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.

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. WCT2 [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 dis-
riminator that predicts a distribution; they use a standard
generator in this approach. BayesianGAN [33] and Prob-
GAN [6] propose to iteratively learn a distribution over gen-
erators 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 bi-
nary segmentation approach that captures label uncertainty.
We propose an image synthesis method that uses two distri-
butions 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 appear-
ce 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 distri-
butions 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 view-
point. 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 distribu-
tion. 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_1^s, \ldots, I_N^s\}, where each I_i^s is a source
image from scene s. The goal is to maximize the likelihood
of \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, \{\tilde{I}_1^s, \ldots, \tilde{I}_N^s\}, which
have the same transient appearance attributes as the cor-
responding 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 | \tilde{I}_0^s, \tilde{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 im-
age 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 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
coder, a content encoder, two distribution parameter regres-
sores, and a decoder. The style encoder is used to extract
a style vector from the source and target images. The content
coder, which is the first half of a ResNet-based U-Net [32],
evaluates a feature map that represents the layout of the source
image. It 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 parame-
ters 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 hori-
zontally flipped target image through an MLP to predict the
parameters of the guided distribution q: mean \mu_q and vari-
ance \sigma_q^2. We apply the horizontal flip to the target image to
Another MLP predicts the unguided distribution parameters \( \mu_p \) and \( \sigma^2_p \) based on 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 \( I \), and target image \( \hat{I} \), 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:

\[
L_l(p, q) = \mathcal{L}(p, q) + \lambda_c h(p) + C
\]

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

\[
L_l(p, q) = -\frac{1}{2\sigma^2_p} \sum_{i=1}^{n} (s_q - \mu_p)^2,
\]

where \( \mu_p \) and \( \sigma^2_p \) 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

\[
\mu_M = \frac{1}{B} \sum_{i=1}^{B} \mu_{p_i}, \quad \Sigma_M = \frac{1}{B} \sum_{i=1}^{B} \sigma^2_{p_i} + \mu_{p_i} \mu_{p_i}^T - \mu_M \mu_M^T,
\]
where \( \mu_{p_u} \) and \( \sigma^2_{p_u} \) 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 \( \frac{1}{B} \sum_{i=1}^B p_u \). We then set the regularization loss \( L_p \) to be the KL divergence between the unit Gaussian and the collapsed mixture of Gaussians \( \mathcal{N}(\mu_M, \sigma^2_M) \):

\[
L_p(p) = D_{KL}(\mathcal{N}(\mathbf{0}, 1), \mathcal{N}(\mu_M, \sigma^2_M)).
\]

The reconstruction loss, \( L_R \), is given by:

\[
L_R(\hat{I}, I) = L_1(\hat{I}, I) + L_F(\hat{I}, I) + 5 \cdot L_T(\hat{I}, I) + L_C(\hat{I}, I) + L_E(\hat{I}, I),
\]

where \( L_F \) is the feature loss [12] using a pretrained VGG network [36], \( L_E \) is the edge loss, and \( L_C \) 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 17368 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 \( \hat{S}_I \) of that scene. Hausdorff distance is given as:

\[
d_H(S_I, \hat{S}_I) = \max \left[ \max_{e \in \hat{S}_I} \Delta_m(e, S_I), \max_{e \in S_I} \Delta_m(e, \hat{S}_I) \right],
\]

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

for any distance measure \( \Delta \); we use \( L_1 \) distance as \( \Delta \). We also use Chamfer distance, \( d_C \), for evaluation:

\[
d_C(S_I, \hat{S}_I) = \frac{1}{|S_I|} \sum_{e \in S_I} \Delta_m(e, \hat{S}_I).
\]

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
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 $256 \times 256$ images. We show qualitative results on $512 \times 512$ 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 $1.2 \times 10^{-4}$, $L_2$ regularization of $1 \times 10^{-5}$, 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 $512 \times 512$. 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.
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.

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


