see Fig. 3).

The overall architecture has the following structure: the content encoder E_c maps the initial image to a 3D tensor $\mathbf{c}$ using several convolutional downsampling layers and residual blocks. The style encoder E_s is a fully convolutional network that ends with global pooling and a compressing 1×1 convolutional layer. The generator G processes $\mathbf{c}$ with several residual blocks with AdaIN modules inside and then upsamples it.

3.4. Enhancement postprocessing

Training a network which can effectively operate with high resolution images is difficult due to the hardware constraints in both memory and computation time. Applying a fully convolutional neural network directly to a higher resolution image or using a guided filter [5] are both commonly used techniques to deal with high-resolution images. Although these techniques show good results in most cases, they have some limitations. A fully convolutional application might yield scene corruption due to limited receptive field, which is the case with sunsets where multiple suns might be drawn, or water reflections where the border between sky and water surface might be confused. Guided filter, on the other hand, works great with water or sun but fails if small details like twigs were changed by the style transfer procedure, or at the horizon or any other highly contrastive border if it was strongly affected, yielding "halo" effect. Such cases might seem a corner case not worth considering but they are crucial in the daytime translation task, which leads us to the necessity of having the semantic preserving upscale method. Also, straight-forward application of superresolution methods [31] and pretrained models are not possible due to the much bigger discrepancy between bicubic downsampling kernel and artifacts, yielded by image to image network. Please refer to the supplementary material for more results.

Inspired by existing multiframe image restoration methods [28], we propose to use a separate enhancement network G_{enh} to upscale the translated image and simultaneously remove the artifacts that are "typical" for the trained and frozen generator G. This approach is similar to [30], but we use several RGB images as input instead of feature maps (see below). Our method relies on usage of the generator in "autoencoder" mode to obtain a paired dataset, training the enhancement network in a supervised way and capturing the most common artifacts and discrepancies between the real image and the one produced by the generator. To further improve generalization to translated images we use generator in "random style" mode to obtain an additional unsupervised set, to which we do not apply supervised (perceptual and feature matching) losses. For brevity, below we discuss loss functions of for the "autoencoder" mode only.

Exactly, we cover a high resolution image $\mathbf{x}_{\text{hi}}$ (in our experiments, 1024×1024) with strongly overlapping frames $\{\mathbf{x}_{\text{hi}}^{(i)}\}_i$ of equal width and height with a stride of 1 pixel; each frame is only a few pixels smaller than $\mathbf{x}_{\text{hi}}$. The frames are downscaled with a bilinear kernel to the resolution suitable for the generator (in our case, 256×256 with scale factor 4), resulting in a set of downscaled crops $\{\mathbf{x}_{\text{lo}}^{(i)}\}_i$. We then apply HiDT to them, getting low resolution images $\{\hat{\mathbf{x}}_{\text{lo}}^{(i)}\}_i$, $\hat{\mathbf{x}}_{\text{lo}}^{(i)} = G(E_c(\mathbf{x}_{\text{lo}}^{(i)}), E_s(\mathbf{x}_{\text{lo}}^{(i)}))$. These frames are stacked into a single tensor in a fixed order and fed to the merging network G_{enh} that intends to restore the original image $\mathbf{x}_{\text{hi}}$, with the result $\hat{\mathbf{x}}_{\text{hi}} = G_{\text{enh}}\left(\{\hat{\mathbf{x}}_{\text{lo}}^{(i)}\}_i\right)$. The process is illustrated in Fig. 4.

For G_{enh}, we use the training setting of *pix2pixHD* [30] with perceptual, feature matching, and adversarial loss functions. We use high-resolution original images as supervision. G_{enh} uses the following loss functions during training: (1) perceptual reconstruction loss [31] between $\hat{\mathbf{x}}_{\text{hi}}$ and $\mathbf{x}_{\text{hi}}$: $\mathcal{L}_{\text{enh}}^{\text{perc}} = \|\text{VGG}(\hat{\mathbf{x}}_{\text{hi}}) - \text{VGG}(\mathbf{x}_{\text{hi}})\|_1$; (2) feature matching loss [30] between $\hat{\mathbf{x}}_{\text{hi}}$ and $\mathbf{x}_{\text{hi}}$, using each feature map of each discriminator (there are three of them in the multi-scale architecture): $\mathcal{L}_{\text{enh}}^{\text{feat}} = \sum_{\text{Layers } l} \left\| D_{\text{enh}}^{(l)}(\hat{\mathbf{x}}_{\text{hi}}) - D_{\text{enh}}^{(l)}(\mathbf{x}_{\text{hi}}) \right\|_1$; (3) ad-

N	HiDT vs `method`	User ↑ score	p-value	Adjusted p-value
1	`DRIT`	0.53	0.997	1.0
	`FUNIT-T`	0.51	0.904	0.999
	`FUNIT-O`	0.57	0.999	1.0
5	`FUNIT-T`	0.48	0.024	0.179
	`FUNIT-O`	0.55	0.481	1.0
10	`FUNIT-T`	0.47	0.001	0.011
	`FUNIT-O`	0.57	0.999	1.0

Table 1: User preference study of HiDT against the baselines. N is the number of styles averaged in the few-shot setting. The user score is the share of users that choose HiDT in the pairwise comparison. Our results show that all methods are competitive. The increase of N leads to the better quality of FUNIT-T.

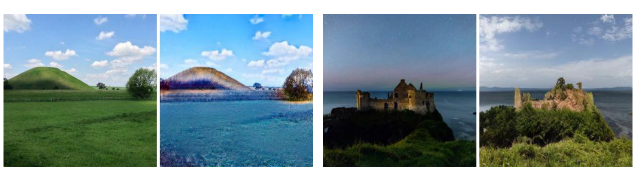
Method	DIPD↓ swapped	DIPD↓ random	CIS↑	IS↑ random	IS↑ swapped
`FUNIT-T`	1.168	-	1.535	-	1.615
`DRIT`	0.863	1.018	1.203	1.251	1.577
`HiDT-AE`	0.321	-	1.179	-	1.524
`HiDT`	0.691	0.88	1.559	1.673	1.605

Table 2: Performance comparison of three models using a hold-out dataset. FUNIT is not applicable in the random setting. According to the selected metrics, none of the models shows complete superiority over the others.

versarial loss based on LSGAN [18]: $\mathcal{L}_{\text{enh}}^{\text{adv}}(D) = \mathcal{L}_{\text{LS}}^{D}(D_{\text{enh}}(\mathbf{x}_{\text{hi}}), D_{\text{enh}}(\hat{\mathbf{x}}_{\text{hi}}))$, $\mathcal{L}_{\text{enh}}^{\text{adv}}(G) = \mathcal{L}_{\text{LS}}^{G}(D_{\text{enh}}(\hat{\mathbf{x}}_{\text{hi}}))$.

Success of our approach suggests that the same scheme can be applied to other image-to-image problems.

4. Experiments



(a) (b)

Figure 5: Training without segmentation losses is prone to failures of semantic consistency. Left: original images. Right: transferred images. (a) Our ablated model, trained without auxiliary segmentation task, turns grass into water; (b) FUNIT hallucinates grass on the building.

4.1. Daytime translation

Training details. In our experiments, the content encoder has two downsampling and four residual blocks; after each downsampling, only 5 channels are used for skip connections. The style encoder contains four downsampling blocks, and then the downsampled result is averaged with respect to spatial information into a three-dimensional vector. The decoder has five residual blocks with AdaIN inside and two upsampling blocks. AdaIN parameters are computed from a style vector via a three-layer feedforward network. Both discriminators are multi-scale, with three downsampling levels. We trained the translation model for 450 thousand iterations with batch size 4 on a single NVIDIA Tesla P40. For training, the images were downscaled to the resolution of 256×256. The loss weights were set to $\lambda_1 = 5, \lambda_2 = 2, \lambda_3 = 3, \lambda_4 = 1, \lambda_5 = 0.1, \lambda_6 = 4, \lambda_7 = 1$. We used the Adam optimizer with $\beta_1 = 0.5$, $\beta_2 = 0.999$, and initial learning rate 0.0001 for both generators and discriminators, halving the learning rate every 200000 iterations.

Dataset and daytime classifier. Following previous works, we collected a dataset of 20,000 landscape photos from the Internet. A small part of these images was manually labeled into four classes (night, sunset/sunrise, morning/evening, noon) using a crowdsourcing platform. A ResNet-based classifier was trained on those labels and applied to the rest of the dataset. We used predicted labels in two ways: (1) to balance the training set for image translation models with respect to daytime classes; (2) to provide domain labels for baseline models. Segmentation masks were produced by an external state of the art model [27] and reduced to 9 classes: sky, grass, ground, mountains, water, buildings, trees, roads, and humans.

Baselines. We used two recent image-to-image translation models as baselines: FUNIT [17] and Multi-domain DRIT++ [13] (refered to as DRIT for brevity). Both of them use domain labels. We trained the models on our dataset: DRIT with original hyperparameters, and FUNIT with both original (FUNIT-O) and properly tuned (FUNIT-T) hyperparameters. At inference time, FUNIT transfers the original image using styles extracted from other images, while DRIT in addition can transfer to randomly sampled styles. As another weak baseline, we train our model with only the autoencoding loss $\mathcal{L}_{\text{rec}}$ (HiDT-AE). The trained HiDT-AE still produces some color shifting when the styles are swapped; the result does not resemble the target daytime well enough, although it preserves the content (details) well.

Evaluation metrics. To compare our model with the baselines, we use several metrics, also commonly employed in previous works. The *domain-invariant perceptual distance* (DIPD) [8, 17] is the L_2 distance between normalized *Conv5* features of the original image and its translated version. It is used to measure content preservation. The *Inception score* (IS) [24] assesses the photorealism of generated images. We use the classifier described above to predict the domain label of the translated image. Styles for the translation may be either sampled from the prior distribution $p^*(\mathbf{s})$ (IS-random) or extracted from other images (IS-swapped). The *conditional inception score* (CIS) [8] measures the di-

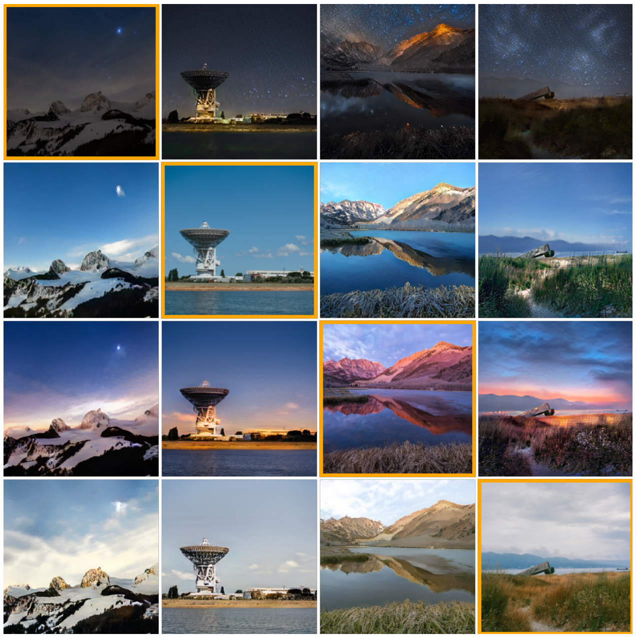


Figure 6: Visualization of swapping styles between two images. Original images are shown on the main diagonal. The examples show that HiDT (ours) is capable to swap the styles between two real images while preserving details.

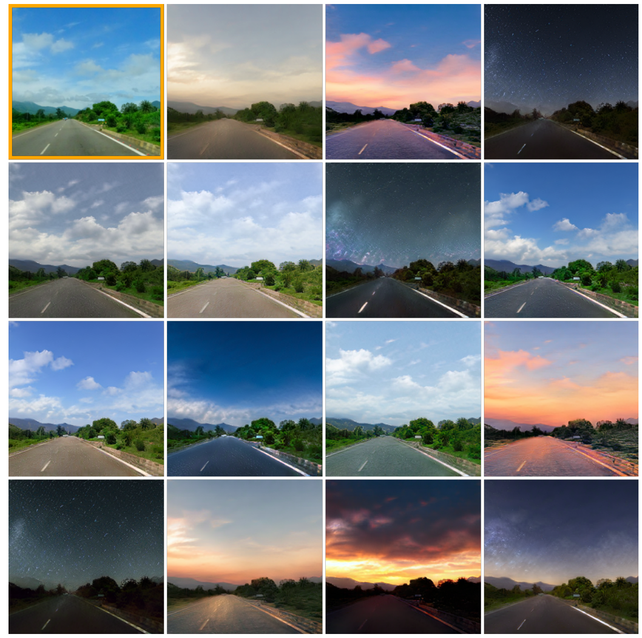


Figure 7: The original content image (top left), transferred to randomly sampled styles from prior distribution. The results demonstrate the diversity of possible outputs.

versity of translation results, which is suitable for our multi-domain setting. We calculate CIS for style swapping translation. To estimate the visual plausibility and photorealism of translation results, we use *human evaluation* with the following protocol. The assessors were given 1. the original image, 2. the image translated with our method, and 3. the image translated using one of the baseline models. We also show assessors the target label (time of day) and ask to choose the image that looks better with respect to both details preserved from the original image and the correct time of day. As both our model and FUNIT support the few-shot setting, styles for translation were obtained by averaging N styles extracted from images with the corresponding labels ($N = 1, 5, 10$). Assessment time was limited to 2 minutes per task, and original images were independently collected from the Internet. For each compared pair of methods, we generate 500 tuples, and each tuple is assessed by five different workers.

Results. Sample results of our image translation model are shown in the teaser figure on the first page. Fig. 6 illustrates style swapping between two different images, and image translation with styles randomly sampled from the prior distribution is shown in Fig. 7. In these experiments, we applied the truncation trick known for improving the average output quality [1, 10] at inference time. Random styles are sampled with reduced variance, and the styles extracted from other images are interpolated with the style extracted from the original image. One important application of our model is daytime timelapse generation using some video as a guidance; we showcase frames from such a timelapse in Fig. 9.

A visual comparison of our model with baselines is shown in Fig. 8. Results of different models are hard to distinguish, which is fully supported by our human evaluation study (Table 1). We report user preference of our model over the baselines and evaluate its statistical significance, applying the one-tailed binomial test to the hypothesis "User score equals 0.5" against "User score is less than 0.5". Due to multiple hypothesis testing, we also apply the Holm-Sidak adjustment and show adjusted p-values. Table 1 clearly shows that unlabeled training is sufficient for time-of-day translation. Traditional image-to-image translation metrics are summarized in Table 2. Again, we see that all models are on par with each other, with different winners according to different metrics. However, our model does not use domain labels.

4.2. Output enhancement

Training details. We used the RRBDNet architecture from ESRGAN [31] with 13 residual blocks for G_{enh} and a multiscale discriminator with three scales and four layers. We experimented with different crop sizes, finding the best results with crops from 9 to 16. The fewer number of crops (4 or 9) makes G_{enh} less robust to the artifacts of the translation network. We used multiplier coefficients of 10 for perceptual and feature matching losses and multiplier coefficients of 1 for adversarial loss and a learning rate of

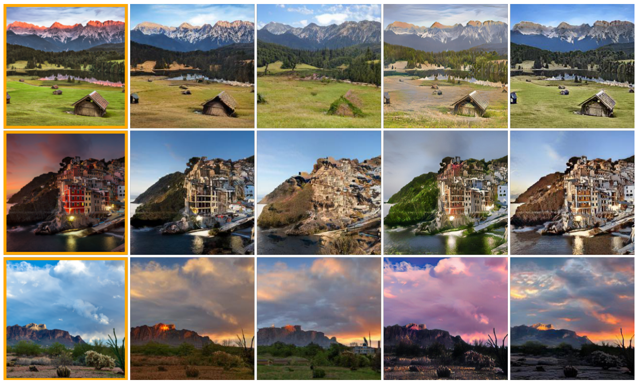


Figure 8: Comparison with baselines. Columns, left to right: original image, FUNIT-T, FUNIT-O, DRIT, HiDT (ours). Our model, trained and inferenced without domain labels, has translation quality similar to the models that require such supervision.

Figure 9: Timelapse generation using styles extracted from a real video. Top: frames from guidance video. Bottom: timelapse generated from a single image with extracted styles.

0.0001 for both the enhancer and discriminator.

Baselines. We compare the proposed enhancement scheme with the following baselines: (1) fully convolutional application of the translation network to a high-resolution image, (2) Lanczos upsampling. The *pix2pixHD* [30] enhancement scheme requires supervision for translated images. Therefore, we do not use *pix2pixHD* as a baseline.

Results. The resulting downsampled images produced with the enhancement procedure are presented in Fig. 1, and a detailed example is shown in Fig. 10. The latter figure contains image patch produced by different models and shows that our model is more plausible than the result of direct Lanczos upsampling: the rightmost patch contains more details from the original.

4.3. Additional task

To show the generality of the proposed HiDT approach, we additionally trained the image translation model on the *Flowers* dataset [22] for 60,000 iterations. Segmentation masks were not used in this experiment. The results of translation to random styles (with no enhancement) are presented in Fig. 11.

Figure 10: Enhancement of our translation network outputs with different methods. Columns, left to right: original image; result of our translation network applied directly to the hi-res input; lo-res translation output upsampled with Lanczos' method; the result of our enhancement network G_{enh}. In this example, direct application to hi-res turns water into sky with stars, while the enhancement network preserves the semantics of the scene.

Figure 11: Visualization of a flower image (original on the left) to several randomly sampled styles.

5. Conclusion

In this work, we have presented a novel image-to-image translation model that does not rely on domain labels during either training or inference. The new enhancement scheme shows promising results for increasing the resolution of translation outputs. We have also shown that our model is able to learn daytime translation for high-resolution landscape images and provided empirical evidence that our approach can be generalized to other domains.

The results show that our method is on par with state of the art baselines that require labels at least at training time. Our model can generate images using styles extracted from images, as well as sampled from the prior distribution. An appealing straightforward application of our model is the generation of timelapses from a single image (the task currently mainly tackled with paired datasets). One interesting direction for further work would be to unite the translation and enhancement networks into a single model trained end-to-end.

References

[1] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. 7

[2] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim, and J. Choo. StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, June 2018. 1, 2, 3

[3] 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. 12

[4] Yuki Endo, Yoshihiro Kanamori, and Shigeru Kuriyama. Animating landscape: Self-supervised learning of decoupled motion and appearance for single-image video synthesis. 38(6):175:1–175:19. 2, 3

[5] K. He, J. Sun, and X. Tang. Guided Image Filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, June 2013. 2, 5, 12

[6] Mingming He, Jing Liao, Dongdong Chen, Lu Yuan, and Pedro V. Sander. Progressive color transfer with dense semantic correspondences. *ACM Trans. Graph.*, 38(2):13:1–13:18, Apr. 2019. 3

[7] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1510–1519, Oct 2017. 2, 5

[8] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal Unsupervised Image-to-Image Translation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 179–196, Cham, 2018. Springer International Publishing. 1, 2, 3, 6

[9] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 3

[10] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 7

[11] Dmytro Kotovenko, Artsiom Sanakoyeu, Sabine Lang, and Bjorn Ommer. Content and style disentanglement for artistic style transfer. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2, 3

[12] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Qian, and James Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Trans. Graph.*, 33(4):149:1–149:11, July 2014. 2

[13] Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Singh, and Ming-Hsuan Yang. DRIT++: diverse image-to-image translation via disentangled representations. *CoRR*, abs/1905.01270, 2019. 1, 2, 6

[14] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse Image-to-Image Translation via Disentangled Representations. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, Lecture Notes in Computer Science, pages 36–52. Springer International Publishing, 2018. 1

[15] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A Closed-Form Solution to Photorealistic Image Stylization. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 468–483, Cham, 2018. Springer International Publishing. 2

[16] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised Image-to-Image Translation Networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 700–708. Curran Associates, Inc., 2017. 1

[17] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 1, 2, 3, 4, 6

[18] 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. 3, 6, 11

[19] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. 3

[20] Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. In *International Conference on Learning Representations*, 2018. 4

[21] Seonghyeon Nam, Chongyang Ma, Menglei Chai, William Brendel, Ning Xu, and Seon Joo Kim. End-to-end time-lapse video synthesis from a single outdoor image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 3

[22] 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. 8, 12

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume abs/1505.04597, 2015. 2

[24] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2234–2242. Curran Associates, Inc., 2016. 6

[25] Yichang Shih, Sylvain Paris, Frdo Durand, and William T. Freeman. Data-driven Hallucination of Different Times of

Day from a Single Outdoor Photo. *ACM Trans. Graph.*, 32(6):200:1–200:11, Nov. 2013. 2

[26] Baochen Sun, Jiashi Feng, and Kate Saenko. Correlation alignment for unsupervised domain adaptation. In Gabriela Csurka, editor, *Domain Adaptation in Computer Vision Applications*, pages 153–171. Springer International Publishing. 4

[27] 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. 3, 6, 11

[28] R Tsai. Multiframe image restoration and registration. *Advance Computer Visual and Image Processing*, 1:317–339, 1984. 3, 5

[29] Radim Tyleček and Radim Šára. Spatial pattern templates for recognition of objects with regular structure. In *Proc. GCPR*, Saarbrucken, Germany, 2013. 12

[30] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, June 2018. 5, 8

[31] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. ESRGAN: Enhanced super-resolution generative adversarial networks. In Laura Leal-Taix and Stefan Roth, editors, *In ECCV 2018 Workshops*, pages 63–79. Springer International Publishing. 5, 7

[32] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, Oct. 2017. 1, 2

Appendices

A. 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

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

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

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

where $\mathcal{L}_{\text{LS}}^{D}$ and $\mathcal{L}_{\text{LS}}^{G}$ denote the least squares GAN [18] 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 obtained mask $\mathbf{m}_r$ mimics the externally provided [27] segmentation $\mathbf{m}$ of the original image $\mathbf{x}$

$$\begin{aligned}\mathcal{L}_{\text{s}}^{r} &= \|\tilde{\mathbf{s}}_r - \mathbf{s}_r\|_1,\ \mathcal{L}_{\text{c}}^{r} = \|\tilde{\mathbf{c}}_r - \mathbf{c}\|_1,\\ \mathcal{L}_{\text{rec}}^{r} &= \|\tilde{\mathbf{x}}_r - \mathbf{x}_r\|_1,\ \mathcal{L}_{\text{seg}}^{r} = \text{CE}(\mathbf{m}, \mathbf{m}_r),\end{aligned}$$

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.

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

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

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

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

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

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 $p^*(\mathbf{s})$.

B. 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. 3 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 3: **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.

C. 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. 12 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. 13 shows 2-dimensional style space of our model trained on datasets of facades [29], cityscapes [3], 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. 14 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. 17, 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. 18.

Translation of an image to styles, sampled from the prior distribution, for the model, trained on the Flowers dataset [22], is showcased in Fig. 16.

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. 15.

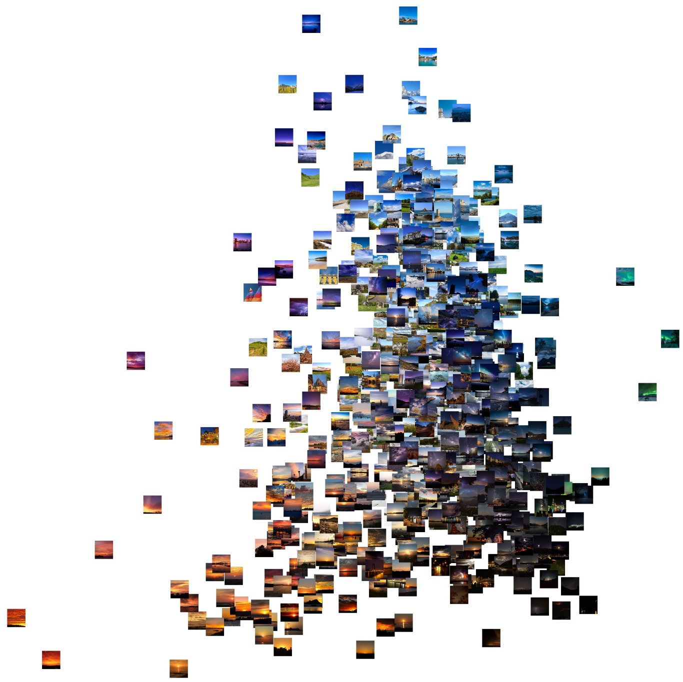
D. Timelapse videos

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.

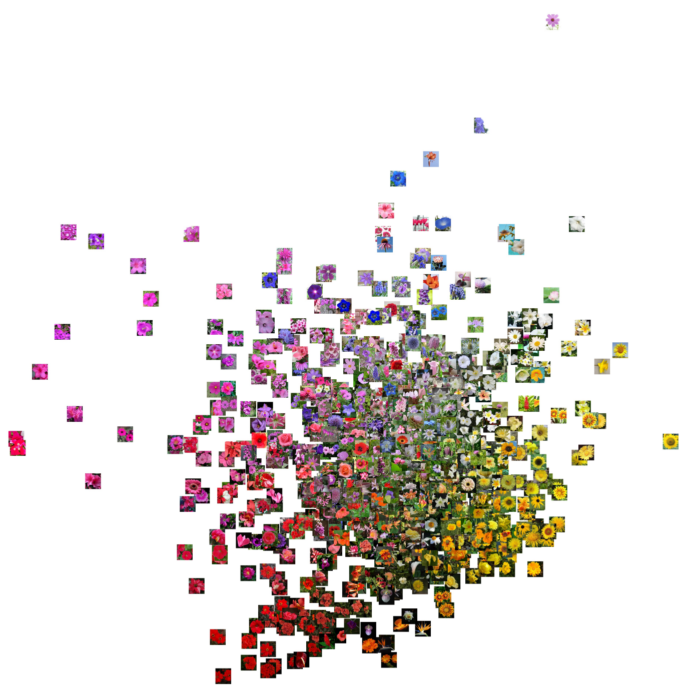
The attached supplementary video shows different timelapses, produced with the out of the model. We demonstrate sample frames with the illumination changes over time in Fig. 19 (translation networks outputs) and Fig. 20 (enhanced).

E. Enhancement

As was mentioned in the main text, we consider three upsampling schemes: guided image filtering [5], 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. 21, and overall the proposed method overcomes some limitations of its competitors.



(a)



(b)

Figure 12: 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.

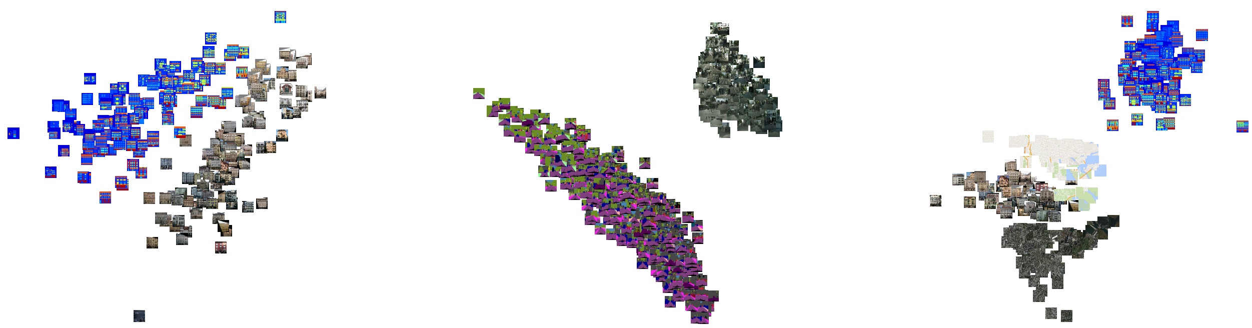


Figure 13: 2D style space for datasets of facades (left), cityscapes (middle), and maps vs aerial images merged with facades (right). A model separates styles from different discrete domains on these datasets. Please zoom in for details.

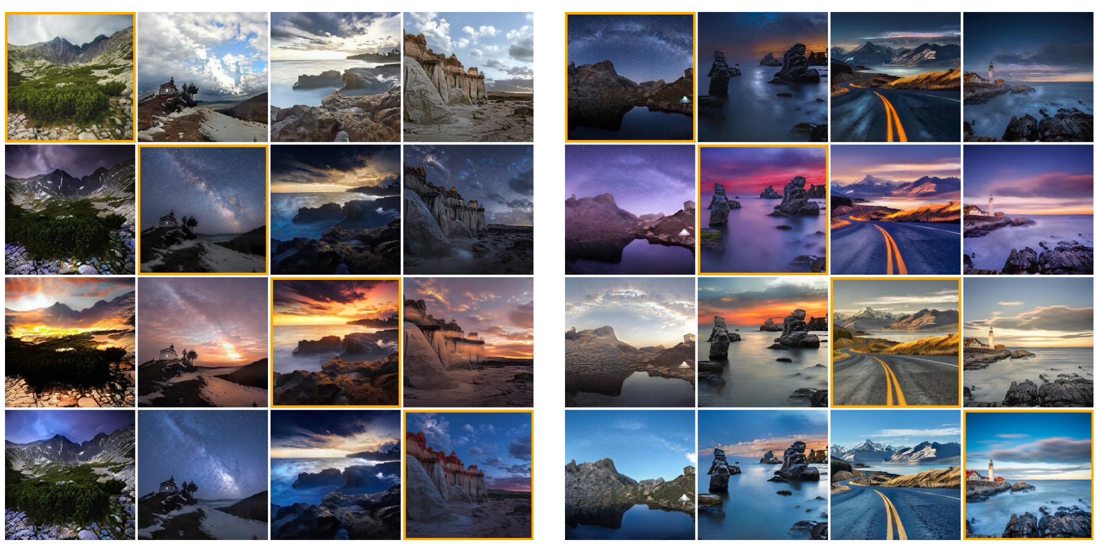


Figure 14: Swapping styles between two images. Original images are shown on the main diagonal, and off-diagonal images correspond to swaps. Swapping successfully combines both content and style from two input images into naturally-looking photographs.

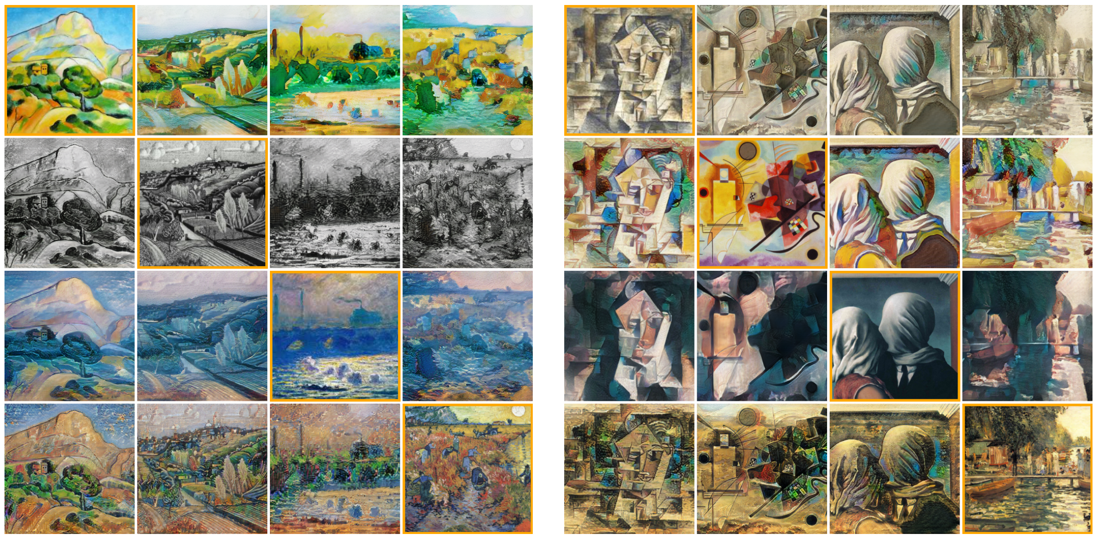
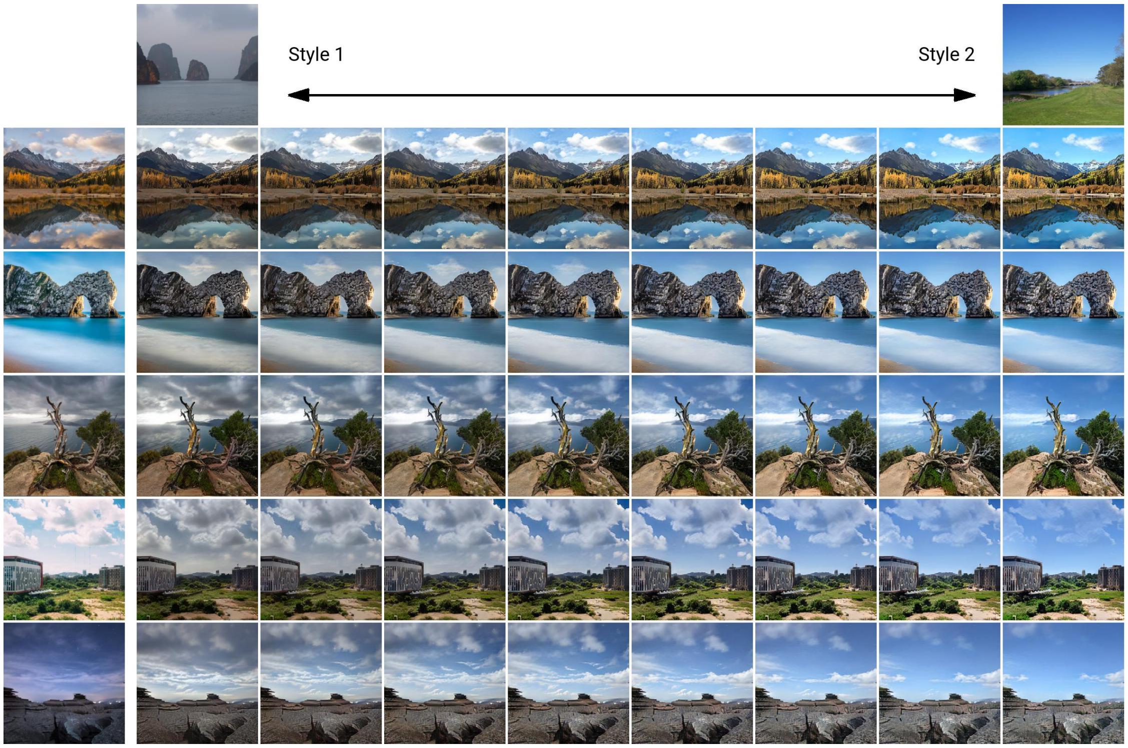
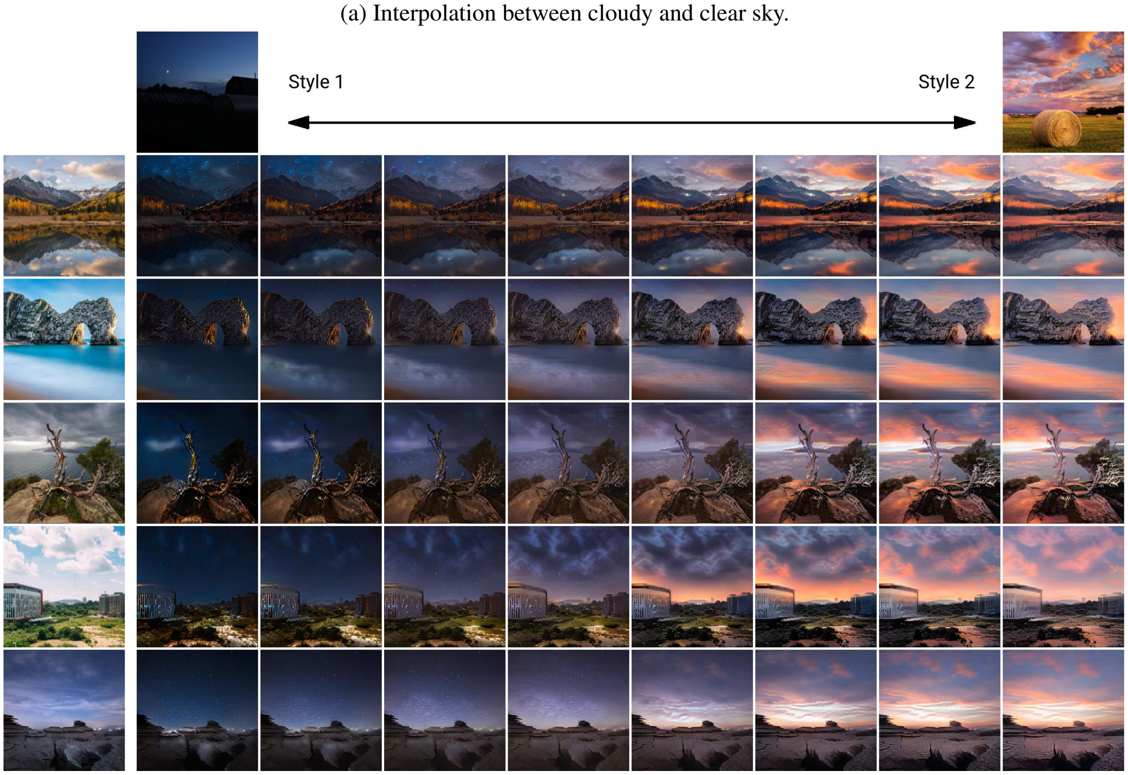


Figure 15: Swapping styles of art picture between two images. Original images are shown on the main diagonal, and off-diagonal images correspond to swaps.

Figure 16: 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).

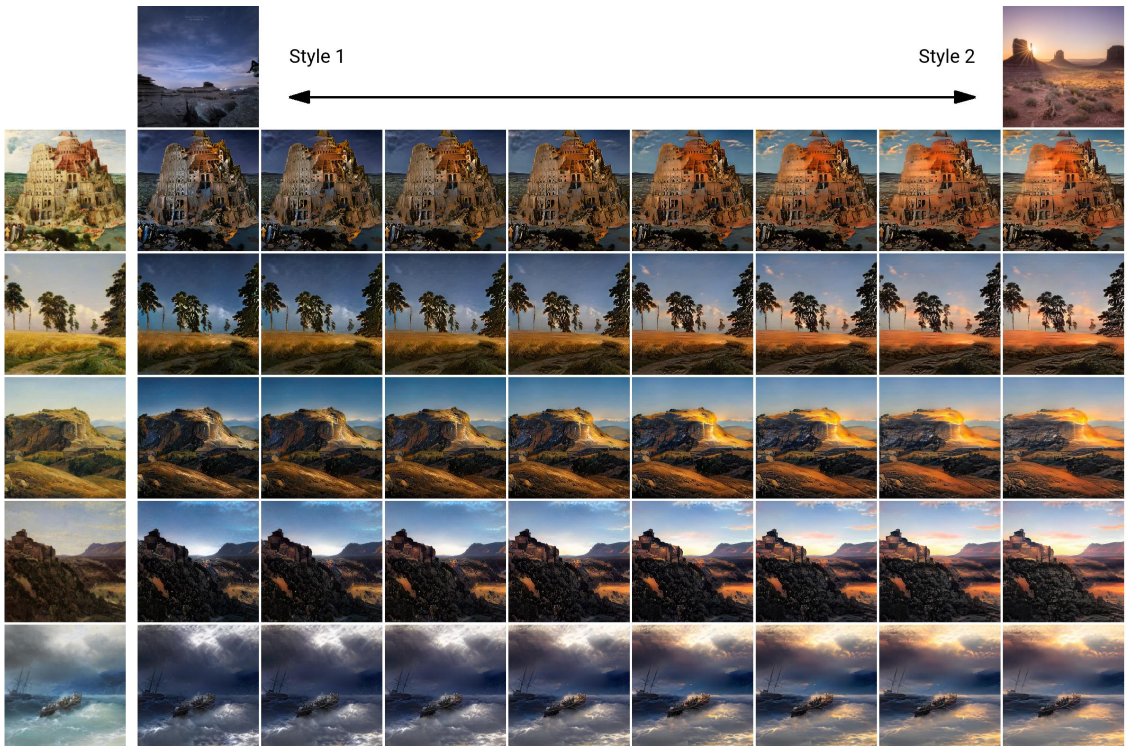


(a) Interpolation between cloudy and clear sky.

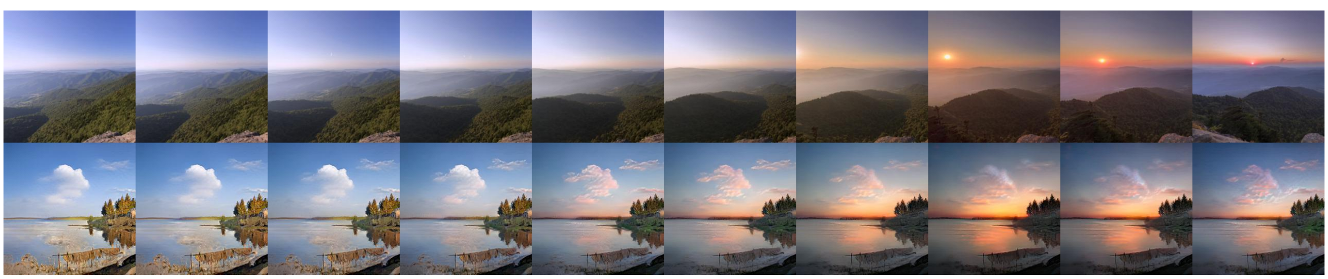


(b) Interpolation between dusk and sunset.

Figure 17: 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.

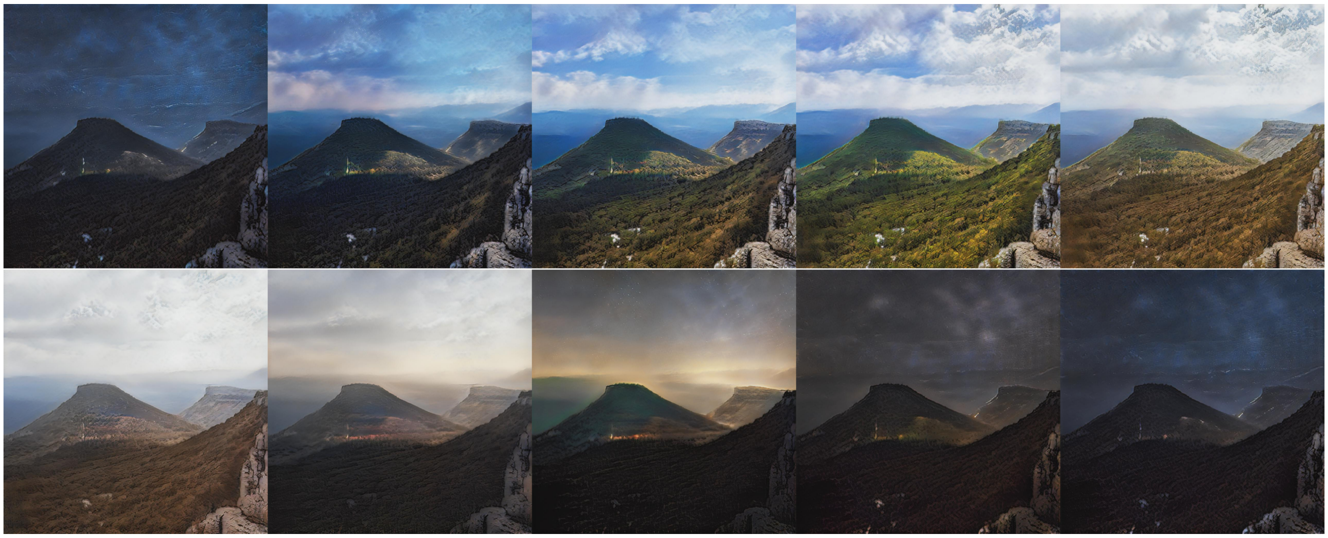
Figure 18: The result of our model, trained on the dataset of landscape photos, applied to the artworks from WikiArt. Our system can handle artworks despite not seeing paintings at train time.

(a)

(b)

(c)

Figure 19: 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.



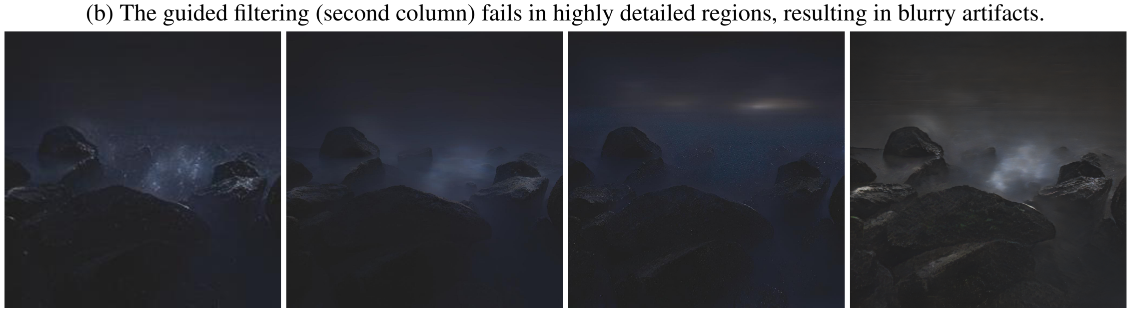
(a)

(b)

Figure 20: 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 21: 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.