GP-GAN: Towards Realistic High-Resolution Image Blending

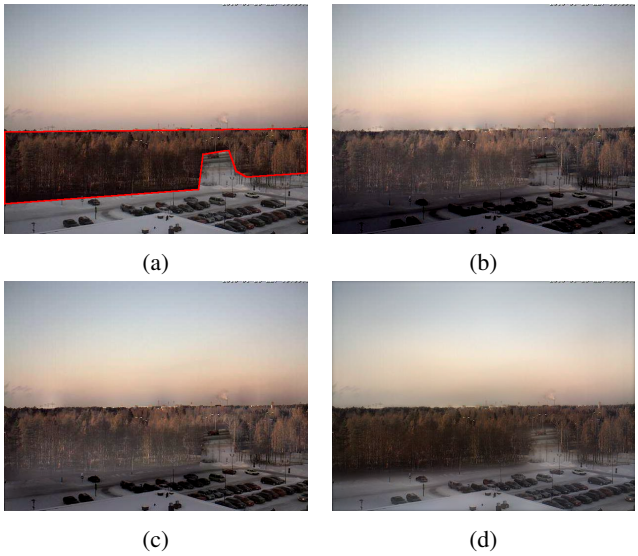
Huikai Wu[1*] Shuai Zheng[2†] Junge Zhang[1§] Kaiqi Huang[1§]

[1]NLPR, Institute of Automation, Chinese Academy of Sciences
[2]University of Oxford

*huikai.wu@cripac.ia.ac.cn †szheng@robots.ox.ac.uk
§{jgzhang, kaiqi.huang}@nlpr.ia.ac.cn

Abstract

Recent advances in generative adversarial networks (GANs) have shown promising potentials in conditional image generation. However, how to generate high-resolution images remains an open problem. In this paper, we aim at generating high-resolution well-blended images given composited copy-and-paste ones, i.e. *realistic high-resolution image blending. To achieve this goal, we propose Gaussian-Poisson GAN (GP-GAN), a framework that combines the strengths of classical gradient-based approaches and GANs, which is the first work that explores the capability of GANs in high-resolution image blending task to the best of our knowledge. Particularly, we propose Gaussian-Poisson Equation to formulate the high-resolution image blending problem, which is a joint optimisation constrained by the gradient and colour information. Gradient filters can obtain gradient information. For generating the colour information, we propose Blending GAN to learn the mapping between the composited image and the well-blended one. Compared to the alternative methods, our approach can deliver high-resolution, realistic images with fewer bleedings and unpleasant artefacts. Experiments confirm that our approach achieves the state-of-the-art performance on Transient Attributes dataset. A user study on Amazon Mechanical Turk finds that majority of workers are in favour of the proposed approach.*



(a) (b) (c) (d)

Figure 1: **Qualitative illustration of high-resolution image blending.** a) shows the composited copy-and-paste image where the inserted object is circled out by red lines. Users usually expect image blending algorithms to make this image more natural. b) represents the result based on Modified Poisson image editing [32]. c) indicates the result from Multi-splines approach. d) is the result of our method GP-GAN. Our approach produces better quality images than that from the alternatives in terms of illumination, spatial, and color consistencies. Best viewed in color.

1. Introduction

A human receives and expresses information mostly in the visual forms, such as drawing paintings, making sculptures, and taking photos. Technologies such as the Internet make it so much easier to obtain a photo than before. However, visual communication still requires talents. For example, the photos edited by expert Photoshop users remain far better than the ones from the newcomers. We would like to address the problem of high-resolution image blending, which plays a key role in many applications, from modifying the visual communication content to automatic photo editing. Particularly, we aim at generating high-resolution and realistic images given the composited copy-and-paste ones. The solution developed from this paper would help to bridge the talented gap between the expert Photoshop users and the beginners. This problem is challenging because most users would often have high expectation on the

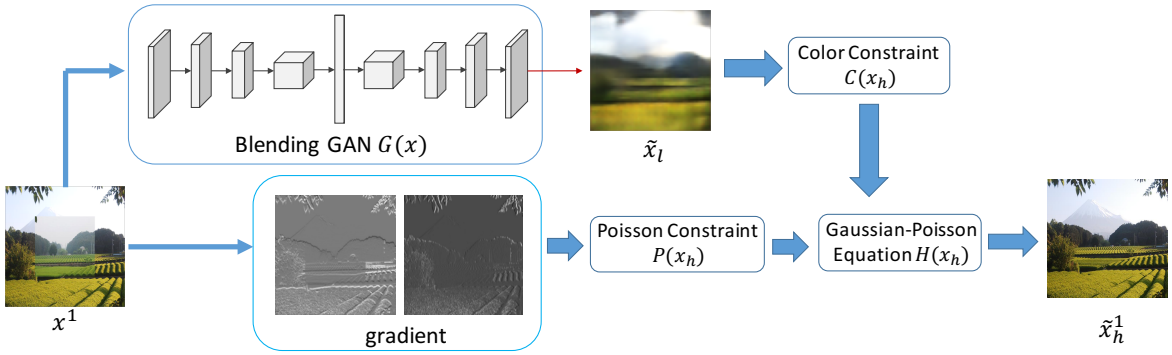


Figure 2: **Framework Overview for GP-GAN.** Given a composited image x, we first generate a low-resolution realistic image $\tilde{x}_l$ using Blending GAN $G(x)$ with x^1 as input where x^1 is the coarsest scale in the Laplacian pyramid of x. Then we optimize the Gaussian-Poison Equation constrained by $C(x_h)$ and $P(x_h)$ using the closed from solution to generate x_h^1 which contains many details like textures and edges. We then set $\tilde{x}_l$ to up-sampled x_h^1 and optimize the Gaussian-Poisson Equation at a finer scale in the pyramid of x. Best viewed in color.

quality of the generated images. As shown in Figure 1, users insert an object in the background image (See Figure 1a, the inserted object is circled out by red lines) and want to make this composited copy-and-paste image more realistic. If the algorithm produces result like (b) or (c), users will give up to use the solution after their first few tries.

Recent researches have achieved significant progress in unsupervised learning with the rise of generative adversarial networks (GANs) [11, 7, 26, 2]. GANs provide a framework for estimating the generative models via simultaneously training a generative model and a discriminative model in a zero-sum game. Natural images can be generated by the generator of GANs. Mirza *et al*. [23, 15] generalize the idea to a condition setting, so that it can be useful for broader applications like image inpainting [24]. Although the existing methods could generate realistic images, they could only produce relative low-resolution and blurred images, which motivates us to come up with a solution to high-resolution conditional image generation by taking advantage of conditional GANs. Particularly, we aim at generating high-resolution well-blended images given composited ones, *i.e*. image blending.

Before the rise of GANs, there are several traditional solutions in the field of image blending, which enables smoothing transition and reduces the colour/illumination differences between images for hiding the artefacts. Gradient domain image blending approach [25] is one of the image blending solutions. In this type of approach, the new gradient vector field is produced based on the source image gradients, while the composite image would be recovered from this gradient vector field by addressing a Poisson equation. This approach allows for adjusting the colour differences caused by the illumination differences. Although these traditional solutions could generate high-resolution images, the images tend to be unrealistic with various kinds of artefacts. We would like to address this by exploring the potentials of GANs in image blending tasks since GANs could generate realistic images. To the best of our knowledge, it is the first work that explores the capability of GANs in high-resolution image blending task.

Our work is related to the recent works [34, 37]. Our work is complementary to Tsai *et al*. [34], which utilises visual semantics to guide compositing images. Yang *et al*. [37] is unsuitable for image blending when the source image and the destination image are significant different.

We develop a framework GP-GAN that combines the strength of GANs and the gradient-based image blending method for conditional image generation, as shown in Figure 2. Our framework consists of two phases. On phase one, the low-resolution realistic images were generated based on our proposed Blending GAN. On phase two, we solve the proposed Gaussian-Poisson Equation based on the gradient vector field fashioned by Laplacian pyramid. This framework allows us to achieve high-resolution and realistic images as shown in Figure 1. Our main contributions are four folds and summarised as follows:

- We develop a high-resolution conditional image generation framework GP-GAN that takes advantages of both GANs and gradient-based image blending methods, which is the first work that explores the capability of GANs in high-resolution image blending task to the best of our knowledge.
- We propose a network called Blending GAN for generating low-resolution realistic images.
- We propose the Gaussian-Poisson Equation for combine gradient information and colour information.
- We also conduct a systematic evolution of the proposed approach, based on both benchmark experiments and user studies on Amazon Mechanic Turk.

2. Related Work

We briefly review the relevant works from the classical image blending approach to generative adversarial networks and conditional generative adversarial networks. We also discuss the difference between our work and the others.

2.1. Image Blending

The goal of classical image blending approaches is to improve the colour consistency between the source images so that we can generate composited images with fewer artefacts. One way [12] is to apply dense image matching approach so that only the corresponding pixels are copied and pasted. However, this method would not work when there are significant differences between the source images. The other way is to make the transition as smooth as possible so that we can hide the artefacts in the composited images. Alpha blending [35] is the simplest and fastest method, but it blurs the fine details when there are some registration errors or fast moving objects between the source images. Burt and Adelson [4] present a fixing solution so-called multi-band blending algorithm. Alternatively, gradient-based approaches [21, 35, 9, 1, 16, 18, 30] also address this problem by adjusting the differences in color and illumination for the composited image globally. Our work is different from these gradient-based approaches in that we introduce GANs to generate a low-resolution realistic image as the colour constraint. Thus our generated images are more natural.

2.2. Generative Adversarial Networks

Generative Adversarial Networks (GANs) [11] is first introduced to address the problem of generating the realistic images. The main idea in GANs is to have a continuing zero-sum game between learning a generator and a discriminator. The generator tries to produce more realistic image samples from random noise, while the discriminator aims to distinguish generated images from the real ones. Although the original method works for creating digital images from MNIST dataset, some generated images are noisy and incomprehensible. Denton *et al*. [7] improve the quality of generated images by generalising GANs with a Laplacian pyramid implementation, but it does not work well for the images containing objects looking wobbly. Gregor *et al*. [13] and Dosovitskiy *et al*. [8] achieve successes in generating natural images, however, they do not leverage the generators for supervised learning. Radfor *et al*. [26] achieve further improvement with a deeper convolutional network architecture, while Zhang *et al*. [39] stack two generators to progressively render more realistic images. InfoGAN [6] learns a more interpretable latent representation. Salimans *et al*. [29] reveal several tricks in training GANs. Arjovsky *et al*. [2] introduce an alternative training method Wasserstein GAN, which relaxes the GAN training requirement for balancing the discriminator and generator. However, existing GANs still do not work well for the image editing applications in that the generated results are not in high-resolution and realistic quality yet.

2.3. Conditional GANs.

Our work is also related to conditional GANs [23], which aims to apply GANs in a conditional setting. There are several works along this research direction. Previous work applies conditional GANs to discrete labels [23], text [27], image inpainting [24], image prediction from a normal map [36], image manipulation guided by user constraints [43], product photo generation [38], style transfer [22], and Image-to-Image translation [15]. Different from previous work, we use an improved adversarial loss and discriminator for training the proposed Blending GAN. We also propose the Gaussian-Poisson Equation to produce high-resolution images.

3. The Approch

In this section, we first presenting what is the problem of image composition using the copying-and-pasting strategy. We then present the framework of our Gaussian-Poisson Generative Adversarial Networks (GP-GANs).

3.1. Preliminary

Given a source image x_{src}, a destination image x_{dst} and a mask image x_{mask}, using the copying-and-pasting strategy, a composite image x can be obtained by:

$$x = x_{src} * x_{mask} + x_{dst} * (1 - x_{mask}), \qquad (1)$$

where $*$ is element-wise multiplication operator. The goal of conditional image generation is to generate a well-blended image $\tilde{x}$ that is semantically similar to the composited image x but looks more realistic and natural with resolution unchanged. x is usually a high-resolution image.

3.2. Framework Overview

Conditionally generating high-resolution images [24] is hard. To tackle this problem, we propose GP-GAN, a framework for generating high-resolution and realistic images, as shown in Figure 2. This is the first time that GAN is used for generating high-resolution and realistic images to the best of our known.

GP-GAN seeks a well-blended high-resolution image $\tilde{x}_h$ by optimising a loss function composited by colour constraint and gradient constraint. The colour constraint tries to make the generated image more realistic and natural while the gradient constraint captures the high-resolution details like textures and edges.

The colour constraint is constructed with a low-resolution realistic image $\tilde{x}_l$. To generate $\tilde{x}_l$, we propose Blending GAN $G(x)$ that learns to blend a composited

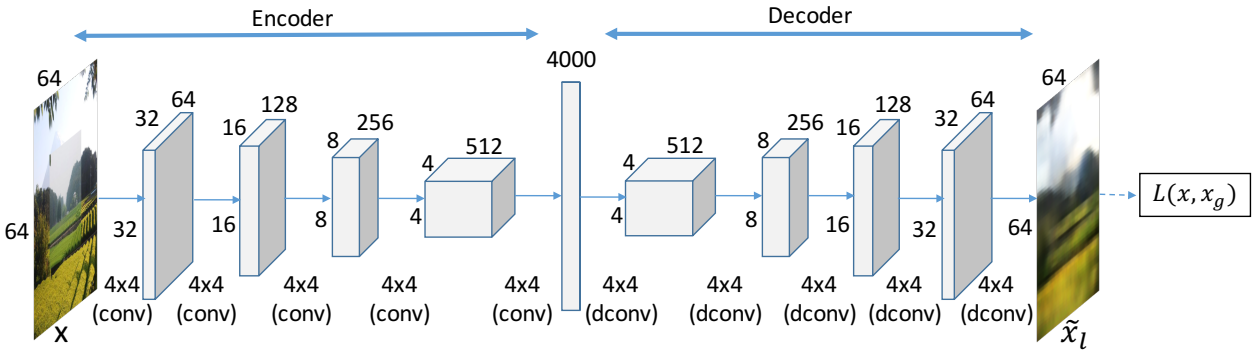


Figure 3: **Network architecture for Blending GAN G(x).** We propose Blending GAN $G(x)$ by leveraging Wasserstein GAN [2] for supervised learning tasks. The encoder-decoder architecture in deployed for $G(x)$ in our experiment. Different from [24], a standard fully connected layer is inserted between the encoder and the decoder as a bottleneck to fuse the global information. The loss function $L(x, x_g)$ is defined in Equation 2, which combines the improved adversarial loss with l_2 loss.

copy-and-paste image and generate a realistic one semantically similar to the input. Once $G(x)$ is trained, we can use it to generate $\tilde{x}_l$ functioning as the colour constraint.

The goal of gradient constraint is to generate the high-resolution details including textures and edges gave the composited image x. Textures and edges of an image are directly captured by their gradients. We propose Gaussian-Poisson Equation to force $\tilde{x}_h$ to have a similar gradient to x while as similar as $\tilde{x}_l$ in colour.

GP-GAN naturally can generate realistic images in arbitrary resolution. Given a composited image x, we first obtain $\tilde{x}_l$ by feeding x^1 to $G(x)$ where x^1 is the coarsest scale in the Laplacian pyramid of x. Then we update $\tilde{x}_h^1$ by optimising Gaussian-Poisson Equation using a closed form solution. $\tilde{x}_l$ is set to up-sampled $\tilde{x}_h^1$ and is optimised at the finer scale in the Laplacian pyramid of x. The final realistic image $\tilde{x}_h$ with the same resolution as x is obtained at the finest scale of the pyramid.

In Section 3.3, we will describe the details of our Blending GAN $G(x)$. The details of GP-GAN and Gaussian-Poisson Equation will be described in Section 3.4.

3.3. Blending GAN

We seek a low-resolution well-blended image $\tilde{x}_l$ that is visually realistic and semantically similar to the input image. A straightforward way is to train a conditional GAN and use the generator to generate realistic images. Since we have both the input image and the corresponding ground truth x_g, we aims to train a generator that can produce a realistic image approximating x_g. To achieve this goal, we propose Blending GAN $G(x)$, which leverages the unsupervised Wasserstein GAN [2] for supervised learning tasks like image blending. The proposed Blending GAN is different from Wasserstein GAN in that it has a proper constructed auxiliary loss and dedicated designed architecture for $G(x)$.

Recent work discusses various loss functions for image processing tasks in general, for instance, l_1 loss [40], l_2 loss, and perceptual loss [17]. l_1 and l_2 loss can fasten the training process but tend to produce blurry images. The perceptual loss is good at generating high-quality images but is very time and memory consuming. We explore with l_2 loss because it could shorten the training process and generate sharp and realistic images when combined with GANs [15]. The combined loss function is defined as:

$$L(x, x_g) = \lambda L_{l_2}(x, x_g) + (1 - \lambda) L_{adv}(x, x_g), \quad (2)$$

where λ is 0.999 in our experiment. L_{l_2} is defined as:

$$L_{l_2}(x, x_g) = \|G(x) - x_g\|_2^2. \quad (3)$$

and L_{adv} is defined as:

$$L_{adv}(x, x_g) = \max_D E_{x \in \chi}[D(x_g) - D(G(x))]. \quad (4)$$

The architecture for Blending GAN $G(x)$ is shown in Figure 3. We propose the encoder-decoder architecture motivated by [24]. We find that a network with only convolutional layers could not learn to blend composited images for the lack of global information across the whole image which is essential for image blending task. This suggests that standard fully connected layers are necessary for conditional image generation. Thus we replace the channel-wise fully connected layer used in [24] with standard fully connected layers.

Training such a network needs masses of data. The composited copy-and-paste image is easy to collect, but the ground truth image x_g could only be obtained by expert users, using image editing software, which is time-consuming. Alternatively, we use x_{dst} to approximate x_g since x_{src} and x_{dst} in our experiment are photos of the same scene under different conditions, *e.g.* season, weather, time

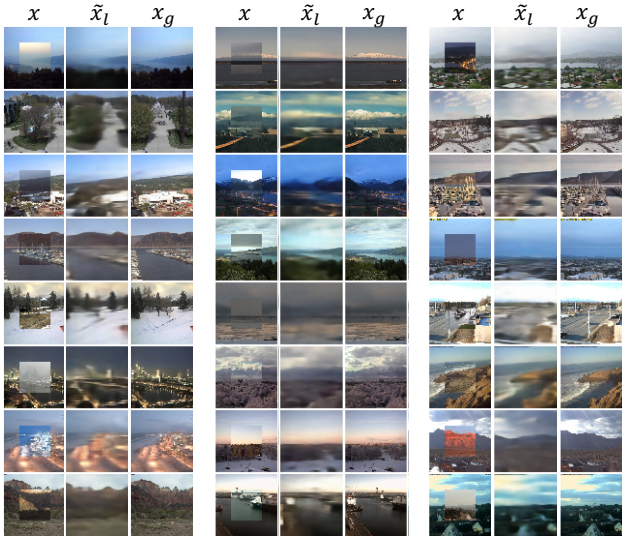


Figure 4: **Image blending results generated by $\mathbf{G(x)}$.** The experiment is conducted on the Transient Attributes Database [20]. x is the copy-and-paste images composited by x_{src} and x_{dst} with central-squared patch as mask. $\tilde{x}_l$ is the output of $G(x)$ with size 64×64. x_g is ground truth images used for training $G(x)$, x_g is the same as x_{dst}. Best viewed in color.

of day, see Section 4 for details. Through this way, we obtain masses of composited images with approximating x_g required for training our $G(x)$ properly, as shown in Figure 4.

When the ground truth is absent, we could use unsupervised generative models to model the distribution of natural images, which is discussed in the **Supplementary Material**.

3.4. Gaussian-Poisson Equation

Networks like the proposed Blending GAN $G(x)$ could only generate low-resolution images as shown in Figure 4. Even for slightly larger images, the results tend to be blurry and have unpleasant artefacts, which is unsuitable for image blending as the task usually need to combine several high-resolution images and blend them into one realistic, high-resolution image. To make use of the natural images generated by Blending GAN $G(x)$, we propose Gaussian-Poisson Equation fashioned by the well-known Laplacian pyramid [3] for generating high-resolution and realistic images.

We observe that although our Blending GAN $G(x)$ couldn't produce high-resolution images, the generated image $\tilde{x}_l$ is natural and realistic as a low-resolution image. So it is possible for us to seek a high-resolution and realistic image $\tilde{x}_h$ by approximating the colour in $\tilde{x}_l$ while capturing rich details like texture and edges in the original high-resolution image x. We formulate the requirement using two constraints: one is the colour constraint, the other is the gradient constraint. The color constraint forces $\tilde{x}_h$ to have similar color to $\tilde{x}_l$, which can be achieved by generating a $\tilde{x}_h$ with the same low-frequency signals as $\tilde{x}_l$. The simplest way to extract the low-frequency signals is using a Gaussian filter. The gradient constraint tries to restore the high-resolution details which are the same as forcing $\tilde{x}_h$ and $\tilde{x}_l$ to have the same high-frequency signals. This could be implemented by using gradient or divergence.

Formally, we need to optimize the objective function defined as:

$$H(x_h) = P(x_h) + \beta C(x_h), \tag{5}$$

$P(x_h)$ is inspired by the well-known Poisson Equation [25] and is defined as:

$$P(x_h) = \int_T \|\mathbf{div}\, v - \Delta x_h\|_2^2 \, dt, \tag{6}$$

$C(x_h)$ is defined as:

$$C(x_h) = \int_T \|g(x_h) - \tilde{x}_l\|_2^2 \, dt, \tag{7}$$

and β represents the color preserving parameter. We set β to 1 in our experiment. In Equation 6, T represents the whole image region, **div** represents the divergence operator and Δ represents the Laplacian operator. v is defined as:

$$v^{ij} = \begin{cases} \nabla x_{src} & \text{if } x_{mask}^{ij} = 1 \\ \nabla x_{dst} & \text{if } x_{mask}^{ij} = 0 \end{cases} \tag{8}$$

where ∇ is the gradient operator. Gaussian filter is used in Equation 7 and is denoted as $g(x_h)$. The discretized version of Equation 5 is defined as:

$$H(x_h) = \|u - L\mathbf{x_h}\|_2^2 + \lambda \|G\mathbf{x_h} - \tilde{\mathbf{x}}_\mathbf{l}\|_2^2, \tag{9}$$

u is the discretized divergence of the vector field v, L is the matrix which represents the Laplacian operator while G represents the Gaussian filter matrix. $\mathbf{x_h}$ and $\tilde{\mathbf{x}}_\mathbf{l}$ are the vector representation of x_h and $\tilde{x}_l$. The closed form solution for minimizing the cost function of Equation 9 could be obtained in the same manner as [10].

We integrate the closed form solution for optimizing Equation 9 and the Laplacian pyramid into our final high-resolution image blending algorithm, which is described by Algorithm 1. Given high-resolution input images x_{src}, x_{dst} and x_{mask}, we first generate the low-resolution realistic image $\tilde{x}_l$ using Blending GAN $G(x)$. Then we generate Laplacian pyramids $x_{src}^s, x_{dst}^s, x_{mask}^s, s = 1, 2, ..., S$, where S is the number of scales. $s = 1$ is the coarsest scale and $s = S$ is the original resolution. We update x_h^s by optimizing Equation 9 at each scale and set $\tilde{x}_l$ to be up-sampled x_h^s. The final realistic image $\tilde{x}_h$ with unchanged resolution is set to be x_h^S.

Algorithm 1: High-Resolution Conditional Image Generation using GAN with Poisson Equation

Input : Source image x_{src}, destination image x_{dst}, mask image x_{mask} and trained Blending GAN $G(x)$

Compute Laplacian Pyramid for x_{src}, x_{dst} and x_{mask}
Compute $\tilde{x}_l$ using $G(x)$
for $s \in [1, 2, ..., S]$ **do**
    Updating x_h^s by optimizing Equation 9 using the closed form solution given x_{src}^s, x_{dst}^s, x_{mask}^s and $\tilde{x}_l$
    Set $\tilde{x}_l$ by up-sampling x_h^s
end
Return x_h^S

4. Experiments

This section describes our experimental settings, the results of our algorithm and comparisons with other methods both quantitatively and qualitatively in detail. We first introduce the datasets we used in our experiments. We then give the network training configurations and experimental settings. Finally, we show the effectiveness of our methods both quantitatively and visually.

Datasets Transient Attributes Database [20] contains 8571 images from 101 webcams. In each webcam, there are well-aligned 60-120 images with severe appearance changes caused by weather, time of day and season, as shown in Figure 5a and Figure 5b. We use this database to train our Blending GAN $G(x)$, because the ground truth for image blending is difficult to obtain, we instead use x_{dst} to approximate x_g since images under the same webcam is perfect-aligned in Transient Attributes Database. When training $G(x)$, we randomly select 2 images from the same camera as x_{src} and x_{dst}, x_g is the same as x_{dst}. With x_{dst} as x_g, $G(x)$ learns to blend x_{src} and x_{dst} to be consistent in weather, time of day and season. x_{mask} is a binary image with a central-squared patch filled with 1s, as shown in Figure 5c. The composited copy-and-paste image is also shown in Figure 5d as well as Figure 4. Although we train $G(x)$ using the central-squared patch as masks, $G(x)$ is still able to generate well-blended images for inputs with arbitrary masks as our experiment showed. We also use this dataset to evaluate our algorithm. To evaluate our method with arbitrary masks, we first manually annotate the object-level masks for Transient Attributes dataset using the LabelMe [28] annotation tool. Then we use such annotations as masks to composite the copy-and-paste images and evaluated different image blending methods with them. Annotated mask and corresponding composited image are shown in Figure 5e and Figure 5f.

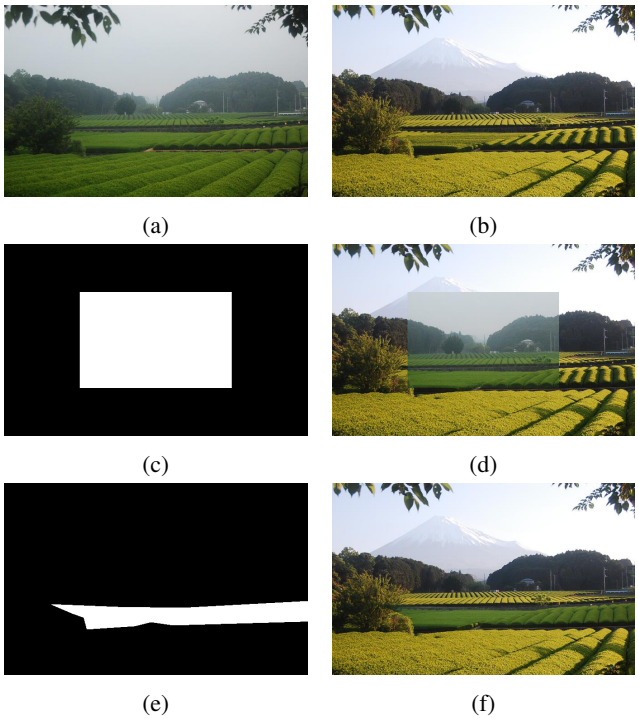


Figure 5: **Transient Attributes Database.** (a) x_{src} and (b) x_{dst} from the same webcam but in different season. (c) the central-squared mask and (d) the corresponding composited image. (e) the annotated object-level mask using LabelMe. (f) composited image using annotated mask. Best viewed in color.

Implementation Details Our algorithms are implemented using chainer[1] [33]. To train Blending GAN $G(x)$, we use ADAM [19] for optimization and set α to 0.002, β_1 to 0.5 for the encoder-decoder network as suggested while setting α to 0.0002 for the discriminator $D(x)$. We randomly generate 150K images from Transient Attributes Database using the central-squared patch as the mask. Then the network is trained for 25 epochs using the generated images with batch size 64.

Experimental Settings We compare our method with several baseline methods on Transient Attributes Database using the annotated masks. Poisson Image Editing [25] and its improved version Modified Poisson Image Editing [32] are selected as baselines because both of them use Poisson Equation as part of their solutions to solve image blending task as our method. We also compare our method with multi-splines blending [31] for its effectiveness and widely spread as Poisson Image Editing [25] and Modified Poisson Image Editing [32].

Quantitative Comparisons We firstly evaluate our method as well as baseline methods quantitatively on Transient Attributes Database using annotated masks. The goal

[1]The code and pre-trained models will be released on Github soon.

(a) (b) (c)

Figure 6: **Role of Blending GAN.** (a) composited copy-and-paste image. (b) using down-sampled (a) as color constraint. (c) using the output of Blending GAN as color constraint. Best viewed in color.

of image blending algorithms is to generate realistic images given the composited copy-and-paste images. However, distinguishing between realistic and artificial images is incredibly hard. There are no suitable metrics for judging how realistic and natural an image is until recently a convolutional neural network called RealismCNN [42] is proposed. RealismCNN learns to predict visual realism of a scene regarding colour, lighting and texture compatibility. Experiment results show that such a network outperforms previous works that rely on hand-crafted heuristics and achieves near-human performance. Thus it is an appropriate choice for evaluating image blending algorithms. We randomly generate 500 images from Transient Attributes Database using annotated masks, applied different algorithms and obtain 2000 blended images. Then we use RealismCNN to evaluate each resulted image. After that, we report the average scores and the standard deviations for our method as well as baselines in Table 1. The result shows that our GP-GAN is better than all the baselines. We attribute this to the nature of our method because the network could learn what contributes to a realistic and natural image through adversarial learning on large datasets. The negative average scores for all evaluated methods show that many blended images are still not realistic, which suggests that there are still many improvements to be made for image blending algorithms.

User Study Realism scores show the effectiveness of our method. To evaluate our method further, it is essential to conduct user study since image blending task is user-oriented. We randomly generate 500 images from Transient Attributes Database using annotated masks, apply different algorithms and obtain 2000 blended images as described in **Quantitative Comparisons**, see Figure 8 for examples. Then we collect the user assessments using the Amazon Mechanical Turk. Each time, a composited copy-and-paste image x is shown to the subjects followed by three blended images produced by three different algorithms with x as input. The subjects are told to pick the most realistic and natural image among these three blended images, as shown in Figure 7. Crowdsourced evaluation tends to be noisy. Thus

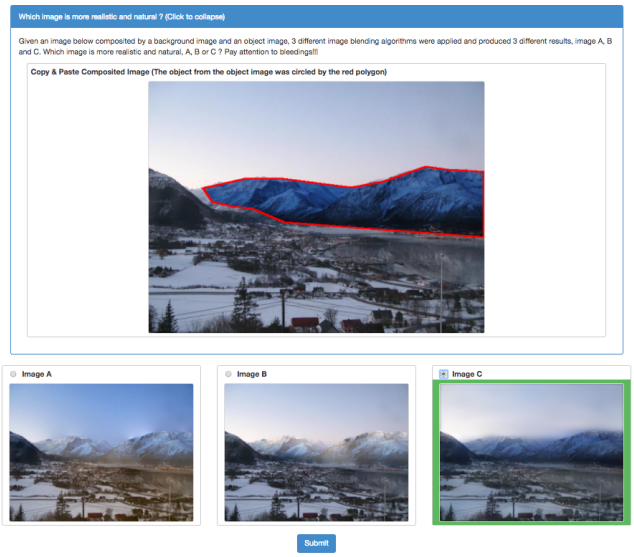


Figure 7: **User interface for user study on Amazon Mechanical Turk.** Followed by the composited image with x_{src} circled out, three blended results generated by different algorithms are shown to subjects, and the most realistic one is picked out. Best viewed in colour.

Method	Average score	Std.
copy-and-paste	-0.696	1.799
PB[25]	-0.192	1.565
MPB[32]	-0.151	1.561
MSB[31]	-0.140	1.557
GP-GAN	-0.069	1.385

Table 1: **Realism scores evaluated by RealismCNN, higher is better.** Our GP-GAN is better than baselines which shows the effectiveness of our method. Negative scores suggest that there's still improvement to be made for image blending algorithms. Large standard deviations show that all the methods are not stable for different images.

data preprocessing needs to be applied before actual data analysis. For each composited image, we give a method one vote if it is picked more by subjects than the other method. The statistical result of the processed data is reported in Table 2. GP-GAN is preferred by the majority users, which is consistent with the result of realism scores in Table 1.

Role of Blending GAN The output of our Blending GAN $G(x)$ serves as the colour constraint in our final algorithm. Realism scores and user study show the effectiveness of GP-GAN. In this section, we demonstrate the role of our low-resolution image blending algorithm by replacing $\tilde{x}_l$ with down-sampled composited image x^1. We compare the blended results with either $\tilde{x}_l$ or x^1 as the colour constraint. As shown in Figure 6, the blended image tends to have more bleedings and illumination inconsistencies if $\tilde{x}_l$ is replaced

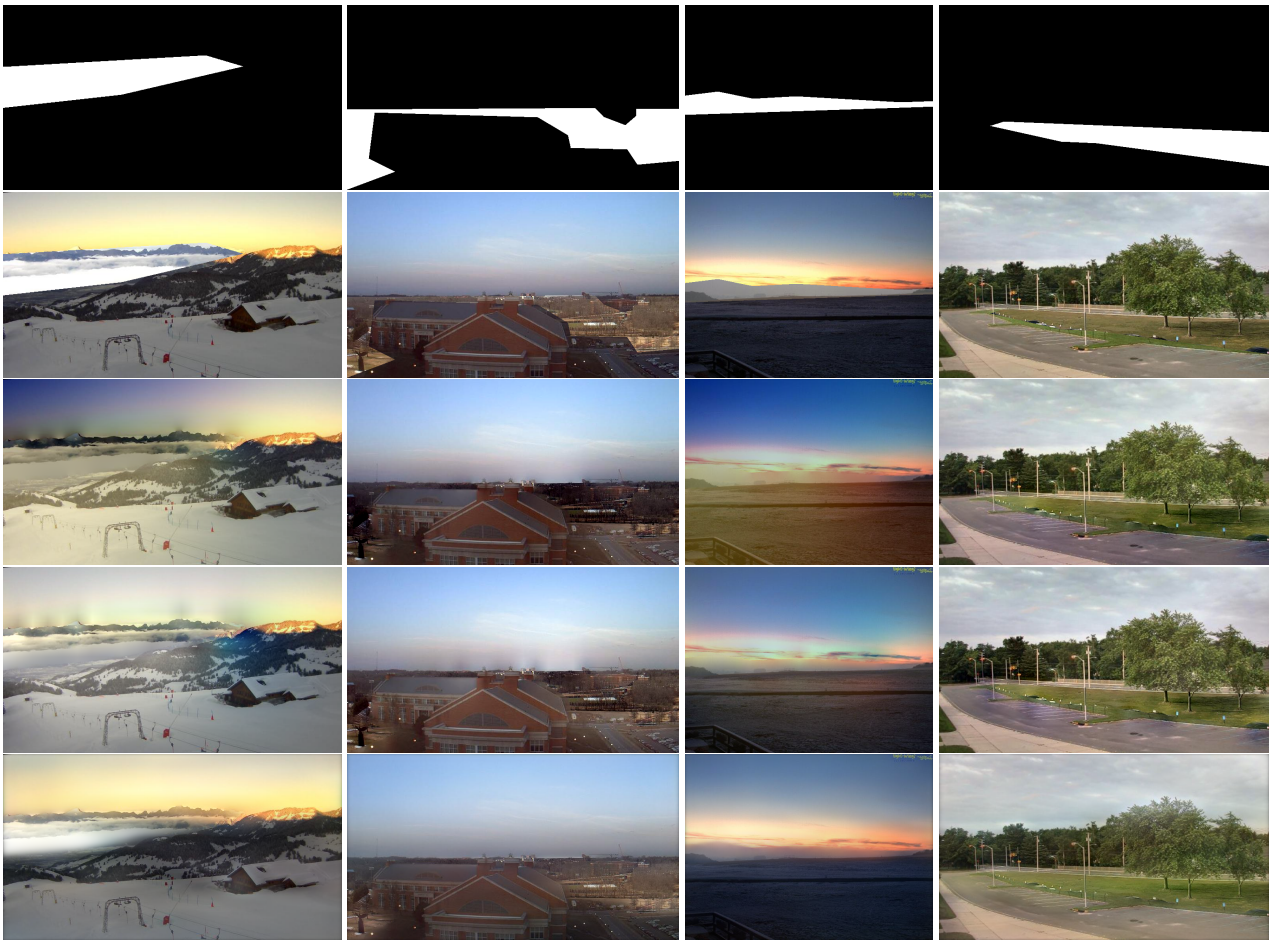


Figure 8: **Results of our high-resolution blending algorithm comparing with baseline methods.** From top to bottom: annotated object-level mask, composited copy-and-paste image, Modified Poisson Image Editing, Multi-splines Blending, GP-GAN(ours). Results of baseline methods have severe bleedings, illumination inconsistencies or other artefacts while PoissonGAN produces pleasant, realistic images. Best viewed in colour.

Method	Total votes	Average votes	Std.
PB[25]	527	1.054	1.065
MPB[32]	735	1.470	1.173
MSB[31]	770	1.540	1.271
GP-GAN	947	1.894	1.311

Table 2: **User study result.** 4 image blending algorithms were compared using Amazon Mechanical Turk. Our method GP-GAN got most votes by users which is consistent with the result of the realism scores. The result also suggests that among all 3 widely used traditional methods, Poisson Image Editing should be the last one to be used. Multi-splines Blending and Modified Poisson Image Editing have similar performance, one of them should be applied if the computation resource is limited and our method couldn't be deployed.

by x^1, which explains the usefulness of low-resolution natural images in our high-resolution image blending algorithm.

Qualitative Comparisons Finally, we demonstrate the results of our high-resolution image blending algorithm visually by comparing with Modified Poisson Image Editing and Multi-splines Blending. Results are shown in Figure 8. Our method tends to generate realistic results while preserving the appearance of both x_{src} and x_{dst}. Compared to the baseline methods, there are nearly no bleedings or illumination inconsistencies in our results while all the baseline methods have more or fewer bleedings and artefacts.

5. Conclusion

We advanced the state-of-the-art in conditional image generation by combining the ideas from the generative model GAN, Laplacian Pyramid, and Gauss-Poisson Equation. This combination is the first time a generative model could produce realistic images in arbitrary resolution. Our insight is, on the one hand, the conditional GAN is good at generating natural images from a particular distribution

but weak in capturing the high-frequency image details like textures and edges. On the other hand, the gradient-based methods perform well at generating high-resolution images with local consistency though the generated images tend to be unnatural and have many artefacts. GAN and gradient-based methods should be integrated together. Hence, this integration would result a conditional image generation system that overcomes the drawbacks of both methods. Our system can also be useful for image-to-image translation task. In spite of the effectiveness, our algorithm fails to generate realistic images when the composited images are far away from the distribution of the training dataset. We aim to address this issue in future work.

References

[1] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. *ACM TOG*, pages 294–302, 2004. 3

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *ArXiv*, 2017. 2, 3, 4, 10

[3] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on communications*, 31(4):532–540, 1983. 5

[4] P. J. Burt and E. H. Adelson. A multiresolution spline with application to image mosaics. *ACM TOG*, 2(4):217–236, 1983. 3

[5] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995. 10

[6] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, 2016. 3

[7] E. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, 2015. 2, 3

[8] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *NIPS*, pages 658–666, 2016. 3

[9] R. Fattal, D. Lischinski, and M. Werman. Gradient domain high dynamic range compression. In *ACM SIGGRAPH*, 2002. 3

[10] R. T. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE TPAMI*, 10(4):439–451, 1988. 5

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014. 2, 3

[12] N. Gracias, A. Gleason, S. Negahdaripour, and M. H. Mahoor:. Fast image blending using watersheds and graph cuts. In *BMVC*, 2006. 3

[13] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. In *ICML*, 2015. 3

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ArXiv*, 2015. 10, 11

[15] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2, 3, 4

[16] J. Jia, J. Sun, C. Tang, and H. Shum. Drag-and-drop pasting. *ACM TOG*, pages 631–637, 2006. 3

[17] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711. Springer, 2016. 4

[18] M. Kazhdan and H. Hoppe. Streaming multigrid for gradient-domain operations on large images. *ACM TOG*, 27:1–10, 2008. 3

[19] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *ArXiv*, 2014. 6

[20] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM TOG*, 33(4), 2014. 5, 6, 10

[21] A. Levin, A. Zomet, S. Peleg, and Y. Weiss. Seamless image stitching in the gradient domain. In *ECCV*, 2004. 3

[22] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *ECCV*, 2016. 3

[23] M. Mirza and S. Osindero. Conditional generative adversarial nets. In *ArXiv*, 2014. 2, 3

[24] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544, 2016. 2, 3, 4

[25] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM TOG*, 22(3):313–318, 2003. 2, 5, 6, 7, 8, 11

[26] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*, 2016. 2, 3

[27] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *ICML*, 2016. 3

[28] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *IJCV*, 77(1-3):157–173, 2008. 6

[29] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *NIPS*, 2016. 3

[30] R. Szeliski, M. Uyttendaele, and D. Steedly. Fast poisson blending using multi-splines. In *MSR-TR-2008-58*, 2008. 3

[31] R. Szeliski, M. Uyttendaele, and D. Steedly. Fast poisson blending using multi-splines. In *ICCP*, pages 1–8. IEEE, 2011. 6, 7, 8, 11, 12, 13

[32] M. Tanaka, R. Kamio, and M. Okutomi. Seamless image cloning by a closed form solution of a modified poisson problem. In *SIGGRAPH Asia 2012 Posters*, page 15. ACM, 2012. 1, 6, 7, 8, 11, 12, 13

[33] S. Tokui, K. Oono, S. Hido, and J. Clayton. Chainer: a next-generation open source framework for deep learning. In *Proceedings of workshop on machine learning systems (LearningSys) in NIPS*, 2015. 6

[34] Y.-H. Tsai, X. Shen, Z. Lin, K. Sunkavalli, and M.-H. Yang. Sky is not the limit: semantic-aware sky replacement. *ACM TOG*, 35:1–149, 2016. 2

[35] M. Uyttendaele, A. Eden, and R. Szeliski. Eliminating ghosting and exposure artifacts in image mosaics. In *CVPR*, 2001. 3

[36] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *ECCV*, 2016. 3

[37] C. Yang, X. Lu, Z. L. anbd Eli Shechtman, O. Wang, and H. Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *CVPR*, 2017. 2

[38] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. S. Kweon. Pixellevel domain transfer. In *ECCV*, 2016. 3

[39] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ArXiv*, 2016. 3

[40] H. Zhao, O. Gallo, I. Frosio, and J. Kautz. Is l2 a good loss function for neural networks for image processing? *ArXiv*, 1511, 2015. 4

[41] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, pages 487–495, 2014. 10

[42] J.-Y. Zhu, P. Krahenbuhl, E. Shechtman, and A. A. Efros. Learning a discriminative model for the perception of realism in composite images. In *ICCV*, pages 3943–3951, 2015. 7, 10

[43] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, 2016. 3

Supplementary Material

In this section, we describe our framework GP-GAN in an unsupervised setting. Then, the architectures of the networks used in our paper are shown. Finally, we show additional qualitative results compared with the baseline methods.

A. Unsupervised Blending GAN

The ground truth for image blending task is expensive to obtain. To generalise the applicability of our high-resolution image blending algorithm GP-GAN, we leverage the Blending GAN described in Section 3.3 (main text) to an unsupervised setting and propose an unsupervised Blending GAN when the ground truth is absent. In particular, we develop an unsupervised Blending GAN based on the Wasserstein GAN [2] to model the distribution of natural images. This unsupervised Blending GAN consists of two networks, a generator $G(z)$ and a discriminator $D(x)$. Through adversarial training using the min-max objective, $G(z)$ can learn to model the distribution of the training set χ_{GAN}.

We first treat the pre-trained $G(z)$ as an ideal manifold of χ_{GAN}. In the unsupervised setting, our goal is to find a vector $\tilde{z}$ that has the closest similar appearance to that with the low-resolution composited input image x. To find an optimal $\tilde{z}$, we formulate an optimisation problem as the follows:

$$\tilde{z} = arg\,min_z L(x, G(z)), \tag{10}$$

where, $L(x_1, x_2)$ is L_2 loss in our system, although it is also possible to use L_1 loss, and perceptual loss. Given L_2 loss setting, this objective function becomes a convex one and it is continuously differentiable. This L_2 loss function choice also allows us to use L-BFGS-B[5], which is faster and takes less memory compared to that with the use of perceptual loss. By optimising Equation 10, we would achieve the sample $G(\tilde{z})$ that approximates x and lyes on the manifold $G(z)$, as shown in Figure 9.

Training GANs requires massive numbers of images. Because there are insufficient numbers of images in the Transient Attributes Dataset [20], we also train the generator $G(z)$ based on the MIT Places dataset [41], which provides 150K landscape images and has the similar collections of images as that in the Transient Attributes Dataset.

Our experimental results indicate that the trained generator $G(z)$ can generate high-quality composited images, as shown in Figure 9c and Figure 9d.

To generate high-resolution and realistic images in an unsupervised setting, we replace Blending GAN with the unsupervised Blending GAN in our high-resolution image blending framework GP-GAN. The colour preserving parameter β is set to 0.1, since the low-resolution natural images generated by $G(\tilde{z})$ is slightly worse than the ones produced by Blending GAN in Section 3.3 (main text).

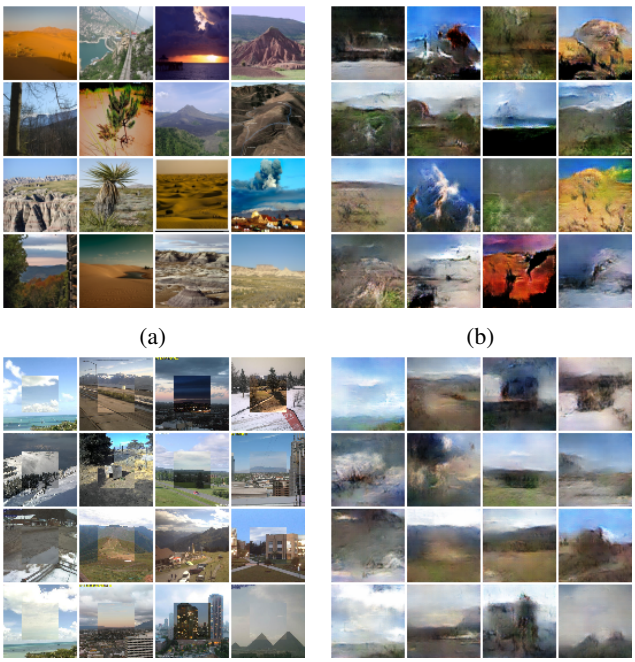


Figure 9: (a) shows the images random sampled from 150K landscape images of MIT Places dataset [41]. (b) illustrates the images generated by $G(z)$ given random sampled vector z. (c) contains low-resolution composited copy-and-paste images x from the Transient Attributes Database. (d) presents the generated images by optimising Equation 10 given (c) as inputs.

We evaluate the unsupervised high-resolution image blending method using the evaluation pipeline described in RealismCNN [42] followed the same manner in Section 4 (main text). Our method achieved the average realism score -0.110 with standard deviation 1.459. These results are better than all the baseline methods [42] and slightly worse than the proposed GP-GAN in the supervised setting. Resulted images are shown in Figure 12 with the same inputs as Figure 8 (main text).

B. Network Architectures

The architecture for the discriminator of Blending GAN in Section 3.3 (main text) is shown in Figure 10. We apply the batch normalization [14] and the leaky ReLU non-linearity after each convolution operation except the first layer and the last layer. The first layer uses convolutional operation and leaky ReLU non-linearity while the last layer contains convolution operation only.

The architecture for the unsupervised Blending GAN is shown in Figure 11. The discriminator $D(x)$ is the same as the discriminator in Figure 10. Every convolution operation in the generator $G(z)$ is followed by batch normaliza-

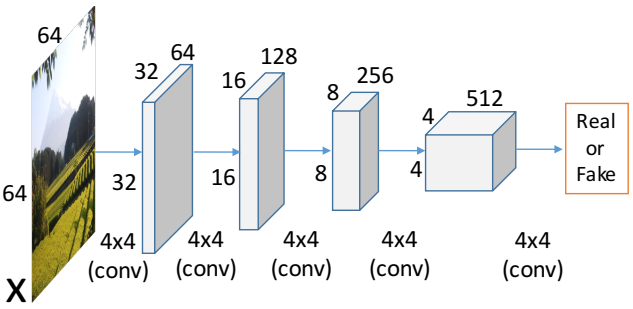


Figure 10: The architecture for the discriminator of Blending GAN in Section 3.3 (main text).

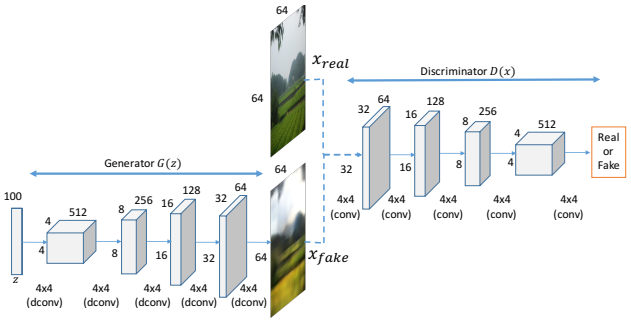


Figure 11: The architecture for unsupervised Blending GAN. Left part is the generator $G(z)$ while right part is the discriminator $D(x)$.

tion [14] and ReLU non-linearity except the last layer. After convolution operation, $tanh$ non-linearity is applied in the last layer.

C. Additional Results

Additional results are shown in Figure 13 and Figure 14. The results of Modified Poisson Image Editing [25], Multi-splines Blending [31] and our GP-GAN both in supervised and unsupervised setting are shown. The results of original Poisson Image Editing [25] are not shown since the other methods [31, 32] are better than it based on both the realism scores and the voting results from user study. Compared to the baseline methods, there are nearly no bleedings or illumination inconsistencies in our results while all the baseline methods have more or fewer bleedings and artefacts. GP-GAN in the unsupervised setting is slightly worse than that in the supervised setting.

Figure 12: Results of our high-resolution blending algorithm in a unsupervised setting. Same inputs as Figure 8 (main text).

Figure 13: Results of our high-resolution image blending algorithm compared with baseline methods. From top to bottom: annotated object-level mask, composited copy-and-paste image, Modified Poisson Image Editing [32], Multi-splines Blending [31], supervised GP-GAN and unsupervised GP-GAN. Results of baseline methods have severe bleedings, illumination inconsistencies or other artefacts while our GP-GAN both in supervised setting and unsupervised setting produce pleasant and realistic images.

Figure 14: Results of our high-resolution image blending algorithm compared with baseline methods. From top to bottom: annotated object-level mask, composited copy-and-paste image, Modified Poisson Image Editing [32], Multi-splines Blending [31], supervised GP-GAN and unsupervised GP-GAN. Results of baseline methods have severe bleedings, illumination inconsistencies or other artifacts while our GP-GAN both in supervised setting and unsupervised setting produce pleasant realistic images.