Compositional GAN: 
Learning Conditional Image Composition

Samaneh Azadi, Deepak Pathak, Sayna Ebrahimi, Trevor Darrell
University of California, Berkeley
{sazadi, pathak, sayna, trevor}@eecs.berkeley.edu

Abstract

Generative Adversarial Networks (GANs) can produce images of surprising complexity and realism, but are generally modeled to sample from a single latent source ignoring the explicit spatial interaction between multiple entities that could be present in a scene. Capturing such complex interactions between different objects in the world, including their relative scaling, spatial layout, occlusion, or viewpoint transformation is a challenging problem. In this work, we propose to model object composition in a GAN framework as a self-consistent composition-decomposition network. Our model is conditioned on the object images from their marginal distributions to generate a realistic image from their joint distribution by explicitly learning the possible interactions. We evaluate our model through qualitative experiments and user evaluations in both the scenarios when either paired or unpaired examples for the individual object images and the joint scenes are given during training. Our results reveal that the learned model captures potential interactions between the two object domains given as input to output new instances of composed scene at test time in a reasonable fashion.

1 Introduction

Generative Adversarial Networks (GANs) have emerged as a powerful method for generating images conditioned on a given input. The input cue could be in the form of an image [11, 40, 18, 2, 32, 23], a text phrase [36, 26, 25, 13] or a class label layout [21, 22, 1]. The goal in most of these GAN instantiations is to learn a mapping that translates a given sample from source distribution to generate a sample from the output distribution. This primarily involves transforming either a single object of interest (apples to oranges, horses to zebras, label to image etc.), or changing the style and texture of the input image (day to night etc.). However, these direct input-centric transformations do not directly capture the fact that a natural image is a 2D projection of a composition of multiple objects interacting in a 3D visual world. In this work, we explore the role of compositionality in learning a function that maps images of different objects sampled from their marginal distributions (e.g., chair and table) into a combined sample (table-chair) that captures their joint distribution.

Modeling compositionality in natural images is a challenging problem due to the complex interaction possible among different objects with respect to relative scaling, spatial layout, occlusion or viewpoint transformation. Recent work using spatial transformer networks [12] within a GAN framework [17] decomposes this problem by operating in a geometric warp parameter space to find a geometric modification for a foreground object. However, this approach is only limited to a fixed background and does not consider more complex interactions in the real world. Another recent work on scene generation conditioned on text and a scene graph, which explicitly provides reasoning about objects and their relations [13].

We develop an approach to model object compositionality in images. We consider the task of composing two input object images into a joint image that captures their joint interaction in natural images. For instance, given an image of a chair and a table, our formulation should be able to generate
an image containing the same chair-table pair interacting naturally. For a model to be able to capture the composition correctly, it needs to have the knowledge of occlusion ordering, i.e., table comes in front of chair, and spatial layout, i.e., a chair slides inside table. To the best of our knowledge, we are among the first to solve this problem in the image conditional space without any prior explicit information about the objects layout.

Our key insight is to reformulate the problem of composition of two objects into first composing the given object images to generate the joint combined image which models the object interaction, and then decomposing the joint image back to obtain individual ones. This reformulation enforces a self-consistency constraint [40] through a composition-decomposition network. However, in some scenarios, one does not have access to the paired examples of same object instances with their combined compositional image, for instance, to generate the joint image from the image of a given table and a chair, we might not have any example of that particular chair besides that particular table while we might have images of other chairs and other tables together. We add an inpainting network to our composition-decomposition layers to handle the unpaired case as well.

Through qualitative and quantitative experiments, we evaluate our proposed Compositional-GAN approach in two training scenarios: (a) paired: when we have access to paired examples of individual object images with their corresponding composed image, (b) unpaired: when we have a dataset from the joint distribution without being paired with any of the images from the marginal distributions.

Our compositional GAN code and dataset are available at [https://github.com/azadis/CompositionalGAN](https://github.com/azadis/CompositionalGAN).

2 Related Work

Generative adversarial networks (GANs) have been used in a wide variety of settings including image generation [7][13][14] and representation learning [24][28][19][6]. The loss function in GANs have been shown to be very effective in optimizing high quality images conditioned on available information. Conditional GANs [21] generate appealing images in a variety of applications including image to image translation both in the case of paired [11] and unpaired data [40], inpainting missing image regions [23][34], generating photorealistic images from labels [21][22], and solving for photo super-resolution [16][15].

Learning compositional structure has been studied in earlier works such as [4] where composition was introduced as a geometric and data-driven approach for measuring similarities between two set of signals. Two signals are locally similar if regions in one signal can be composed using large contiguous chunks of data from the other signal. Similarly, composition could address the problem of detecting irregularities in visual data as a process of constructing a puzzle [3]. More recently, each image was explained as a spatial composite of partial regions taken from different database images with the end goal of unsupervised scene segmentation [27].

Image compositing is a challenging problem in computer graphics where masked objects from different images are to be overlaid in one single image. Appearance and geometric differences between these objects are obstacles that can result in non-realistic composed images. Zhu et.al. [39] proposed to learn optimal parameters of a compositing method as well as a discriminator (a CNN) that could distinguish realistic composite images from non-realistic ones. Tsai et.al. [31] developed an end-to-end deep CNN for image harmonization which could automatically capture the context and semantic information of the composite image and was able to outperform its precedents [30][33] which used transferring statistics of hand-crafted features including color and texture to harmonize the foreground and background in the composite image. A more recent paper [17] introduced a novel GAN that uses spatial-transformer networks as the generator that performs geometric corrections to sequentially warp a masked object to adapt to a fixed background image. Another recent paper [13] computed a scene layout starting from given scene graphs which revealed an explicit reasoning about relationships between objects, and converted the layout to an output image. Despite the success all these approaches gained in improving perceptual realism, they lack the realistic complex problem statement where no prior information about the scene layout is given. In the general case which we address, each object should be rotated, scaled, and translated in addition to occluding others and/or being occluded to generate a realistic composite image.
3 Compositional GAN Architecture

We propose a generative network for composing two objects by learning how to handle their relative scaling, spatial layout, occlusion, and viewpoint transformation. Given a set of images from the marginal distribution of the first object, \( X = \{x_1, \ldots, x_n\} \), and a set of images from the marginal distribution of the second object, \( Y = \{y_1, \ldots, y_n\} \), in addition to a set of real images from their joint distribution containing both objects, \( C = \{c_1, \ldots, c_n\} \), we generate realistic composite images containing objects given from the first two sets. We propose a conditional generative adversarial network for two scenarios: (1) paired inputs-output in the training set where each image in \( C \) is correlated with an image in \( X \) and one in \( Y \), and (2) unpaired training data where images in \( C \) are not paired with images in \( X \) and \( Y \). It is worth noting that our goal is not to learn a generative model of all possible compositions, but to learn to output the mode of the distribution. The modular components of our proposed approach are critical in learning the mode of plausible compositions. For instance, our relative appearance flow network handles the viewpoint and the spatial transformer network handles affine transformation eventually making the generator invariant to these transformations. In the following sections, we first summarize a conditional generative adversarial network, and then will discuss our network architecture and its components for the two circumstances.

3.1 Conditional Generative Adversarial Networks

Starting from a random noise vector, \( z \), GANs generate images \( c \) of a specific distribution by adversarially training a generator, \( G \), versus a discriminator, \( D \). While the generator tries to produce realistic images, the discriminator opposes the generator by learning to distinguish between real and fake images. In the conditional GAN models (cGANs), an auxiliary information, \( x_i \), in the form of an image or a label is fed into the model alongside the noise vector \( \{x, z\} \) \([8,21]\). The objective of cGANs would be therefore an adversarial loss function formulated as:

\[
L_{cGAN}(G, D) = \mathbb{E}_{x,c \sim \text{data}}[\log D(x, c)] + \mathbb{E}_{x \sim \text{data}, z \sim \text{p}(z)}[1 - \log D(x, G(x, z))]
\]

where \( G \) and \( D \) minimize and maximize this loss function, respectively.

The convergence of the above GAN objective and consequently the quality of generated images would be improved if an \( L_1 \) loss penalizing deviation of generated images from their ground-truth is added. Thus, the generator’s objective function would be summarized as:

\[
G^* = \arg \min_G \max_D L_{cGAN}(G, D) + \lambda \mathbb{E}_{x,c \sim \text{data}}[\|c - G(x, z)\|_1]
\]

In our proposed compositional GAN, model \( \{(x, y, z) \to c\} \) is conditioned on \( \{x, y\} \), concatenated channel-wise in order to generate an image from the target distribution \( \text{p}_{\text{data}}(c) \). We have access to real samples of these three distributions during training (in two paired and unpaired scenarios). Similar to \([11]\), we ignore random noise as the input to the generator, and dropout is the only source of randomness in the network.

3.2 Relative Appearance Flow Network (RAFN)

In some specific domains, the relative viewpoint of the objects should be changed accordingly to generate a natural composite image. Irrespective of the paired or unpaired inputs-output cases, we train a relative encoder-decoder appearance flow network \([38]\), \( G_{\text{RAFN}} \), taking the two input images and synthesizing a new viewpoint of the first object, \( x_{\text{RAFN}} \), given the viewpoint of the second one, \( y_i \) encoded in its binary mask. The relative appearance flow network is trained on a set of images in \( X \) with arbitrary azimuth angles \( \alpha_i \in \{-180^\circ, -170^\circ, \ldots, 170^\circ, 180^\circ\} \) along with their target images in an arbitrary new viewpoint with azimuth angle \( \theta_i \in \{-180^\circ, -170^\circ, \ldots, 170^\circ, 180^\circ\} \) and a set of foreground masks of images in \( Y \) in the target viewpoints. The network architecture for our relative appearance flow network is illustrated in the appendix and its loss function is formulated as:

\[
\mathcal{L}(G_{\text{RAFN}}) = \mathcal{L}_{\text{L1}}(G_{\text{RAFN}}) + \lambda \mathcal{L}_{\text{BCE}}(G_{\text{RAFN}}^M) = \mathbb{E}_{(x_i, y_i) \sim \text{plan}(x_i, y_i)}[\|x_i' - G_{\text{RAFN}}(M_{fg}^x, x_i)\|_1] + \lambda \mathbb{E}_{x_i \sim \text{plan}(x_i)}[M_{fg}^x \log M_{fg}^x + (1 - M_{fg}^x) \log (1 - M_{fg}^x)]
\]

As mentioned above, \( G_{\text{RAFN}} \) is the encoder-decoder network predicting appearance flow vectors, which after a bilinear sampling generates the synthesized view. The encoder-decoder mask generating network, \( G_{\text{RAFN}}^M \), shares weights in its encoder with \( G_{\text{RAFN}} \), but its decoder is designed for predicting
AFN discussed in the previous section, to relatively translate the center-oriented input objects, we make them look realistic in addition to multiple simplicity. A GAN loss with gradient penalty [9] is applied on top of generated images based on their spatial relation encoded in the training composite images. The architecture of this network is illustrated in Appendix B.

The main backbone of our proposed model consists of a self-consistent composition-decomposition network both as conditional generative adversarial networks. The composition network, $G_c$, takes the two translated input RGB images, $x_i^T$ and $y_i^T$, concatenated channel-wise in a batch, with size $N \times 6 \times H \times W$, and generates their corresponding output, $\hat{c}_i$, with size $N \times 3 \times H \times W$ composed of the two input images appropriately. This generated image will be then fed into the decomposition network, $G_{\text{dec}}$, to be decomposed back into its constituent objects, $\hat{x}_i^T$ and $\hat{y}_i^T$, in addition to $G^{M\text{dec}}$ that predicts probability segmentation masks of the composed image, $M_x$, and $M_y$.

The two decomposition components $G_{\text{dec}}$ and $G^{M\text{dec}}$ share their weights in their encoder network but are different in the decoder. We assume the ground-truth foreground masks of the inputs and the target composite image are available, thus we remove background from all images in the network for simplicity. A GAN loss with gradient penalty [9] is applied on top of generated images $\hat{c}_i, \hat{x}_i^T, \hat{y}_i^T$ to make them look realistic in addition to multiple $L_1$ loss functions penalizing deviation of generated images from their ground-truth. An schematic of our full network, $G$, is represented in Figure 1 and the loss function is summarized as:

$$L(G) = \lambda_1 [L_{L_1}(G_c) + L_{L_1}(G_{\text{dec}}) + L_{L_1}(\text{STN})] + \lambda_2 L_{\text{CE}}(G^{M\text{dec}}) + \lambda_3 L(G_{\text{RAFN}}) + \lambda_4 [L_{\text{cGAN}}(G_c, D_c) + L_{\text{cGAN}}(G_{\text{dec}}, D_{\text{dec}})]$$ (2)
where

\[ L_{L_1}(\text{STN}) = \mathbb{E}_{(x_i, y_i) \sim p_{\text{data}}(x_i, y_i)}[\| (x_i^c, y_i^c) - (x_i^T, y_i^T) \|_1] \]

\[ L_{L_1}(G_c) = \mathbb{E}_{(x_i, y_i, c_i) \sim p_{\text{data}}(x_i, y_i, c_i)}[\| c_i - \hat{c}_i \|_1] \]

\[ L_{L_1}(G_{\text{dec}}) = \mathbb{E}_{(x_i, y_i) \sim p_{\text{data}}(x_i, y_i)}[\| (x_i^c, y_i^c) - G_{\text{dec}}(\hat{c}_i) \|_1] \]

\[ L_{\text{GAN}}(G_c, D_c) = \mathbb{E}_{(x_i, y_i, c_i) \sim p_{\text{data}}(x_i, y_i, c_i)}[\log D_c(x_i^c, y_i^c, c_i)] + \mathbb{E}_{(x_i, y_i) \sim p_{\text{data}}(x_i, y_i)}[1 - \log D_c(x_i^c, y_i^c, \hat{c}_i)] \]

\[ L_{\text{GAN}}(G_{\text{dec}}, D_{\text{dec}}) = \mathbb{E}_{(x_i, y_i) \sim p_{\text{data}}(x_i, y_i)}[\frac{1}{2} \log D_{\text{dec}}(\hat{c}_i, x_i^c) + \frac{1}{2} \log D_{\text{dec}}(\hat{c}_i, y_i^c)] + \mathbb{E}_{(x_i, y_i) \sim p_{\text{data}}(x_i, y_i)}[\frac{1}{2} (1 - \log D_{\text{dec}}(\hat{c}_i, \hat{x}_i^T)) + \frac{1}{2} (1 - \log D_{\text{dec}}(\hat{c}_i, \hat{y}_i^T))] \]

and \( \hat{c}_i = G_i(x_i^T, y_i^T), \) \((x_i^c, y_i^c) = \text{STN}(x_i^\text{RAFN}, y_i^\text{RAFN}), \) \((x_i^c, y_i^c)\) are the ground-truth transposed full object inputs corresponding to \( c_i. \)

Moreover, \( L_{L_1}(G_{\text{dec}}) \) is the self-consistency constraint penalizing deviation of decomposed images from their corresponding transposed inputs and \( L_{\text{CE}} \) is the cross entropy loss applied on the predicted probability segmentation masks. We also added the gradient penalty introduced in [9] to improve convergence of the GAN loss functions. If viewpoint transformation is not needed for the objects’ domain, one can replace \( x_i^\text{RAFN} \) with \( x_i \) in the above equations. The benefit of adding decomposition networks will be specifically clarified in Section 3.5.

3.4 Unpaired Training Data

Here, we propose a variant of our model discussed in section 3.3 for broader object domains where paired inputs-outputs are not available or hard to collect. In this setting, there is not one-to-one mapping between images in sets \( X, Y \) and images in the composite domain, \( C. \) However, we still assume that foreground and segmentation masks are available for images in all three sets during training. Therefore, the background is again removed for simplicity.

Given the segmentation masks, \( M_{x_i}, M_{y_i}, \) of the joint ground-truth image, \( c_i, \) we first crop and resize object segments \( x_{i,s}^c = c_i \odot M_{x_i}, y_{i,s}^c = c_i \odot M_{y_i} \) to be at the center of the image, similar to the input center-oriented objects at test time, calling them as \( x_{i,s} \) and \( y_{i,s}. \) For each object, we add a self-supervised inpainting network [23], \( G_i, \) as a component of our compositional GAN model to generate full objects from the given segments of image \( c_i, \) reinforcing the network to learn object occlusions and spatial layouts more accurately. For this purpose, we apply a random mask on each \( x_i \in X \) to zero out pixel values in the mask region and train a conditional GAN, \( G_i^T, \) to fill in the missing regions. To guide this masking process toward a similar task of generating full objects from segmentations, we can use the background mask of images in \( Y \) for zeroing out images in \( X. \) Another cGAN network, \( G_i^T, \) should be trained similarly to fill in the missing regions of masked images in \( Y. \)

The loss function for each inpainting network would be as:

\[ L(G_i) = L_{L_1}(G_i) + \lambda L_{\text{GAN}}(G_i, D_i) \]

Therefore, starting from two inpainting networks trained on sets \( X \) and \( Y, \) we generate full object from each segment of image \( c_i \) both for the center oriented segments, \( x_{i,s} \) and \( y_{i,s}, \) and the original segments, \( x_{i,s}^c \) and \( y_{i,s}^c. \) This step is summarized in the bottom left block of Figure 1. Now, given \( x_{i,s}^c, y_{i,s}^c, c_i, \) we can train a spatial transformer network similar to the model for paired data discussed in section 3.3 followed by the composition-decomposition networks to generate composite image and its probability segmentation masks. Since we start from segmentations of the joint image rather than an input \( x_i \) from a different viewpoint, we skip training the RAFN end-to-end in the compositional network, and use its pre-trained model discussed in section 3.2 at test time.

3.5 Inference Refinement Network

After training the network, we study performance of the model on two new images, \( x, y, \) from the marginal distributions of sets \( X \) and \( Y \) along with their foreground masks to generate a natural-looking composite image containing the two objects. However, since generative models cannot generalize very well to a new example, we continue optimizing network parameters given the two input test instances to remove artifacts and generate sharper results [2]. However, the ground-truth for the composite image and the target spatial layout of the objects are not available at test time, thus the self-consistency cycle in our decomposition network provides the only supervision.
for penalizing deviation from the original objects through an $L_1$ loss. We freeze the weights of
the relative spatial transformer and appearance flow networks, and only refine the weights of the
composition-decomposition layers where the GAN loss will be still applied given the real samples
from our training set. Similar to the training time, we ignore background for simplicity given the
foreground masks of the input instances. The ground-truth masks of the transposed full input images,
$M_x^g, M_y^g$ can be also obtained by applying the pre-trained RAFN and STN on the input masks. We
then use the hadamard product to multiply the predicted masks $\hat{M}_x, \hat{M}_y$ with the objects foreground
masks $M_x^g, M_y^g$, respectively to eliminate artifacts outside of the target region for each object. One
should note that $M_x^g, M_y^g$ are the foreground masks of the full transposed objects while $\hat{M}_x$ and $\hat{M}_y$
are the predicted segmentation masks. Therefore, loss function for this refinement would be:

$$L(G) = \lambda(\|\hat{x}^T - x^T\|_1 + \|\hat{M}_x \odot \hat{c} - \hat{M}_x \odot x^T\|_1$$

$$+ \|\hat{y}^T - y^T\|_1 + \|\hat{M}_y \odot \hat{c} - \hat{M}_y \odot y^T\|_1)$$

$$+ [L_{GAN}(G_c, D_c) + L_{GAN}(G_{dec}, D_{dec})]$$

(5)

where $\hat{x}^T, \hat{y}^T$ are the generated decomposed images, and $x^T$ and $y^T$ are the transposed inputs.

In the experiments, we will present two type of generated images: (1) image generated directly from
the composition network, $\hat{c}$, before and after this refinement step, and (2) image generated directly
based on the predicted segmentation masks as $\hat{c}_a = \hat{M}_x \odot x^T + \hat{M}_y \odot y^T$.

4 Experiments

In this section, we study the performance of our proposed compositional GAN model for both the
paired and unpaired scenarios through multiple qualitative experiments in different domains. First,
we use the Shapenet dataset [5] as our main source of input objects and study two cases: (1) a chair
next to a table, (2) a bottle in a basket. Second, we show that our model performs equally well
when one object is fixed and the other one is relatively scaled and linearly transformed to generate
a composed image. We present our results on the CelebA dataset [20] composed with sunglasses
downloaded from the web. In all our experiments, the values for the training hyper-parameters are set
to $\lambda_1 = 100, \lambda_2 = 50, \lambda_3 = 100, \lambda_4 = 1$, and the inference $\lambda = 100$.

4.1 Composing a Chair with a Table

Composition of a chair and a table is a challenging problem since viewpoints of the two objects
should be similar and one object should be partially occluded and/or partially occlude the other one
depending on their viewpoint, the problem that cannot be solved by considering each object as a
separate individual layer of an image. By feeding in the two objects simultaneously to our proposed
network, the model learns to relatively transform each object and composes them reasonably.

We manually made a collection of 1000 composite images from Shapenet chairs and tables, thus our
collected dataset can be used for both the paired and unpaired scenarios. In the paired scenario, we
use the pairing information between each composite image and its constituent full chair and table
besides their foreground masks. On the other hand, to show the performance of our model on the
unpaired examples as well, we ignore the individual chairs and tables used in each composite image,
and use a different subset of Shapenet chairs and tables as real examples of each individual set. We
made sure that these two subsets do not overlap with the chairs and tables in composite images to
avoid the occurrence of implicit pairing in our experiments.

Chairs and tables in all three sets can pose in a random azimuth angle in the range $[-180^\circ, 180^\circ]$.
As discussed in section 3.2, we feed in the foreground mask of a random table with a random azimuth angle in addition to the input chair to our relative appearance flow network. Irrespective of the paired or unpaired scenarios, we use the azimuth angle information of chairs and
tables in Shapenet to train the RAFN model given the target chair in the same viewpoint as the mask
of a random table. Therefore, network learns to synthesize a chair in the viewpoint encoded in the
table’s mask. In the third column of Figure 2, we represent the synthesized test chairs as $X^{RAFN}$.

In addition, to study our network components, we visualize outputs of the model at different steps in
Figure 2. To evaluate our network with paired training data on a new input chair and table represented
as $X$ and $Y$, respectively, we find its nearest neighbor composite example in the training set in terms
of its constituent chair and table features extracted from a pre-trained VGG19 network [29]. As
Figure 2: Test results on the chair-table composition task trained with either paired or unpaired data. “NN” stands for the nearest neighbor image in the paired training set, and “NoInpaint” shows the results of the unpaired model without the inpainting network. In both paired and unpaired cases, \( \hat{c}_{\text{before}} \) and \( \hat{c}_{\text{after}} \) show outputs of the generator before and after the inference refinement network, respectively. Also, \( \hat{c}_{\text{s}} \) represents summation of masked transposed inputs after the refinement step.

shown in the fourth column of Figure 2, nearest neighbors are different enough to be certain that network is not memorizing its training data. We also illustrate output of the network before and after the inference refinement step discussed in section 3.5 in terms of the generator’s prediction, \( \hat{c} \), as well as the direct summation of masked transposed inputs, \( c_{\text{s}} \), for both paired and unpaired training models. The refinement step sharpens the synthesized image and removes artifacts generated by the model. Our results from the model trained on unpaired data is comparable with those from paired data. Moreover, we depict the performance of the model without our inpainting network in the eighth column, where occlusions are not correct in multiple examples. The second and last rows of Figure 2 emphasize that our model has successfully resolved the challenges involved in this composition task, where in some regions such as the chair handle, table is occluding the chair while in some other regions such as table legs, chair is occluding the table. We also report some of the failure cases for both paired and unpaired scenarios in Figure 6.

We have also conducted a mechanical turk evaluation to compare the performance of our algorithm in different scenarios including training with and without paired data and before and after the final inference refinement network. From a set of 90 test images of chairs and tables, we have asked 60 evaluators to select their preferred composite image generated by the model trained on paired data.
versus images generated by the model trained on unpaired data, both after the inference refinement step. As shown in Table 1, 57% of composite images generated by our model trained on paired inputs-outputs were preferred to the ones generated through the unpaired scenario. It shows that even without paired examples during training, our proposed model performs reasonably well. We have repeated the same study to compare the quality of images generated before and after the inference refinement step. As summarized in Table 1 there was a 71.3% preference to the refined images at inference time revealing the benefit of the last refinement module in generating higher-quality images.

### 4.2 Composing Bottle with a Basket

Here, we represent another compositional example from Shapenet dataset to study our network’s performance where the task would be to put a bottle in a basket. Similar to the chair-table problem, we manually composed Shapenet bottles with baskets to prepare a training set of 100 paired examples. We trained the model both with and without the paired data, similarly to section 4.1, and represent outputs of the network before and after the inference refinement in Figure 3. In addition, nearest neighbor examples in the paired training set are shown for each new input instance (third column) as well as the model’s predictions in the unpaired case without the inpainting network (seventh column). As clear from the results, our inpainting network plays a critical role in the success of our unpaired model specially for handling occlusions. This problem statement is similarly interesting since the model should identify which pixels to be occluded. For instance in the first row of Figure 3 region inside in the basket is occluded by the blue bottle while the region outside is occluding the later. More examples are shown in Figure 7. In addition, we have shown results of a Amazon Mechanical Turk study on this task, with 60 evaluators and a set of 45 test images, in the second row of Table 1 confirming the benefit of the refinement module and the comparable performance of training in the unpaired scenario with the paired training case.

**Ablation Study:** We repeat the experiments on composing a bottle with a basket, with each component of the model removed at a time, to study their effect on the final composite image. Qualitative results are illustrated in Figure 4. First and second columns show bottle and basket images which are concatenated channel-wise as the input to the network. Following columns are:

- **3rd column:** no reconstruction loss on the composite image results in wrong color and faulty occlusion,
Table 1: AMT user evaluation comparing different components of our proposed model. First column shows percentage of preferences to the images being refined during inference (versus non-refined) in the unpaired scenario. The second column shows the percentage of preferences to the refined images generated through paired training scheme compared with the unpaired case.

| Inputs          | Unpaired & (After > Before) refinement | (Paired > Unpaired) & After refinement |
|-----------------|----------------------------------------|----------------------------------------|
| Chair+Table     | 71.3%                                   | 57%                                    |
| Basket+Bottle   | 64.2%                                   | 57%                                    |

Figure 4: (a) Ablation Study: Output of our model without the component specified on top of each column. Input is the channel-wise concatenation of the bottle and basket shown in the first two columns. (b) Baselines: As the input (9th column), each bottle is added to the basket after being scaled and translated with constant parameters. Pix2Pix and CycleGAN outputs are shown on the right.

- 4th column: no cross-entropy mask loss in training results in faded bottles,
- 5th column: no GAN loss in training and inference generates outputs with a different color and lower quality than the input image,
- 6th column: no decomposition generator ($G_{dec}$) and self-consistent cycle results in partially missed bottles,
- 7, 8th columns represent full model in paired and unpaired scenarios.

Other Baselines: The purpose of our model is to capture object interactions in the 3D space projected onto a 2D image by handling their spatial layout, relative scaling, occlusion, and viewpoint for generating a realistic image. These factors distinguish our model from CycleGAN [40] and Pix2Pix [11] models whose goal is only changing the appearance of the given image. In this section, we compare to these two models. To be able to compare, we use the mean scaling and translating parameters of our training set to place each input bottle and basket together and have an input with 3 RGB channels (9th column in Figure 4). We then train a ResNet generator on our paired training data with an adversarial loss added with a $L_1$ regularizer. Since the structure of the input image might be different from its corresponding ground-truth image (due to different object scalings and layouts), ResNet model works better than a U-Net but still generating unrealistic images (10th column in Figure 4). We follow the same approach for the unpaired data through the CycleGAN model (11th column in Figure 4). As apparent from the qualitative results, it is not easy for either Pix2Pix or CycleGAN networks to learn the transformation between samples from the input distribution and that of the occluded outputs.

4.3 Composing Faces with Glasses

In this experiment, we compose a pair of sunglasses with a face image, similar to [17], where the latter should be fixed while sunglasses should be rescaled and transformed relatively. We used the CelebA dataset [20], followed its training/test splits and cropped images to $128 \times 128$ pixels. We hand-crafted 180 composite images of celebrity faces from the training split with sunglasses downloaded from
the web to prepare a paired training set. However, we could still use our manual composite set for the unpaired case with access to the segmentation masks separating sunglasses from faces. In the unpaired scenario, we used 6000 images from the training split while not overlapping with our composite images to be used as the set of individual faces during training. In this case, since pair of glasses is always occluding the face, we report results based on summation of the masked transposed inputs, $\hat{c}_s$ for both the paired training data and the unpaired one. We also compare our results with the ST-GAN model [17] which assumes faces as a fixed background and warps glasses in geometric warp parameter space. Our results both in paired and unpaired cases, shown in Figure 5, look more realistic in terms of the scale, rotation angle, and location of the sunglasses with the cost of only 180 paired training images or 180 unpaired images with segmentation masks. More example images are illustrated in Figure 8.

To confirm this observation, we have studied the results by asking 60 evaluators to score our model predictions versus ST-GAN on a set of 75 test images. According to this study where we compare our model trained on paired data with ST-GAN, 84% of the users evaluated favorably our network predictions. Moreover, when comparing ST-GAN with our unpaired model, 73% of the evaluators selected the latter. These results confirm the ability of our model in generalizing to the new test examples and support our claim that both our paired and unpaired models significantly outperform the recent ST-GAN model in composing a face with a pair of sunglasses.

5 Conclusion and Future Work

In this paper, we proposed a novel Compositional GAN model addressing the problem of object composition in conditional image generation. Our model captures the relative linear and viewpoint transformations needed to be applied on each input object (in addition to their spatial layout and occlusions) to generate a realistic joint image. To the best of our knowledge, we are among the first to solve the compositionality problem without having any explicit prior information about object’s layout. We evaluated our compositional GAN through multiple qualitative experiments for two cases of paired versus unpaired training data. In the future, we plan to extend this work toward generating images composed of multiple (more than two) and/or non-rigid objects. We believe this paper opens a new line of research in image compositionality.
Figure 6: Failure test cases for both the paired and unpaired models on the chair-table composition task.

Figure 7: More test results on the basket-bottle composition task trained with either paired or unpaired data. “NN” stands for the nearest neighbor image in the paired training set, and “NoInpaint” shows the results of the unpaired model without the inpainting network. In both paired and unpaired cases, \( \hat{c}_{\text{before}} \) and \( \hat{c}_{\text{after}} \) show outputs of the generator before and after the inference refinement network, respectively. Also, \( \hat{c}_{\text{after}} \) represents summation of masked transposed inputs after the refinement step.
Figure 8: Test examples for the face-sunglasses composition task. First two columns show input sunglasses and face images, 3rd and 4th columns show the output of our compositional GAN for the paired and unpaired models, respectively. Last column shows images generated by the ST-GAN [17] model.
A  Relative Appearance Flow Network (RAFN)

Architecture of our relative appearance flow network is illustrated in Figure 9 which is composed of an encoder-decoder set of convolutional layers for predicting the appearance flow vectors, which after a bilinear sampling generates the synthesized view. The second decoder (last row of layers in Figure 9) is for generating foreground mask of the synthesized image following a shared encoder network [38]. All convolutional layers are followed by batch normalization [10] and a ReLU activation layer except for the last convolutional layer in each decoder. In the flow decoder, output is fed into a Tanh layer while in the mask prediction decoder, the last convolutional layer is followed by a Sigmoid layer to be in the range $[0, 1]$.

![Figure 9: Relative Appearance Flow Network: Input is an image of a chair with 3 RGB channels concatenated channel-wise with the table foreground mask. Output is the appearance flow for synthesizing a new viewpoint of the chair. All layers are convolutional.](image)

B  Relative Spatial Transformer Network

Diagram of our relative spatial transformer network is represented in Figure 10. The two input images (e.g., chair and table) are concatenated channel-wise and fed into the localization network to generate two set of parameters, $\theta_1, \theta_2$ for the affine transformations. This single network is simultaneously trained on the two images learning their relative transformations required to getting close to the given target images. In this figure, orange feature maps are the output of a conv2d layer (represented along with their corresponding number of channels and dimensions) and yellow maps are the output of max-pool2d followed by ReLU. The blue layers also represent fully connected layers.

![Figure 10: Relative Spatial Transformer Network: Input is an image of a chair with 3 RGB channels concatenated channel-wise with the table RGB image. Output is two transformed images each with 3 RGB channels.](image)
References

[1] A. Antoniou, A. Storkey, and H. Edwards. Data augmentation generative adversarial networks. arXiv preprint arXiv:1711.04340, 2017.

[2] S. Azadi, M. Fisher, V. Kim, Z. Wang, E. Shechtman, and T. Darrell. Multi-content gan for few-shot font style transfer. arXiv preprint arXiv:1712.00516, 2017.

[3] O. Boiman and M. Irani. Detecting irregularities in images and in video. International journal of computer vision, 74(1):17–31, 2007.

[4] O. Boiman and M. Irani. Similarity by composition. In Advances in neural information processing systems, pages 177–184, 2007.

[5] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

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

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

[8] I. Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160, 2016.

[9] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In NIPS, 2017.

[10] S. Ioffe and C.Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

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

[12] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In NIPS, 2015.

[13] J. Johnson, A. Gupta, and L. Fei-Fei. Image generation from scene graphs. In CVPR, 2018.

[14] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196, 2017.

[15] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In CVPR, 2017.

[16] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. arXiv preprint, 2016.

[17] C.-H. Lin, E. Yumer, O. Wang, E. Shechtman, and S. Lucey. St-gan: Spatial transformer generative adversarial networks for image compositing. arXiv preprint arXiv:1803.01837, 2018.

[18] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In NIPS, 2017.

[19] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In Advances in neural information processing systems, pages 469–477, 2016.

[20] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In ICCV, 2015.

[21] M. Mirza and S. Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784, 2014.

[22] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. arXiv preprint arXiv:1610.09585, 2016.

[23] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros. Context encoders: Feature learning by inpainting. In CVPR, 2016.

[24] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In ICLR, 2016.

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

[26] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee. Learning what and where to draw. In NIPS, 2016.

[27] B. Russell, A. Efros, J. Sivic, B. Freeman, and A. Zisserman. Segmenting scenes by matching image composites. In Advances in Neural Information Processing Systems, pages 1580–1588, 2009.

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

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[30] K. Sunkavalli, M. K. Johnson, W. Matusik, and H. Pfister. Multi-scale image harmonization. In ACM Transactions on Graphics (TOG). ACM, 2010.

[31] Y.-H. Tsai, X. Shen, Z. Lin, K. Sunkavalli, X. Lu, and M.-H. Yang. Deep image harmonization. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[32] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. arXiv preprint arXiv:1711.11585, 2017.

[33] S. Xue, A. Agarwala, J. Dorsey, and H. Rushmeier. Understanding and improving the realism of image composites. ACM Transactions on Graphics (TOG), 2012.

[34] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li. High-resolution image inpainting using multi-scale neural patch synthesis. In CVPR, 2017.
[35] J. Yang, A. Kannan, D. Batra, and D. Parikh. Lr-gan: Layered recursive generative adversarial networks for image generation. arXiv preprint arXiv:1703.01560, 2017.

[36] 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 ICCV, 2017.

[37] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In ECCV, 2016.

[38] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros. View synthesis by appearance flow. In ECCV, 2016.

[39] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Learning a discriminative model for the perception of realism in composite images. In ICCV, 2015.

[40] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In ICCV, 2017.