Unifying Guided and Unguided Outdoor Image Synthesis

Muhammad Usman Rafique, Yu Zhang, Benjamin Brodie, Nathan Jacobs
University of Kentucky, Lexington, KY
{usman.rafique, y.zhang, benjamin.brodie, nathan.jacobs}@uky.edu

Abstract

Given a source image, our goal is to synthesize novel images of the same scene under different conditions, which could include changes in the time of day, season, or weather conditions. We consider two variants, unguided and guided synthesis, both of which require a way to generate diverse output images that cover the range of possible conditions. For the former task, the layout of the output image should match the source image and the conditions should appear realistic. For the latter task, the conditions should match those of a provided auxiliary guidance image. We address both tasks simultaneously using a probabilistic formulation, with separate distributions for each task, and use an end-to-end training method. We draw samples from these distributions to synthesize plausible images of the source scene. We prepare a new large-scale dataset and propose three benchmark tasks. The dataset, benchmarks, and evaluation code are available at https://mvrl.github.io/un_guided.

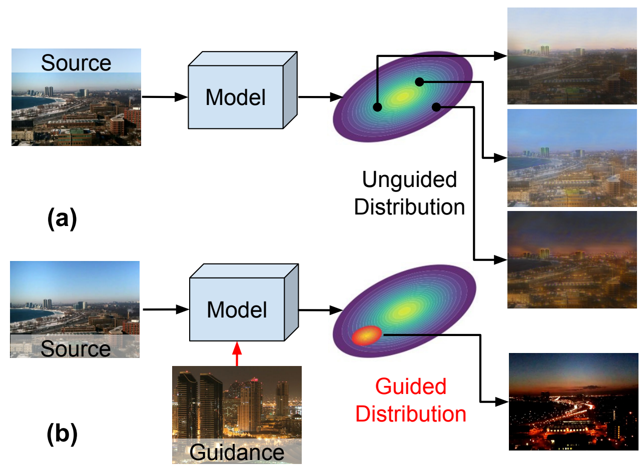


Figure 1. We propose a probabilistic approach for unguided and guided outdoor-image synthesis. Our method generates diverse images using a single forward pass through a neural network.

1. Introduction

We address the task of synthesizing images of a scene, given a single source image, under different conditions. To do this well requires understanding scene geometry, texture, and illumination. For outdoor scenes, synthesis also requires understanding appearance changes due to the time of day, weather conditions, and the seasons. Applications of outdoor image synthesis include providing semantically meaningful tools for image editing and generating training data for autonomous driving systems. We explore two related tasks: unguided and guided synthesis, as shown in Figure 1. In unguided synthesis, the task is to generate new images of a scene from a single source image. For the guided synthesis task, we are given a guidance image and aim to change the appearance of the source image to match that of the guidance image, while preserving the scene contents.

We formulate a probabilistic model with two distributions, unguided and guided. The unguided distribution, which is conditioned on a source image, can be sampled from in order to synthesize images of the source scene under diverse conditions. The guided distribution, which is conditioned on both a source and a guidance image, can be sampled from to synthesize images of the source scene with appearance that matches the guidance image. During training, we jointly optimize for the likelihood of the unguided and guided distributions, as well as minimizing for image reconstruction error. A key benefit of our approach is that we achieve our performance without the extensive annotation effort that is required for competing approaches, such as transient attributes.

Generative adversarial networks (GANs) have gained attention due to the ability to generate photorealistic images [15, 40, 47, 48]. Early GANs focused on unconditional generation, where the goal was to be able to sample random images that were indistinguishable from real images. This setting is limited because there is little user control over output scene layout. Conditional GANs can generate images based on a source image or segmentation mask, making it easy for a user to control the output. Typically, these methods require discrete source and target domains. For example they could be used to convert summer images into winter images. However, appearance changes in outdoor scenes are continuous and it is limiting to divide into discrete domains.

Several approaches, like [13, 14], overcome the limitation of synthesis between discrete domains by conditioning the generation on a rich description of the desired output, which we will call guidance. The guidance can come in the form of an explicit description of the illumination conditions. For example, Karacan *et al*. [14] requires the user to specify 40 transient appearance attributes [20]. Such methods typically require segmentation labels to control the scene layout and a full specification of transient attributes, which can be difficult to specify correctly. Our method only requires sets of images from the same scene for training; there is no need for segmentation labels or transient attribute specification. Thus, we can use unlabelled images from outdoor webcams and use any image as the guidance image.

We introduce a large dataset of outdoor webcam images, with associated benchmarks, to support training and evaluation of this and future methods. We find that our model performs well at both guided and unguided synthesis, outperforming many natural baseline methods without the need for extensive annotations. Our main contributions include:

- We propose a probabilistic framework to synthesize appearances of an outdoor scene that can be used for both guided and unguided synthesis.
- We formulate the latent representation as a probability distribution and show that this distribution is better than using a deterministic latent vector.
- We prepare a training dataset of outdoor images containing short-term and long-term changes along with evaluation benchmarks for guided and unguided synthesis of outdoor images.

2. Related Work

The task of outdoor image synthesis is related to conditional image generation and style transfer approaches.

2.1. Conditional GANs

A conditional GAN, such as Pix2Pix [10], is capable of synthesizing high-quality images in a target domain given a source-domain image. Many methods have been proposed to address problems with the early methods: CycleGAN [47], DualGAN [40], and CUT [28] eliminate the need for aligned image pairs; Pix2PixHD [38] generates higher-resolution outputs; and BicycleGAN [48] can generate more diverse images. These models do not scale to arbitrary styles: a limited numbers of domains are defined, and typically a model learns to convert between two domains only. It is imperative that a sufficient number of images from every domain are available for training. There are various methods to generate images from segmentation masks such as SPADE [29], and SEAN [49]. Domain adaptation methods like [21, 39] learn to transfer images from one domain to another. Existing conditional GANs, like Pix2PixHD [38], are trained to transfer between two narrowly defined domains, such as day-and-night, and a different model is trained for every domain transfer. We train a single model that can generate realistic images under diverse conditions.

2.2. Style Transfer

Earlier neural style transfer methods required optimization for a given style image during inference [5]. (We use the terms "style image" and "guidance image" interchangeably here.) Subsequent methods, like [12, 44], trained a model for every possible style transfer: one model for transfer from style A to style B and vice versa. Recently, several arbitrary style transfer methods [2, 8, 9, 22, 23, 27, 34, 37] have been proposed to generalize to any style without separate training. FST [41] can apply filters from style images to the source image. AdaIN [8] transfers global feature statistics by simply matching the mean and variance between content and style image. Avatar-Net [34] proposes a patch-based feature manipulation module to bridge the gap between the content and style image distribution. WCT [22] uses feature transforms, i.e., whitening and coloring, to match content feature statistics to those of a style image in the deep feature space. WCT^2 [42] uses whitening and color transforms to transfer the style. SANet [27] uses a learnable attention module and replaces the fixed cosine similarity with a flexible similarity kernel. However, these style transfer methods not only require style images for guidance, but also diverse domains and sufficient images from every domain for training.

2.3. Natural Image Synthesis

Existing datasets for natural image synthesis are typically used for either modeling short- or long-term changes. The methods that model long-term changes are typically guided by the transient attributes dataset [20], which provides images with manual annotations of attributes of outdoor scenes, such as cloudy, sunny etc. The method by Karacan *et al*. [14] synthesizes an image based on desired transient attributes. However, this method operates in an explicitly supervised way by requiring scene layout and desired attributes. Transient attributes are hard to decouple, and there is no straightforward way of specifying all 40 attributes. In our case, the desired conditions are specified by a guidance image, and so our method does not require manual annotations of transient attributes or segmentation masks.

There are datasets that include only short-term changes. High-resolution day-time transfer (HiDT) [1], uses a disentanglement approach to swap the style of any two images. HiDT can generate photorealistic images of outdoor scenes, but as the name suggests, it is limited to day-time transfer. Lu *et al*. [43] look at recreating a scene under changes in lighting conditions. Several methods have been proposed for time-lapse generation from a single source image [25, 26]. The method by Cheng *et al*. [3] proposes to generate a short-term sequence that resembles the style of a provided reference time-lapse. To our knowledge, our dataset is the only one that includes both short- and long-term changes.

2.4. Probabilistic Image Synthesis

Probabilistic GANs have been proposed for unconditional image generation. The probabilistic GAN [4] proposes a discriminator that predicts a distribution; they use a standard generator in this approach. BayesianGAN [33] and ProbGAN [6] propose to iteratively learn a distribution over generators that best match the true distribution of the data. We present a probabilistic conditional image generation method by modeling the latent representation with a distribution. Our approach is inspired by Probabilistic U-Net [18], a binary segmentation approach that captures label uncertainty. We propose an image synthesis method that uses two distributions for different tasks of guided and unguided synthesis. Our network architecture, formulation of both tasks, and loss function are different from Probabilistic U-Net.

3. Problem Definition

Consider a statically mounted outdoor camera, recording images of a scene over a long period of time. The recorded images would likely include many types of transient appearance changes. Depending on the scene, some of these would be common, such as the change from day to night, and some might be less common, such as the presence of snow. It is possible to model the distribution over these changes for a single scene by analyzing long-term image archives captured by a single outdoor webcam [11]. We estimate these distributions from an exemplar image, using image collections as training data. Given a single exemplar image, we address the task of modeling the distribution of natural images that appear to be of the same scene captured from the same viewpoint. The goal is to synthesize realistic images, preserving the content of the exemplar, while enabling the sampling of images that reflect the likely transient appearance distribution. We consider two variants of the task, unguided and guided. The latter being useful when some degree of artistic control over the generation process is needed.

For the unguided synthesis task, we are given a large number of images, $\{I_0^s \dots I_N^s\}$, where each I_i^s is a source image from scene s. The goal is to maximize the likelihood of the other images from the same scene $\sum_s \sum_{i=1}^N p(I_i^s|I_0^s)$, where we assume I_0 is the exemplar image and the rest are target images. For the guided synthesis task, we are also given a set of guidance images, $\{\bar{I}_1^s \dots \bar{I}_N^s\}$, which have the same transient appearance attributes as the corresponding target image but a different scene layout and are potentially from a different scene. The goal is to maximize $\sum_s \sum_{i=1}^N q(I_i^s|I_0^s, \bar{I}_i^s)$. In addition, for both tasks we want to be able to sample from the distribution to generate novel images and generalize to novel scenes that aren't present in the training dataset. Please note that we are considering a much harder problem because 1) the input is a single image without any labels of the scene content or geometry, 2) the target *domain* is diverse and it includes all appearances unlike existing works, such as [48], that restrict to a single target domain such as winter or night, and 3) we train a single model that captures the all visual conditions, in contrast to methods that train a separate model for every target domain.

4. Approach

The high-level architecture of our proposed approach is shown in Figure 2. Inference from our trained model is as follows. Output images are generated by a decoder network which takes as input a feature map describing scene layout and a sample from the n-dimensional latent style space. We define two distributions over style: the unguided, p, which models likely appearances for a given source image, and the guided, q, which is a narrower distribution that is also conditioned on a guidance image. We model the distributions as independent multivariate Gaussian distributions having n dimensions. For unguided synthesis, we sample from the unguided distribution, p, based on the source image, and pass these through the decoder, as in Figure 1 (a). For guided synthesis, we sample from the guided distribution, q, as in Figure 1 (b). For both tasks, we can draw multiple samples to make diverse predictions.

4.1. Network Architecture

Our architecture consists of several sub-networks: a style encoder, a content encoder, two distribution parameter regressors, and a decoder. The style encoder is used to extract a style vector from the source and target images. The content encoder, which is the first half of a ResNet-based U-Net [32], extracts a feature map that represents the layout of the source image. It is also extracts an additional content vector that can capture high-level scene content, such as whether the scene includes mountains or a beach. The two distribution parameter regressors are small multi-layer perceptrons (MLPs) that predict the parameters of the style distributions. Each has two heads with n outputs, one for the means and the other for the variances. In the decoder, we use adaptive instance normalization (AdaIN) to combine the sampled style vector and source content feature map [1, 8, 24]. Please see the supplemental material for details of network architectures.

4.2. Training

Our training overview is presented in Figure 2. During training, we sample a source and target image from a scene. The target image is flipped horizontally and treated as the guidance image. We pass the style encoding of the horizontally flipped target image through an MLP to predict the parameters of the guided distribution q: mean μ_q and variance σ_q^2. We apply the horizontal flip to the target image to

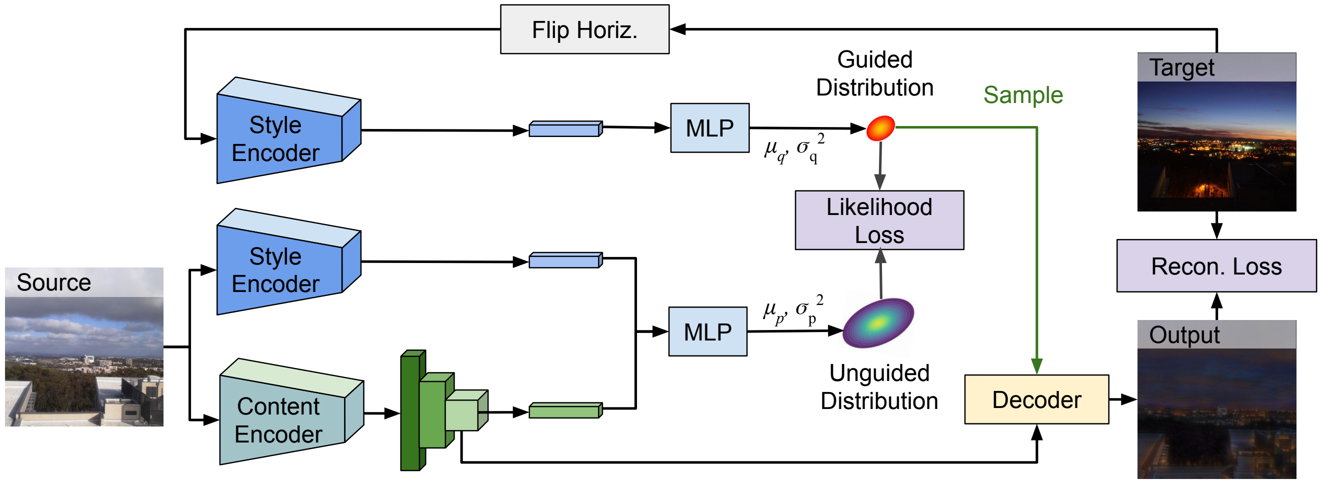


Figure 2. The proposed probabilistic visual appearance network. Two style encoders, with shared weights, extract style vectors from the source and target images. A content encoder extracts a content feature map and content vector. Two MLPs predict distribution parameters given the style vectors, and a content vector for the unguided distribution. The decoder synthesizes an output image using the content feature map and a style vector sampled from the guided distribution.

limit information leakage and encourage generalization to the cases where the guidance image is from a different scene. Another MLP predicts the unguided distribution parameters μ_p and σ_p^2 based on the the style and content of the source image. We use content of the source image because possible appearances of a scene are correlated with the scene content. For example, we are more likely to observe snow and fog in a scene if there are mountains in it. During training, we draw a sample from the guided distribution and a decoder combines this with source content features to synthesize the final image. A key difference between our approach and disentanglement based methods that swap style and content, such as [1] and [30], is that in our case, the content might not be visible in the target image for conditions such as night and fog. Therefore, as shown in Figure 2, we only extract the style from the target image.

We enforce the constraint that every sample from the guided distribution (each example representing an appearance condition) could reasonably be a sample from the unguided distribution. While training, we draw samples from the guided distribution, which are used to synthesize an image which should match the target image. The network predicts an unguided distribution based on the source and a guided distribution based on the target image. We jointly optimize for unguided distribution, guided distribution and the output image. For an unguided distribution p, guided distribution q, output image $\hat{I}$, and target image I, the complete loss function is:

$$L = \lambda_p L_p(p) - \lambda_{pq} L_l(p, q) + L_R(\hat{I}, I).$$

Here L_p is the conditioning loss for the unguided distribution, L_l is a likelihood estimation between unguided and guided distributions, and L_R is the reconstruction loss between the output image and target image. We set the weights $\lambda_p = 0.2$ and $\lambda_{pq} = 0.2$. The likelihood estimation between p and q is given by:

$$L_l(p, q) = \mathcal{L}(p, q) + \lambda_e h(p) + C$$

where $h(p)$ is the entropy of the unguided distribution, and $\mathcal{L}$ is the log-likelihood. We set $\lambda_e = n$, where n is the dimension of p, and set $C = \frac{n}{2}\ln(2\pi)$. The likelihood function is

$$L_l(p, q) = -\frac{1}{2\sigma_p^2} \sum_{i=1}^{n} (s_q - \mu_p)^2,$$

where μ_p and σ_p^2 are the mean and variance of p, and s_q is a sample from the guided distribution ($s_q \sim q$). Adding the entropy regularization discourages the network from predicting only distributions with small variance. The problem of small variance has been discussed in InfoVAE [46] as well.

At inference time, we want to generate diverse samples from the unguided distribution. A common approach for this is to impose a unit Gaussian prior over the unguided distributions, as in variational auto encoders (VAE) [17]. We relax this constraint and allow the unguided distribution of individual images to vary, providing greater appearance variations. During training, we perform this regularization at the batch-level by introducing a regularization loss L_p. We model the batch-wide collection of B predicted unguided distributions as the Gaussian mixture $\frac{1}{B}\sum_{i=1}^{B} p_i$. We then collapse the mixture down to a single multivariate Gaussian using the distribution $\mathcal{N}(\mu_M, \Sigma_M)$ with parameters

$$\mu_M = \frac{1}{B}\sum_{i=1}^{B} \mu_{p_i}, \Sigma_M = \frac{1}{B}\sum_{i=1}^{B} \sigma_{p_i}^2 + \mu_{p_i}\mu_{p_i}^T - \mu_M \mu_M^T,$$

where μ_{p_i} and $\sigma_{p_i}^2$ are mean and variance of the unguided distributions. Note that $\mathcal{N}(\mu_M, \Sigma_M)$ is the multivariate Gaussian that minimizes the KL divergence to the Gaussian mixture $\frac{1}{B}\sum_{i=1}^{B} p_i$. We then set the regularization loss L_p to be the KL divergence between the unit Gaussian and the collapsed mixture of Gaussians $\mathcal{N}(\mu_M, \sigma_M^2)$:

$$L_p(p) = D_{KL}(\mathcal{N}(\mathbf{0}, \mathbf{1}), \mathcal{N}(\mu_M, \sigma_M^2)).$$

The reconstruction loss, L_R, is given by:

$$L_R(\hat{I}, I) = L_1(\hat{I}, I) + L_F(\hat{I}, I) + 5 \cdot L_T(\hat{I}, I) + L_G(\hat{I}, I) + L_E(\hat{I}, I),$$

where L_F is the feature loss [12] using a pretrained VGG network [36], L_E is the edge loss, and L_G is the GAN loss from a multi-scale discriminator [38]. $L_T = |T(\hat{I}) - T(I)|$, is the difference of transient attributes using a pretrained network T that regresses transient attributes of an image.

5. A New Dataset for Natural Image Synthesis

We introduce a new derivative dataset of outdoor images that contains short- and long-term appearance changes. It contains images from 188 scenes: 94 time-lapse videos from the TLVDB dataset [35] that have short-term changes and 94 cameras from transient attributes dataset [20] that have long-term changes. While we collect images from these datasets, we manually separate out source images, define a training regime, and make evaluation benchmarks. Taking images from existing datasets is a common practice and images in [20] are also taken from other sources such as AMOS [11]. We randomly selected 150 scenes for training, 19 for validation, and 19 for testing. We manually select clear, daytime images to be used as source images. In total, there are 5864 source and 17 368 target images.

We use this dataset to define three benchmarks for guided and unguided synthesis, defined below. To our knowledge, this the only large-scale dataset that contains 1) short-term and long-term appearance changes, 2) manually filtered daytime source images, 3) aligned images suitable for training and evaluation, and 4) image synthesis benchmarks for both guided and unguided synthesis. Our dataset is available at `https://mvrl.github.io/un_guided`.

5.1. Unguided Synthesis Benchmark

We defined a benchmark to assess how well a method is able to synthesize diverse, realistic samples from a single image. To evaluate this task, we need diverse examples for any given scene. As with all tasks, we select clean daylight images as the source images. In the test set, we have 595 source images and 1140 target images. For quantitative evaluation, we use standard point set distance measures and Fréchet Inception Distance (FID) which compares quality of generated images with real images [7].

To compute point set metrics, we use every source image to generate k unguided images from the unguided distribution where k is the number of real target images for that scene. We use Hausdorff distance and Chamfer distance as measures of distances between the set of real target images S_I and the set of output images $S_{\hat{I}}$ of that scene. Hausdorff distance is given as:

$$d_H(S_I, S_{\hat{I}}) = \max\left[\max_{e\in S_I} \Delta_m(e, S_{\hat{I}}), \max_{e\in S_{\hat{I}}} \Delta_m(e, S_I)\right],$$

$$\Delta_m(x, S) = \min_{y\in S} \Delta(x, y)$$

for any distance measure Δ; we use L_1 distance as Δ. We also use Chamfer distance, d_C, for evaluation:

$$d_C(S_I, S_{\hat{I}}) = \frac{1}{|S|}\sum_{e\in S_I} \Delta_m(e, S_{\hat{I}}).$$

While the Hausdorff distance measures the maximum distance between any two points on the closest matching pairs, the Chamfer distance measures the average distance of the closest pairs. To compute FID [7], we randomly select source images to synthesize the same number images as the true target images (1140). We then compute FID between the output images from all scenes and all target images. To establish lower bounds for these metrics, we split the target images into two partitions and compute the metrics between the partitions. We refer to this as the *Oracle Test Set*.

5.2. Same-Scene Guided Synthesis Benchmark

In this benchmark, the guidance image is from the same scene as the source image. This is intended to serve as an easier case for the guided synthesis task. To create this, we flip the target image horizontally and treat it as the guidance image. Since we typically have more target images from every scene, we make a fixed benchmark by randomly selecting a source image (from the same scene) for every target image. We have 1140 examples in this benchmark. Since source and target images are from the same scene, we use standard image matching metrics including L_1 error, peak signal to noise ratio (PSNR), and structural similarity (SSIM). We also include perceptual similarity (LPIPS) [45] (using a pretrained AlexNet [19]), that has been shown to closely match human judgement.

5.3. Cross-Scene Guided Synthesis Benchmark

This benchmark, also having 1140 examples, estimates generalization of methods; in this task the guidance image is from a different scene. To make this benchmark, we use the following procedure to select a guidance image that has similar appearance as the target image. We train a model on the transient attributes [20] dataset which gets only 1.3% mean squared error on the held-out validation set for attributes like

Method	Same-Scene Guided				Cross-Scene Guided				Unguided Synthesis		
	$L_1 \downarrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	$L_1 \downarrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	Hausdorff $\downarrow$	Chamfer $\downarrow$	FID $\downarrow$
Oracle Test Set	-	-	-	-	-	-	-	-	0.1446	0.0609	26.4512
BicycleGAN [48]	0.1216	0.4668	15.8983	0.5249	0.1376	0.5937	14.6051	0.4217	0.2987	**0.1145**	121.6977
SANet [27]	0.1209	0.4175	16.0552	0.5057	**0.1218**	0.4216	**15.7852**	0.4949	-	-	-
Ours w/o guided distribution	0.2141	0.3984	12.1147	0.4821	0.2139	0.3982	14.0321	0.5084	0.5728	0.1539	219.5103
Ours w/o prior loss	**0.1124**	0.3392	**16.9831**	**0.5889**	0.1569	0.3594	14.7710	0.5354	0.2269	0.1147	84.8741
Ours w/o likelihood loss	0.1947	0.3892	13.3184	0.5064	0.1936	0.3926	13.3602	0.5083	0.2995	0.1791	91.3697
Ours full	0.1197	**0.3367**	16.4931	0.5858	0.1495	**0.3490**	15.0803	**0.5566**	**0.2259**	0.1231	**79.8313**

Table 1. Test set results of the three benchmarks.

Method	Input	$L_1 \downarrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$
Cheng [3]	Image + Segmentation	0.1462	**0.2842**	14.7220	0.4858
Ours	Image	**0.1102**	0.2947	**16.9018**	**0.6822**

Table 2. Results of time-lapse generation.

cloudy, snow etc. For every target image in the test set, we randomly select a source image from the scene. To select the guidance image from a different scene, we use our network trained on transient attributes to find the most similar image from other scenes, in terms of transient attributes.

6. Evaluation

Please see the supplemental material for network details and additional visualizations.

6.1. Baseline Methods

We compare our method with three similar methods. We compare with BicycleGAN [48] that was originally designed to generate diverse samples from a single source. We also compare with a recent arbitrary style-transfer method, SANet [27], and a time-lapse generation method [3].

6.2. Implementation Details

We use PyTorch [31] to implement our model. Following existing methods, we train all methods on 256×256 images. We show qualitative results on 512×512 images from our model to demonstrate that we can generate realistic high-resolution images. We optimize using the Adam optimizer [16] ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with a learning rate of 1.2×10^{-4}, L_2 regularization of 1×10^{-5}, and batch size of 24. All models are trained for 50 epochs and the learning rate is reduced by a factor of 0.9 after every 5 epochs. During training, we randomly crop and flip images. We set the latent dimension $n = 32$.

6.3. Quantitative Results

We show the results of all three benchmarks in Table 1. Our method performs better on the same-scene guided synthesis benchmark than SANet and BicycleGAN. For cross-scene guided synthesis, our method gets the best LPIPS and SSIM while SANet gets better L_1 and PSNR. For unguided synthesis, our method performs significantly better than BicycleGAN on Hausdorff distance and FID metrics, while getting a comparable Chamfer distance. SANet, a style transfer method, cannot be used for unguided synthesis without a style image.

6.4. Qualitative Results

We show results of unguided synthesis in Figure 4. The source image is shown on the left (a) and several images sampled from the unguided distribution are shown in (b)-(e). We can see that our method can generate realistic outputs under diverse lighting and weather conditions. Results of cross-scene guided synthesis are shown in Figure 3. Since we model the style using a guided distribution, we can generate multiple samples from this during test set. For every example, we show two synthesized outputs in Figure 3 (d)-(e). We can see that there are some variations in these images, like sky color and minor lighting variations.

6.5. Time-Lapse Generation

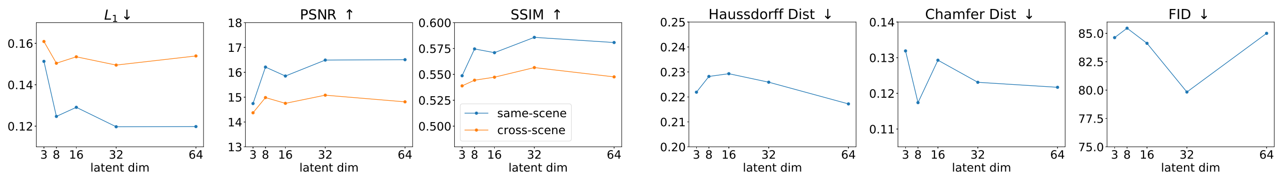
We compare our method for time-lapse generation with the state-of-the-art method by Cheng *et al.* [3]. Both models are trained and evaluated on image size 512×512. We show quantitative results only on the time-lapse videos in the test set, comprising of 8 sequences and 800 total examples. These time-lapses are from the TLVDB dataset [35] which is the test set used by Cheng *et al.* [3]. For this evaluation, we select a source image from every sequence and use the horizontally flipped version of other frames as the guidance. This allows us to compare the output images with the reference images. We show results in Table 2. Please note that Cheng *et al.* [3] require the true segmentation labels of source and guidance images during training and inference. Our method does not need segmentation for training or inference. We can see from Table 2 that our method performs better than [3] on all metrics except LPIPS. We show qualitative results in Figure 6; it can be seen that our method generates more realistic outputs with natural colors of the sky.

(a) Source (b) Guidance (c) Target (d) Synthesis 1 (e) Synthesis 2

Figure 3. Qualitative results: cross-scene guided synthesis on the test set. We show two different synthesized images, (d) and (e), which are sampled from the guided distribution q for the given guidance image.

(a) Source (b) Synthesis1 (c) Synthesis2 (d) Synthesis3 (e) Synthesis4

Figure 4. Qualitative results: unguided synthesis. Note that these results are from the unseen test set.



(a) Guided Synthesis (b) Unguided Synthesis

Figure 5. Assessing the impact of changing the dimensionality, n, of the unguided/guided distributions.

Figure 6. Time-lapse results based on a reference sequence (top row). Our method works only on source images while Cheng *et al*. [3] also requires segmentation masks of source image and the reference scene.

6.6. Ablation and Analysis

We provide ablation of the key choices in Table 1. We show the significance of probabilistic modeling of the guided distribution. If we extract a deterministic vector (ours w/o guided distribution), the method performs significantly worse on all metrics than our full method. We analyse our proposed modeling of unguided distributions as mixture of Gaussians: we prepare a baseline (ours w/o prior loss) in which we use the standard KL divergence loss. This baseline performs well on the same-scene synthesis and gets slightly worse results on other benchmarks. Finally, we analyze our proposed likelihood loss by developing a baseline (ours w/o likelihood loss) that uses KL divergence between unguided and guided distributions. This baseline, which closely resembles probabilistic U-Net, performs worse on all benchmarks.

We analyze the size of the latent vector n as shown in Figure 5. We see that even as the latent vector size increases, the performance of our method remains stable. We hypothesize that this is because of two factors. First, our probabilistic formulation encourages generalization during training by drawing a sample from the guided distribution and not by extracting the exact vector, as shown in the ablation of our method vs. a method that does not use probability distribution (ours w/o guided distribution). Second, in our network design, we extract a style encoding using global average pooling and then feed this to an MLP which removes spatial information.

7. Conclusion

We introduced a novel approach for synthesizing natural appearance variations from a single source image, simultaneously addressing the tasks of unguided and guided image synthesis. We formulated this as a probabilistic model with an end-to-end training strategy. We also introduced a large-scale dataset for training and three evaluation benchmarks. We found that our method is able to synthesize diverse and realistic images, improving upon several baseline methods. We also significantly outperform the existing state of the art for time-lapse image generation. On the other tasks we perform at or near the state of the art. This evaluation highlights the value of our dataset and hope that it will spur further research in this field.

References

[1] Ivan Anokhin, Pavel Solovev, Denis Korzhenkov, Alexey Kharlamov, Taras Khakhulin, Aleksei Silvestrov, Sergey Nikolenko, Victor Lempitsky, and Gleb Sterkin. High-resolution daytime translation without domain labels. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[2] Tian Qi Chen and Mark Schmidt. Fast patch-based style transfer of arbitrary style. *arXiv preprint arXiv:1612.04337*, 2016.

[3] Chia-Chi Cheng, Hung-Yu Chen, and Wei-Chen Chiu. Time flies: Animating a still image with time-lapse video as reference. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[4] Hamid Eghbal-zadeh and Gerhard Widmer. Probabilistic generative adversarial networks. *arXiv preprint arXiv:1708.01886*, 2017.

[5] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[6] Hao He, Hao Wang, Guang-He Lee, and Yonglong Tian. ProbGAN: Towards probabilistic gan with theoretical guarantees. In *International Conference on Learning Representations (ICLR)*, 2019.

[7] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[8] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[9] X. Huang, Ming-Yu Liu, Serge J. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. *arXiv preprint arXiv:1804.04732*, 2018.

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

[11] Nathan Jacobs, Nathaniel Roman, and Robert Pless. Consistent temporal variations in many outdoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[12] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, 2016.

[13] Levent Karacan, Zeynep Akata, Aykut Erdem, and Erkut Erdem. Learning to generate images of outdoor scenes from attributes and semantic layouts. *arXiv preprint arXiv:1612.00215*, 2016.

[14] Levent Karacan, Zeynep Akata, Aykut Erdem, and Erkut Erdem. Manipulating attributes of natural scenes via hallucination. *arXiv preprint arXiv:1808.07413*, 2018.

[15] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

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

[17] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[18] Simon Kohl, Bernardino Romera-Paredes, Clemens Meyer, Jeffrey De Fauw, Joseph R Ledsam, Klaus Maier-Hein, SM Ali Eslami, Danilo Jimenez Rezende, and Olaf Ronneberger. A probabilistic U-Net for segmentation of ambiguous images. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 2017.

[20] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Qian, and James Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics (SIGGRAPH)*, 2014.

[21] Rui Li, Wenming Cao, Qianfen Jiao, Si Wu, and Hau-San Wong. Simplified unsupervised image translation for semantic segmentation adaptation. *Pattern Recognition*, 2020.

[22] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[23] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[24] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[25] E. Logacheva, R. Suvorov, Oleg Khomenko, A. Mashikhin, and Victor Lempitsky. Deeplandscape: Adversarial modeling of landscape videos. In *European Conference on Computer Vision (ECCV)*, 2020.

[26] Seonghyeon Nam, Chongyang Ma, M. Chai, William Brendel, N. Xu, and S. Kim. End-to-end time-lapse video synthesis from a single outdoor image. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[27] Dae Young Park and Kwang Hee Lee. Arbitrary style transfer with style-attentional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[28] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision (ECCV)*, 2020.

[29] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[30] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A. Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming

Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention*. Springer, 2015.

[33] Yunus Saatci and Andrew G Wilson. Bayesian GAN. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

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

[35] Yichang Shih, Sylvain Paris, Frédo Durand, and William T Freeman. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Transactions on Graphics (TOG)*, 2013.

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[37] Jan Svoboda, Asha Anoosheh, Christian Osendorfer, and Jonathan Masci. Two-stage peer-regularized feature recombination for arbitrary image style transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[38] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional GANs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[39] Yanchao Yang and Stefano Soatto. FDA: Fourier domain adaptation for semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[40] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[41] Jonghwa Yim, Jisung Yoo, Won-joon Do, Beomsu Kim, and Jihwan Choe. Filter style transfer between photos. In *European Conference on Computer Vision (ECCV)*, 2020.

[42] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. Photorealistic style transfer via wavelet transforms. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[43] Ye Yu, Abhimitra Meka, M. Elgharib, H. Seidel, Christian, Theobalt, and W. Smith. Self-supervised outdoor scene relighting. In *European Conference on Computer Vision (ECCV)*, 2020.

[44] Hang Zhang and Kristin Dana. Multi-style generative network for real-time transfer. *arXiv preprint arXiv:1703.06953*, 2017.

[45] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[46] Shengjia Zhao, Jiaming Song, and Stefano Ermon. InfoVAE: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017.

[47] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[48] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[49] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.