

Shape-conditioned Image Generation by Learning Latent Appearance Representation from Unpaired Data^{*}

Yutaro Miyauchi¹[0000-0002-5951-3172], Yusuke Sugano¹[0000-0003-4206-710X], and Yasuyuki Matsushita¹[0000-0002-1935-4752]

Graduate School of Information Science and Technology, Osaka University, Japan
{miyauchi.yutaro,sugano,yasumat}@ist.osaka-u.ac.jp

Abstract. Conditional image generation is effective for diverse tasks including training data synthesis for learning-based computer vision. However, despite the recent advances in generative adversarial networks (GANs), it is still a challenging task to generate images with detailed conditioning on object shapes. Existing methods for conditional image generation use category labels and/or keypoints and are only give limited control over object categories. In this work, we present SCGAN, an architecture to generate images with a desired shape specified by an input normal map. The shape-conditioned image generation task is achieved by explicitly modeling the image appearance via a latent appearance vector. The network is trained using unpaired training samples of real images and rendered normal maps. This approach enables us to generate images of arbitrary object categories with the target shape and diverse appearances. We show the effectiveness of our method through both qualitative and quantitative evaluation on training data generation tasks.

1 Introduction

Generating realistic images is a central task in both computer vision and computer graphics. Despite recent advances in generative adversarial networks (GANs), it is still challenging to fully control how the target object should appear in the output images. There have been several approaches for conditional image generation which introduce additional conditions to GANs such as class labels [23,31] and keypoints [21,25]. However, previous approaches still suffer from an inability to control detailed object shapes and lack generalizability to arbitrary object categories.

Training data synthesis is one of the most promising applications of conditional image generation. Since recognition performance of machine learning-based methods heavily depends on the amount and quality of training images, there is an increasing demand for methods and datasets for training recognition models using synthetic data [33,24,26,12]. However, when synthetic training images are rendered with off-the-shelf computer graphics techniques, the trained estimators still suffer from an appearance gap from actual, often degraded test images. GANs have also

^{*} Yusuke Sugano is supported by and JST CREST Grant Number JPMJCR1781, and Yasuyuki Matsushita is supported by the New Energy and Industrial Technology Development Organization (NEDO), Japan.

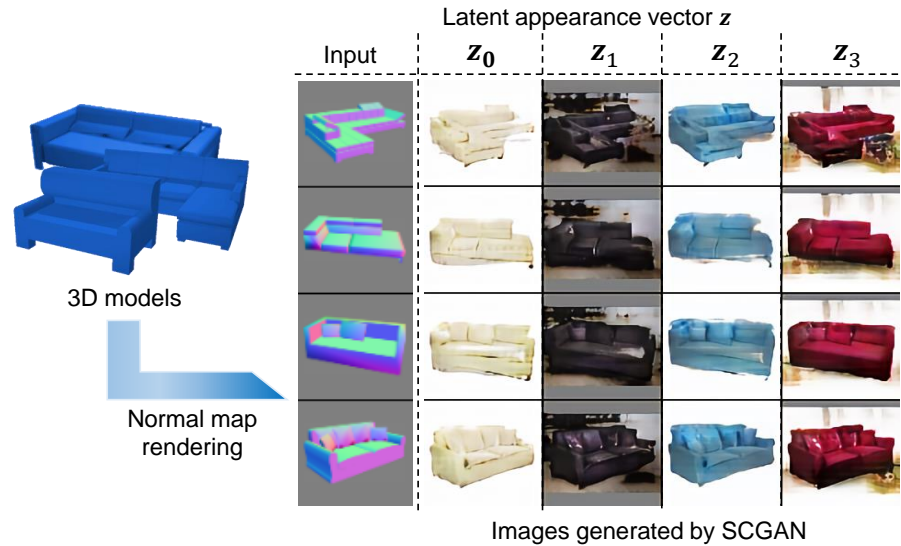


Fig. 1: The proposed shape-conditioned image generation network (SCGAN) outputs images of an arbitrary object with the same shape as the input normal map, while controlling the image appearances via latent appearance vectors.

been used to modify synthetic data to more realistic training images, and it has been shown that such data can improve the performance of learned estimators [27,28,2]. These methods use synthetic data as a condition on image generation so that output images remain visually similar to the input images and therefore keep their original ground-truth labels. In this sense, the aforementioned limitation of conditional image generation severely restricts the application of such training data synthesis approaches. If the method allows for more fine-grained control of object shapes, poses, and appearances, it can open a way for generating training data for, *e.g.*, generic object recognition and pose estimation.

In this work, we propose SCGAN (Shape-Conditioned GAN), a GAN architecture for generating images conditioned by input 3D shapes. As illustrated in Fig. 1, the goal of our method is to provide a way to generate images of arbitrary objects with the same shape as the input normal map. The image appearance is explicitly modeled as a latent vector, which can be either randomly assigned or extracted from actual images. Since we cannot always expect paired training data of normal maps and images, the overall network is trained using the cycle consistency loss [39] between the original and back-reconstructed images. In addition, the proposed architecture employs an extra discriminator network to examine whether the generated appearance vector follows the assumed distribution. Unlike prior work using a similar idea for feature learning [7], this appearance discriminator allows us to not only control the image appearance, but also to improve the quality of generated images. We demonstrate the effectiveness of our method in comparison with baseline approaches through

qualitative analysis of generated images, and quantitative evaluation of training data synthesis performance on appearance-based object pose estimation tasks.

Our contributions are twofold. First, to the best of our knowledge, we present the first GAN architecture which uses normal maps as the input condition for image generation. This provides a flexible and generic way for generating shape-conditioned images without relying on any assumption on the target object category. Second, through experiments, we show that the proposed method allows us to generate training data for appearance-based object pose estimation, with better performances than synthetic data generated by baseline GAN architectures.

2 Related work

Our method aims at generating shape-conditioned images with realistic appearances, related to prior methods on conditional image generation GANs. One of the potential applications of our method is generating realistic training data, and hence our method is further related to methods applying GANs for bridging the gap between synthetic training data and real images.

2.1 GANs for Conditional Image Generation

Generative Adversarial Networks (GANs) have made considerable advances in recent years [9,22,18,17], and have been successfully applied to various tasks such as image super-resolution [20,16], inpainting [15], and face aging [37]. GANs consist of mainly two networks, generator and discriminator, which are trained in an adversarial manner. The generator generates images so that they are recognized as real ones, while the discriminator learns to discriminate generated images from real images from a training dataset. The generator usually receives a vector of random numbers sampled from an arbitrary probability distribution as input, and outputs an image through the network. However, as discussed earlier, most of the standard GAN architectures do not allow for fine-grained control of the output images.

To address this limitation, much research has been conducted on GAN architectures for conditional image generation. There have been several approaches to use class labels as a condition on generated images and to specify which object category to be drawn in the output image [23]. Similarly, some prior work proposed to control the generated images by conditioning them on human-interpretable feature vectors built in an unsupervised manner [5,29]. To increase the flexibility of image generation, some works further used input features indicating where and how the target object should be drawn, such as bounding box [25] and keypoints [21]. Alternatively, iGAN [38] and the Introspective Adversarial Network [3] take an approach to use user drawings as a condition for image generation. However, the conditions used in these methods still have a limitation that precise 3D shape control is only possible with specific object categories with hand-designed keypoint locations. In contrast, our method allows for direct control on arbitrary object shapes using normal map rendering, without requiring paired training data.

2.2 Learning with Simulated/Synthesized Images

Due to the limited availability of fully-labeled training images for diverse computer vision tasks, there is an increasing attention on synthetic training data. Computer graphics pipelines have been employed to synthesize images with desired ground-truth labels. Such a learning-by-synthesis approach is especially efficient for tasks whose ground-truth labels require costly manual annotation, such as semantic segmentation [24,26] and eye gaze estimation [33]. However, synthetic images still suffer from a large gap from real images in terms of object appearance and often degraded imaging quality, and hence the learned estimator cannot directly achieve desired performance on real-world input images.

To fill the gap between training (synthetic) and test (real) image domains, there have been proposed many domain adaptation techniques. In addition to research attempts on the learning process [36,30,32,1,8], GANs have been also shown as promising tools for bridging the domain gap. Shrivastava *et al.* proposed the SimGAN that modifies the input synthetic images to be visually similar to real images, and showed that such an approach improves the baseline performances on tasks like hand pose and gaze estimation [27]. RenderGAN [28] takes a similar approach to convert simple barcode-like input images into realistic images. CycleGAN [39] architecture provides a way to mutually convert images from two different domains without requiring paired images, and also be applicable to the domain adaptation task. Bousmalis *et al.* proposed the pixel-level domain adaptation (PixelDA) approach which transfers source images to the target domain under the pixel-level similarity constraint. Essentially, synthetic images were used as a strong constraint on output images in these methods, and GANs were restricted only to modify the imaging properties of the target object. In contrast, since our method uses texture-less normal maps to provide purely shape-related information to the generator, it allows for a full flexibility to control object and background appearances.

3 SCGAN: Shape-conditioned Image Generation Network

The goal of SCGAN is to generate images of arbitrary object categories, with the same shape as the input normal map. While the training process requires an access to both normal maps and real images of the target object, in practice it is almost impossible to assume paired training data. To this end, SCGAN adopts the idea of cycle-consistency loss [39] and the whole network is trained using unpaired training images. Furthermore, to maximize the flexibility of object appearances, the image generator also takes an appearance vector as input, in addition to the normal map. By training the network so that appearance-related information is represented only with the appearance vector, our method realizes the shape-conditioned image generation task more efficiently and accurately.

3.1 Network Architecture

As illustrated in Fig. 2, the proposed architecture consists of five convolutional neural networks. G_I is an image generator that takes an appearance vector $\mathbf{z} \in \mathbb{R}^n$ and

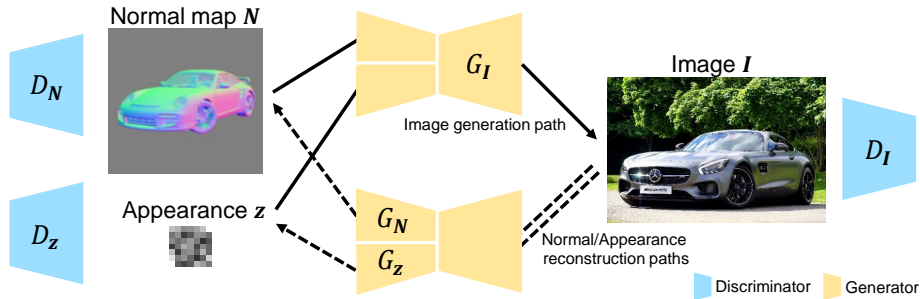


Fig. 2: Overview of the proposed architecture. G_I, G_N , and G_z are generators, and D_I, D_N , and D_z are discriminators. G_N and G_z share weights at the first few layers, and simultaneously output N and z from the input real image I .

a normal map $N \in \mathbb{R}^{m \times m}$ as input, and outputs an image $I \in \mathbb{R}^{m \times m}$. Conversely, G_N and G_z correspond to the normal map and appearance vector generators with partially shared network weights that converts an image I to an appearance vector z and a normal map N . Each data modality has their own discriminators D_I, D_N , and D_z . While D_I and D_N judge whether the input image and normal map are real or generated, D_z judges whether the input appearance vector is the one sampled from a Gaussian distribution or not.

As described earlier, the proposed network is designed to be trained on unpaired training samples using the cycle-consistency loss [39]. While the main goal of our approach is to train the image generator G_I , normal map and appearance generators (G_N and G_z) are also trained and used to back-reconstruct each modality and compare with the original input. However, if we only consider generators and discriminators of images and normal maps, generators tend to satisfy the cycle-consistency loss by *embedding* hidden information to intermediate data. For example, if the image generator learns to embed input information to the output image, the normal map generator can recover the original normal map without taking into account the object shape in the intermediate image. To avoid such situations, we also enforce the network to learn to separate shape and appearance information by introducing the appearance generator and discriminator. The proposed network effectively generates shape-conditioned images by modeling the appearance variation in the training data as a Gaussian appearance vector, while also allowing us to explicitly sample appearance information from actual images using the appearance generator G_z .

3.2 Training Loss

We train discriminators and generators using the WGAN-GP loss [10] which is based on the Wasserstein-1 distance between real and generated data distributions. The loss functions L_d and L_g for discriminators and generators, respectively, are defined

as:

$$\begin{aligned} L_d(D) &= \mathbb{E}_{\mathbf{x} \sim P_{\mathbf{x}}} [D(G(\mathbf{x}))] - \mathbb{E}_{\hat{\mathbf{x}} \sim P_{\hat{\mathbf{x}}}} [D(\hat{\mathbf{x}})] + \lambda_p \mathbb{E}_{\hat{\mathbf{x}} \sim P_{\hat{\mathbf{x}}}} [(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2], \\ L_g(G) &= -\mathbb{E}_{\mathbf{x} \sim P_{\mathbf{x}}} [D(G(\mathbf{x}))], \end{aligned} \quad (1)$$

where \mathbf{x} is real data (image, normal map, appearance vector), $\hat{\mathbf{x}}$ is generated data from their corresponding generators, and $\hat{\mathbf{x}} = \epsilon G(\mathbf{x}) + (1-\epsilon)\hat{\mathbf{x}}$. $\epsilon \sim \mathcal{U}[0, 1]$ is random-weighted sum of input and generated data. $P_{\mathbf{x}}$, $P_{\hat{\mathbf{x}}}$, and $P_{\hat{\mathbf{x}}}$ indicate distributions of each data, and \mathbb{E} represents the mean of the distribution. The third term of Eq. (1) has an effect of stabilizing the adversarial training [10].

In our implementation, while three discriminators are trained using the individual discriminator losses, all generators are jointly trained as:

$$G_I, G_N, G_z = \arg \min_{G_I, G_N, G_z} L(G_I, G_N, G_z),$$

where $L(G_I, G_N, G_z)$ is the joint loss function also taking into account the cycle-consistency losses:

$$\begin{aligned} L(G_I, G_N, G_z) &= L_g(G_I) + L_g(G_N) + L_g(G_z) + \lambda_N \|\mathbf{N} - G_N(G_I(\mathbf{N}, \mathbf{z}))\|_F^2 \\ &\quad + \lambda_I \|\mathbf{I} - G_I(G_N(\mathbf{I}), G_z(\mathbf{I}))\|_F^2 + \lambda_z \|\mathbf{z} - G_z(G_I(\mathbf{N}, \mathbf{z}))\|_2^2. \end{aligned} \quad (2)$$

λ_I, λ_N , and λ_z are weights for each cycle-consistency loss term which are defined as the distance between the input and the back-reconstructed output. These weights are required to take balance between discriminator and cycle-consistency losses in each domain, and they control how strictly the model should maintain the input shapes. Image \mathbf{I} and normal map \mathbf{N} are sampled from the distribution of real data P_I and P_N , and \mathbf{z} are an appearance vector sampled from a zero-mean Gaussian distribution $\mathcal{N}(0, \sigma^2)$.

3.3 Implementation Details

Figure 3 shows the details of generator/discriminator networks. The architecture of the generator network follows Zhu *et al.* [39] and the network mainly consists of convolution (*Convolution-Pixelwise normalization-ELU*) block, deconvolution (*Deconvolution-Pixelwise normalization-ELU*) block, and ResNet block [11]. As described earlier, parameters of the first six convolution blocks of the normal map generator G_N and the appearance vector generator G_z are shared. The discriminator network for images and normal maps also consists of the convolution block and outputs a scalar value indicating discrimination results through a fully connected layer. The appearance discriminator network consists of a fully connected layer followed by a *Instance normalization-ELU* block, and also outputs a scalar value through a fully connected layer. The size of input images and normal maps are set to 64×64 pixels, and is downsampled to 16×16 before the ResNet blocks.

During training, each discriminator was trained independently with respect to the corresponding discriminator losses. Then the generators were trained as Eq. (2),

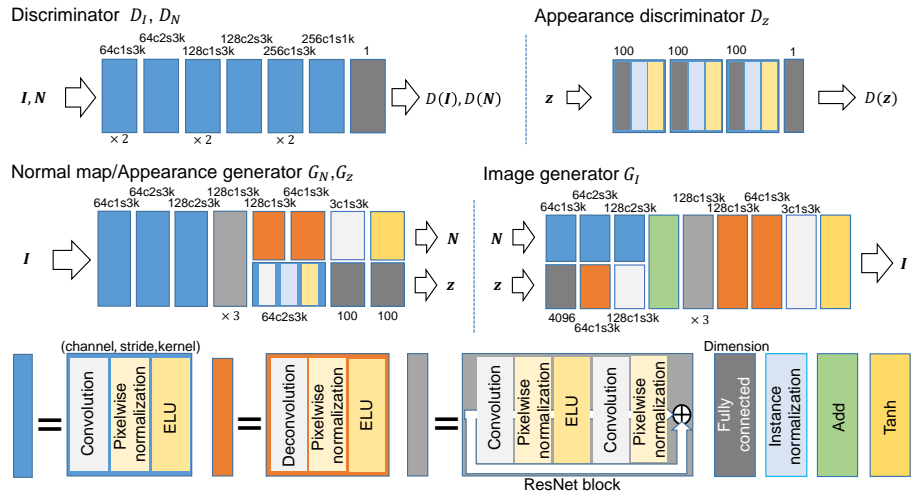


Fig. 3: Details of the generator/discriminator networks. N , I , and z indicate normal map, real image, and appearance vector. Parameters of the convolutional layers are indicated as $CcSsKk$, *i.e.*, a feature map is convolved into C channels with stride S and kernel size K .

with the discriminator parameters fixed. Following [10], discriminators were updated five times as much as generators. The networks are optimized using the Adam algorithm [19] ($\alpha = 0.001, \beta_1 = 0.5, \beta_2 = 0.9$), with the batch size of 16. We fixed hyperparameters in Eq. (2) as $\lambda_I = 1.0, \lambda_N = 1.0$, and $\lambda_z = 10.0$. The variance of the Gaussian distribution was set to $\sigma^2 = 0.5$.

4 Experiments

We demonstrate the performance of the proposed SCGAN architecture through both qualitative analysis and quantitative evaluation. As a qualitative analysis, we compare shape-conditional generated images from the proposed method and other baseline methods in terms of both accuracies of object shape and diversity of object appearances. In addition, we show some ablation studies to analyze the efficiency of the proposed network design. As a quantitative evaluation, we further compare the performance of appearance-based object pose estimator using these generated images from different methods as training data.

Training Datasets In both qualitative and quantitative experiments, we take three object classes as examples: cars, sofas, and chairs. Table 1 shows details of the training datasets. Each dataset consists of both real images and normal maps. Real images were sampled from the LSUN dataset [35] with a simple filtering process to select images showing a single and sufficiently large target object. Using a pre-trained

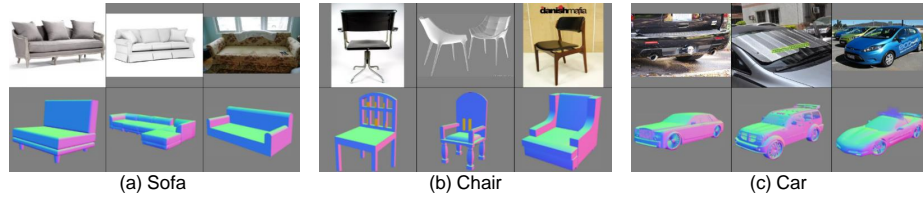


Fig. 4: Examples of the training data taken from the LSUN dataset [35] and the ShapeNet dataset [4]. The top row is real images from the LSUN dataset after post-processing, and the bottom row is normal maps rendered using models from the ShapeNet dataset.

Table 1: Training detail about the number of real data, 3D model, normal map, and the range of camera angles. When azimuth and elevation are 0, it means the camera is located in front of the object. The camera moves in increments of 5 degrees.

	Num. images	Num. 3D models	Camera angle [degrees]			Num. normal maps
			Azimuth	Elevation	Num. angles	
Sofa	83,765	3,173	-45~45	10~25	36	114,228
Car	386,370	3,385	-90~90	0~15	76	257,260
Chair	151,758	6,778	-90~90	10~25	76	515,128

object detector [14], we accepted images with only one bounding box of the target class whose area is larger than 25% of the whole image. After the filtering process, there were in total 83,765, 151,758, and 386,370 images for sofa, chair, and car, respectively. These images were extended to 1:1 aspect ration by filling the borders by zero padding. Figures 4 (a), (b), and (c) show samples of the sofa, chair, and car images used for training. The top row is real images from the LSUN dataset after post-processing, and the bottom row is normal maps rendered using models from the ShapeNet dataset. As can be seen in the cases such as the top-middle example in Figs. 4 (b) and the top-left example in (c), the real images still contain some occlusions and mismatched object poses compared to the normal maps even after the automatic filtering, which illustrates the fundamental difficulty of handling unpaired data.

We used 3D models taken from the ShapeNet dataset [4] to render normal maps. Using 3,173, 6,778, and 3,385 models for sofa, chair, and car, the normal maps were rendered so that the pose distribution roughly resembles the real image dataset. Table 1 lists the ranges of camera poses for each object, where the virtual camera was placed with increments of 5 degrees. In total, there were 114,228, 515,128, and 257,260 normal maps for sofa, chair, and car. Since the position of the object also differs in the real images, during training we also applied random shifting and scaling to these normal maps.



Fig. 5: Comparison with SimGAN and PixelDA for each object class. In each figure, the first row is input normal maps. The second and third rows are output from SCGAN and SimGAN using these normal maps.

Baseline Methods Although there is no other method directly addressing the same task of shape-conditioned image generation, we picked two closely related approaches as baseline methods: SimGAN [27] and pixel-level domain adaptation (PixelDA) network [2]. The network architectures, discriminator losses, and training hyper-parameters of these baseline methods were set to the same as our method (SCGAN) for fair comparison, while method-specific losses stayed the same as the original papers. Following the original method, SimGAN does not have the input appearance vector and there is no mechanism to change the appearance of generated images. Since these methods were designed to modify rendered images of textured 3D models, we also evaluated them using textured rendering as input condition. The textured images were rendered in the same settings as the normal maps.

4.1 Comparison of Generated Images

Figure 5 shows examples of generated images from each method. Figures 5 (a), (b), and (c) correspond to the cases of sofa, chair, and car, respectively. In each figure, the first row shows the input normal maps, and the second and third rows show the output from SCGAN and SimGAN using these normal maps as input.

It can be seen that SCGAN generates more naturalistic images than baseline methods. SimGAN could not successfully modify normal maps and failed to generate



Fig. 6: Examples of generated images using the same normal map with several different appearance vectors. Each image shows the input normal map and textured image in the first column, the rest shows generated images with SCGAN, SimGAN, and PixelDA.

realistic images in most cases. In addition, there are many cases where the baseline methods failed to generate realistic background in Fig. 5 (a). This illustrates the advantage of our method which does not rely on a strong constraint unlike baseline methods minimizing the distance between the generated and input images.

Figure 6 further shows more output examples of SCGAN, using the same normal map but with different appearance vectors. Figures 6 (a), (b), and (c) are the input normal map and generated images of sofa, chair, and car, respectively. In each figure, the first column shows the input normal map and textured image. The remaining first row shows the generated images from the normal map using SCGAN, and the second row shows the output images from the textured image using both SimGAN and PixelDA. Since SimGAN cannot control the output image appearance, it only shows one example. While the baseline methods cannot control object shapes separately from the appearance, SCGAN can generate images with the same shape and diverse appearances. It is noteworthy that in Fig. 6 (a) the output images also keep the cushion placed on the sofa, which is not an easy case for keypoint-based methods.

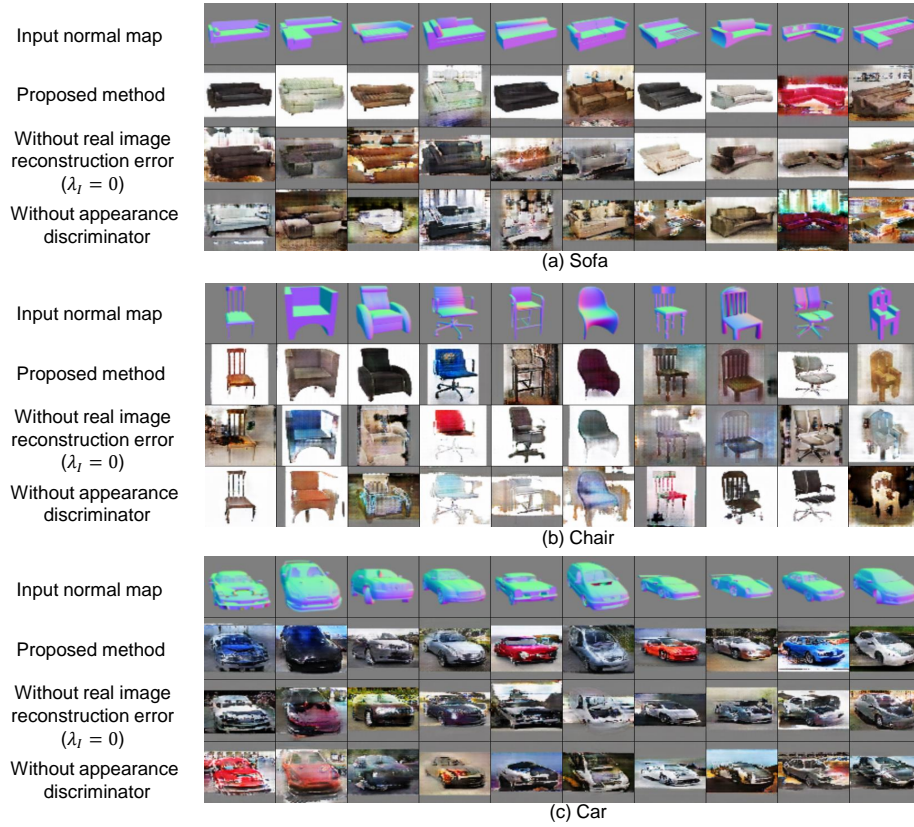


Fig. 7: Generated images without real image reconstruction error and appearance discriminator. The first rows show input normal maps, and the rest shows output images generated by SCGAN, without the real image reconstruction loss, without the appearance discriminator loss.

Ablation Study In Fig. 7, we further show the effectiveness of individual loss terms in Eq. (2). To demonstrate the effect of the proposed architecture using the separate appearance modeling and the cycle-consistent real image reconstruction loss, we evaluated models trained without real image reconstruction error and appearance discriminator. Figures 7 (a), (b), and (c) correspond to the cases of sofa, chair, and car, respectively. In each figure, the first row shows the input normal maps. The second row shows the output using all losses in Eq. (2). The third row corresponds to the training result without the real image reconstruction error (λ_I was set to zero), and the fourth row corresponds to the case trained without the appearance discriminator.

These examples show that the proposed approach improves the overall image quality by using these losses. The real image reconstruction error significantly con-



Fig. 8: Generated images with sampled appearances from real images. The first and second rows are input source images for appearance vectors and normal maps, and the third row shows the generated images.

tributes to the realism of generated images, and the results without image reconstruction error mostly failed to generate object appearances. When the network was trained without the appearance discriminator, the generated images sometimes become highly distorted as can be seen in middle columns of Fig. 7 (a).

Appearance Representation As a consequence of the cycle-consistent training, the appearance generator G_z can also be used to extract appearance vectors from real images for generating new images. Figure 8 shows some examples of images generated using appearances sampled from real images. Figures 8 (a), (b), and (c) correspond to the cases of sofa, chair, and car, respectively. As can be seen in these examples, SCGAN can generate shape-conditioned images with the similar appearance with the source images. This illustrates the potential of SCGAN for modifying pose and shape of objects in existing images.

Handling Unknown 3D Shapes Another advantage of our method is that it can take an arbitrary normal map as input, even ones rendered using hand-crafted objects. In Fig. 9, we further show the output from the sofa image generator using hand-crafted sofa objects and shapes from the other object classes. The hand-crafted models were created by a person who has never experienced 3D modeling, and consists of basic 3D shapes without any texture.

Each block corresponds to the result of one 3D model, with the same three appearances. The first rows are input normal maps, and the second rows are generated images. Even when the object shape is significantly different from ordinary sofa shapes, SCGAN successfully generates their corresponding images. As can be seen in the bottom-right blocks, the proposed method tries to map the object texture to the input shape even when the shape comes from completely different object categories.

4.2 Training Data Generation for Object Pose Estimation

Since SCGAN also keeps the object pose as the same as the input normal maps, the generated images can serve as a training data for appearance-based object pose estimation. In this section, we compare the effectiveness of SCGAN as training data

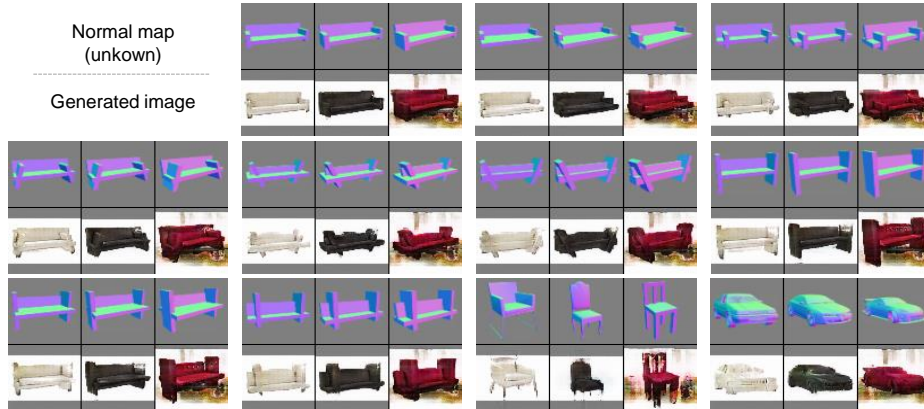


Fig. 9: Examples of generated images from unknown 3D shapes. Each block corresponds to one 3D model. The first rows are input normal maps, and the second rows are generated images.

generation framework, by comparing the accuracy of trained pose estimator with the cases using generated images from baseline methods. The architecture of the pose estimation network follows the DenseNet [13], while the last fully connected layer is modified to output 3-dimensional pose parameters. The network weights were pre-trained on the ImageNet [6], and the whole network including the last layer was trained on each target object dataset. Object poses are represented as Euler angles (azimuth, elevation, theta), and the loss function is set to be the Euclidean distance between ground-truth and estimated poses.

Test data were taken from the ObjectNet3D dataset [34] which consists of images annotated with pose-aligned 3D models. We selected images with the corresponding object annotations, and whose object poses stay within the pose range set for the training data. In total, we used 886, 2,547, and 3,939 test images for sofas, chairs, and cars, respectively.

We compare the performance of the pose estimator with the ones trained using data generated by SimGAN [27] and PixelDA [2]. As in the training of image generators, random shifting and scaling were also applied to the normal maps. As an indicator of the upper-bound accuracy of this task, we also trained the same pose estimator using the test data via 5-fold cross-validation. In addition, we evaluated the pose estimator directly using the textured images to show the estimator performance without any domain adaptation. Similarly, to show the baseline performance of each task we also evaluated a naïve estimator which always output the mean pose in each object category.

Table 2 lists pose estimation errors for each method and object category. The estimation error was evaluated as the geodesic distance between the ground-truth rotation matrix \mathbf{R}_t and the estimated rotation matrix \mathbf{R} as $\frac{1}{\sqrt{2}}\|\log(\mathbf{R}^T\mathbf{R}_t)\|$ [34]. The first column (*Target data*) shows the upper-bound performance obtained via cross-

Table 2: Mean pose error for ObjectNet3D dataset when the pose estimator is trained using dataset generated by SimGAN, PixelDA, and textured images. Naïve baseline means that all predictions are an average of the target data.

	Target data	Normal map		Textured			Naïve baseline
		SCGAN	SimGAN [27]	SimGAN [27]	PixelDA [2]	No op.	
Sofa	21.1	23.9	27.1	23.8	24.9	28.7	30.0
Chair	21.5	26.0	33.8	35.2	28.2	41.3	47.9
Car	16.9	18.2	26.0	22.4	20.4	33.6	38.9

validation. The second and third columns show the result using the dataset generated from normal maps, with SCGAN and SimGAN, respectively. Similarly, the third and fourth columns show the result using the dataset generated from textured images, with SimGAN and PixelDA, respectively. The fifth column (*No op.*) additionally shows the result directly using the original textured images. The sixth column shows the naïve baseline performance of the average predictor.

The result shows that SCGAN achieved better pose estimation performances than SimGAN-based training results using normal maps, and better or close performance in comparison with SimGAN and PixelDA based training using textured 3D model images. SCGAN significantly improved the pose estimation performance especially in the case of the chair dataset. This is mainly because chair images have larger appearance gaps from the textured images, and SCGAN successfully generated training images closer to the actual test images.

5 Conclusion

In this work, we proposed SCGAN, a GAN architecture for shape-conditioned image generation. Given a normal map of the target object category, SCGAN generates images with the same shape as the input normal map. The network can be trained without relying on paired training data with cycle-consistency losses, and it is able to generate images with diverse appearances through the latent modeling of image appearances. Unlike prior work on conditional image generation, our method does not rely on any object-specific keypoint design and can handle arbitrary object categories. The proposed method therefore provides a flexible and generic framework for shape-conditioned image generation tasks.

We demonstrated the advantage of SCGAN through both qualitative and quantitative evaluations. SCGAN not only improves the quality of generated images while maintaining the input shape, but also efficiently handles the training data synthesis task for appearance-based object pose estimation. In future work, we will further investigate applications of the proposed method including a wider range of learning-by-synthesis approaches, together with more detailed human evaluation on generated images.

References

1. Baktashmotlagh, M., Harandi, M.T., Lovell, B.C., Salzmann, M.: Unsupervised Domain Adaptation by Domain Invariant Projection. Proc. ICCV pp. 769–776 (2013)
2. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. Proc. CVPR pp. 95–104 (2017)
3. Brock, A., Lim, T., Ritchie, J.M., Weston, N.: Neural Photo Editing with Introspective Adversarial Networks. Proc. ICLR (2017)
4. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository. arXiv preprint arXiv:1512.03012 (2015)
5. Chen, X., Duan, Y., Houthoofd, R., Schulman, J., Sutskever, I., Abbeel, P.: InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. Proc. NIPS , 1–14 (2016)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. Proc. CVPR pp. 248–255 (2009)
7. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial Feature Learning. Proc. ICLR (2017)
8. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-Adversarial Training of Neural Networks. JMLR pp. 1–35 (2015)
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M.: Generative Adversarial Networks. Proc. NIPS pp. 2672–2680 (2014)
10. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved Training of Wasserstein GANs. Proc. NIPS pp. 5769–5779 (2017)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. Proc. CVPR pp. 770–778 (2016)
12. Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model Based Training, Detection and Pose Estimation of Texture-less 3D Objects in Heavily Cluttered Scenes. Proc. ACCV pp. 548–562 (2012)
13. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely Connected Convolutional Networks. Proc. CVPR pp. 2261–2269 (2017)
14. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K.: Speed/accuracy Trade-offs for Modern Convolutional Object Detectors. Proc. CVPR pp. 3296–3305 (2017)
15. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and Locally Consistent Image Completion. ACM Transactions on Graphics (13), 1–14 (2017)
16. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual Losses for Real-Time Style Transfer and Super-Resolution. Proc. ECCV pp. 694–711 (2016)
17. Junbo Zhao, M.M., LeCun, Y.: ENERGY-BASED GAN. Proc. ICLR pp. 32–48 (2015)
18. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive Growing of GANs for Improved Quality, Stability, and Variation. Proc. ICLR pp. 1–25 (2018)
19. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. Proc. ICLR (2015)
20. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. ACM Multimedia pp. 4681–4690 (2016)
21. Ma, L., Jia, X., Sun, Q., Schiele, B., Tuytelaars, T., Van Gool, L.: Pose Guided Person Image Generation. Proc. NIPS pp. 405–415 (2017)
22. Mao, X., Li, Q., Xie, H., Lau, R.Y.K., Wang, Z., Smolley, S.P.: Least Squares Generative Adversarial Networks. Proc. ICCV (nov 2017)
23. Odena, A., Olah, C., Shlens, J.: Conditional Image Synthesis With Auxiliary Classifier GANs. Proc. ICML (2017)

24. Qiu, W., Yuille, A.: UnrealCV: Connecting Computer Vision to Unreal Engine. Proc. ECCV pp. 909–916 (2016)
25. Reed, S., Akata, Z., Mohan, S., Tenka, S., Schiele, B., Lee, H.: Learning What and Where to Draw. Proc. NIPS pp. 217–225 (2016)
26. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. Proc. CVPR, 3234–3243 (2016)
27. Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R.: Learning from Simulated and Unsupervised Images through Adversarial Training. Proc. CVPR p. 6 (2017)
28. Sixt, L., Wild, B., Landgraf, T.: RenderGAN: Generating Realistic Labeled Data. *Frontiers in Robotics and AI* p. 66 (2018)
29. Springenberg, J.T.: Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks. Proc. ICLR (2016)
30. Sun, B., Saenko, K.: From Virtual to Reality: Fast Adaptation of Virtual Object Detectors to Real Domains. Proc. BMVC pp. 82.1–82.12 (2014)
31. Tan, W.R., Chan, C.S., Aguirre, H., Tanaka, K.: ArtGAN: Artwork Synthesis with Conditional Categorical GANs. Proc. ICIP p. 10 (2017)
32. Vazquez, D., Lopez, A.M., Marin, J., Ponsa, D., Geronimo, D.: Virtual and Real World Adaptation for Pedestrian Detection. IEEE TPAMI pp. 797–809 (2014)
33. Wood, E., Baltrušaitis, T., Morency, L.P., Robinson, P., Bulling, A.: Learning an Appearance-based Gaze Estimator from One Million Synthesised Images. ACM Symposium on Eye Tracking Research & Applications pp. 131–138 (2016)
34. Xiang, Y., Kim, W., Chen, W., Ji, J., Choy, C., Su, H., Mottaghi, R., Guibas, L., Savarese, S.: Objectnet3D: A large scale database for 3D object recognition. Proc. ECCV pp. 160–176 (2016)
35. Yu, E., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop. arXiv preprint arXiv:1506.03365 (2015)
36. Zhang, Y., David, P., Gong, B.: Curriculum Domain Adaptation for Semantic Segmentation of Urban Scenes. Proc. ICCV pp. 2039–2049 (2017)
37. Zhang, Z., Song, Y., Qi, H.: Age Progression / Regression by Conditional Adversarial Autoencoder. Proc. CVPR pp. 5810–5818 (2017)
38. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. Proc. ECCV pp. 597–613 (2016)
39. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. Proc. ICCV (2017)