PartGAN: Weakly-supervised Part Decomposition for Image Generation and Segmentation

Yuheng Li[1]
li2464@wisc.edu
Krishna Kumar Singh[2]
krishsin@adobe.com
Yang Xue[2]
ayxue@ucdavis.edu
Yong Jae Lee[1]
yongjaelee@cs.wisc.edu

[1] University of Wisconsin–Madison
Madison, WI, USA
[2] University of California-Davis
Davis, CA, USA

Abstract

We propose PartGAN, a novel generative model that disentangles and generates background, object shape, object texture, and decomposes objects into parts without any mask or part annotations. To achieve object-level disentanglement, we build upon prior work and maximize the mutual information between the generated factors and sampled latent prior codes. To achieve part-level decomposition, we learn a part generator, which decomposes an object into parts that are spatially localized, disjoint, and consistent across instances. Extensive experiments on multiple datasets demonstrate that PartGAN discovers consistent object parts, which enable part-based controllable image generation.

1 Introduction

Consider Fig. 1. Prior disentangled image generation work [23, 25, 37] can take the object texture from one image (B) and combine it with the object shape and background of another (A) to generate a new image (left). But what if we want a model which can take the texture of a specific object *part* like the bird's head? Such a part decomposition and generation model would be valuable for a number of different applications, including (1) visual recognition tasks since parts are a robust representation for dealing with occlusion and changes in camera viewpoint [18] as well as for recognizing localized object details for fine-grained category recognition [34, 51]; (2) data augmentation, e.g., to improve a model's invariance to the appearance changes of an object at the part-level; and (3) artistic applications like swapping clothing items [28], if the discovered parts are semantically meaningful.

In recent years, several unsupervised/weakly-supervised generative models [25, 28, 37, 40, 45] have been proposed to disentangle and model different factors of variation for image generation. For example, MixNMatch [25] can disentangle and combine object shape, texture, pose, and background from real images to generate a new image, as shown in Fig. 1

Figure 1: **(Left)** Prior disentanglement work (e.g., [25]) can only disentangle and transfer appearance at the full object-level. **(Right)** Our model, PartGAN, can disentangle and transfer texture at the *part*-level without any object mask or part annotations.

('Prior work'). However, a common limitation is that these methods cannot achieve *part-level* control. This means that if one wants to change the texture of only one part of an object, like the bird's head, it is not possible to do with such methods. The only exception is the approach of [28], which can disentangle an object's shape and texture at the part-level. However, it has two key limitations: (1) it uses a Gaussian distribution to model each part's spatial location, and (2) it cannot disentangle foreground from background. These two limitations impede its performance in part decomposition as demonstrated in our experiments.

In this paper, our goal is to learn a generative model that can generate images with part-level control with only object-level bounding box annotations and without any part or object mask supervision.[1] Importantly, we want our model to learn *detailed pixel-level masks* for each discovered part, rather than Gaussian distributions, and to disentangle the foreground from background. There are two key reasons why we believe a mask is better than a Gaussian distribution for part representation. First, a mask can provide precise pixel-level boundaries of parts while a Gaussian distribution is only able to provide a blurry description of part shape, such as its location (mean) and rough extent (variances). Second, a Gaussian representation is not suitable for describing all shapes, especially those that are non-convex; for example, it is not appropriate to use a Gaussian distribution to model the shape of a bird's head with its pointed beak, which with a mask, in contrast, can be precisely outlined. These reasons make a mask representation better at preserving the part shape when transferring its texture from one image to another (e.g., for conditional image generation). Moreover, we desire disentanglement of foreground from background because we can obtain better and more consistent object part decomposition by avoiding interference with the background. This can be especially useful when we want to only transfer the object texture from one image to another without changing the background.

In order to fulfill our goal of part-based image generation with only object-level bounding box supervision, we need a model that can disentangle background, object shape, pose, texture, and decompose an object into parts. This is because the model needs to understand what an object part is, without being confused with *background*, and irrespective of specific *shape* (e.g., understand that a duck's head and sparrow's head are the same part), *pose* (e.g., understand that a left-facing duck's head and right-facing duck's head are the same part), and *texture* (e.g., understand that a green duck's head and brown duck's head are the same part). And this can be extremely challenging to achieve without any part or mask annotations.

To this end, we propose PartGAN, a hierarchical generative model that learns part decomposition for image generation and segmentation. To disentangle background, object shape, and texture, we build upon prior work [25, 40] and use information theory [7] to maximize

[1] We focus on single category, object-centric datasets that are often cropped using bounding box annotations, following prior related work [21, 25, 28, 40, 53].

the mutual information between the generated images and their latent codes. To decompose objects into parts, we design a novel part generator, which via several complementary losses learns to predict spatially disjoint part masks without supervision. In order to help discover consistent object parts, we force the generator to reconstruct the object with spatially-pooled features within each generated part mask. In this way, the learned parts converge to regions that are relatively homogeneous (intuitively, in deep feature space, a region covering a single object part will be more homogeneous than that covering multiple or non-object parts) so that the pooled feature can accurately reconstruct the corresponding part.

To summarize, our main contribution is a novel generative model that can discover and generate different parts of objects. Unlike prior work [28], PartGAN represents parts with masks rather than Gaussian distributions, and can disentangle foreground from background, leading to more consistent and precise part decomposition.

2 Related Work

Image generation and disentanglement. Unconditional image generation [2, 14, 35, 40] takes as input random noise vectors, whereas conditional image generation takes input priors such as class labels [4, 30, 32], text [36, 46, 50], or semantic maps or images [17, 20, 25, 28, 33, 43, 55, 56]. Some conditional work disentangle factors of variation (e.g., shape vs appearance) for controllable image generation [12, 22, 25, 37, 40, 44]. However, [12] requires pairs of images depicting the same object appearance; [22] only focuses on human data; [37] usually works on data with less shape/pose variation such as faces, and generalizes poorly to objects with large shape/pose changes such as birds, as indicated in [25]; [44] is a self-supervised approach that disentangles object shape and color, but it requires video data. Among these FineGAN [40] and MixNMatch [25] are most related to our work. They can disentangle four factors (background, object shape, pose, texture). However, FineGAN is an unconditional model, thus cannot synthesize an image based on real reference images. Although MixNMatch is a conditional model, like all other mentioned work, it cannot further decompose an object at the *part* level. Supervised part-based generative models do exist [3, 38, 56], but require expensive (and often difficult to define) part-level annotations. There are also VAE models considering the compositional nature of images [5, 10, 15]. However, they can only decompose and generate images in toy data. ([15] does have results on ImageNet, but it only groups regions by color.) In contrast, our approach is applicable to real-world images and can decompose foreground objects into parts without any part or mask annotations in both unconditional and conditional settings.

Part-based learning. Describing an object as a composition of its different parts is a well-studied problem. Most methods propose discriminative models [13, 29, 31, 49], sometimes without any supervision [1, 8, 18, 39, 41, 47]. In contrast, we propose an part decomposition model that is *generative*, which also allows it to generate and modify specific object parts. A related generative model [28] can achieve part-level control for image generation. However, it uses Gaussian distributions to represent the spatial extent of the discovered parts and cannot disentangle background from foreground. We instead learn detailed pixel-level part masks and disentangle background from foreground, which lead to more accurate shape preservation for part-based texture transfer for image generation.

3 Approach

Let $\mathcal{I} = \{x_1, \ldots, x_N\}$ be an image collection of a single object category (e.g., birds). Our goal is to learn a generative model, which can automatically represent objects as an assembly

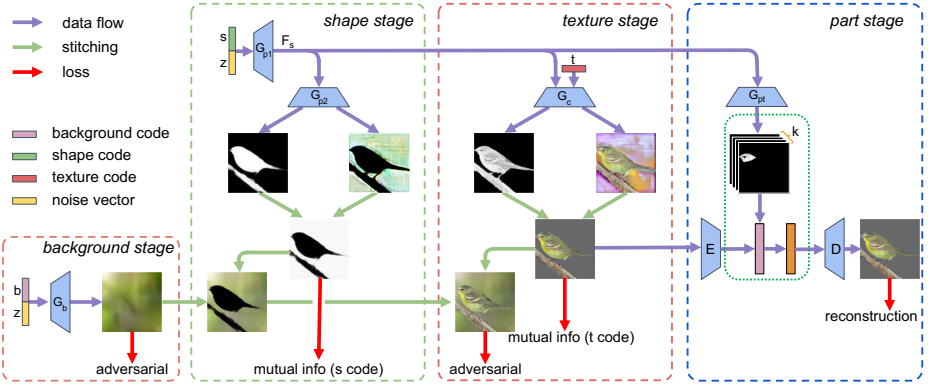

Figure 2: **PartGAN architecture.** The first three stages aim for object-level (background, shape/pose and texture) generation and disentanglement. The part stage further decomposes the object into parts. No part or mask annotations are used during training.

of parts with only object-level bounding box supervision, and without any object mask or part supervision. To this end, we propose PartGAN, which can (1) achieve object-level disentanglement (i.e., background, object shape, texture, and pose), (2) further decompose objects into parts by learning a part-based mask representation, and (3) control the texture at the part-level for both unconditional and conditional image generation.

Fig. 2 shows our overall architecture. PartGAN takes in four randomly sampled codes from prior distributions to hierarchically generate images in four stages (background, shape, texture, and part). Specifically, it takes in a noise vector $z \sim \mathcal{N}(0,1)$, a categorical background code $b \sim \text{Cat}(K = N_b, p = 1/N_b)$, a shape code $s \sim \text{Cat}(K = N_s, p = 1/N_s)$, and a texture code $t \sim \text{Cat}(K = N_t, p = 1/N_t)$. N_b, N_s, N_t are the number of latent background, shape, and texture categories and are set as hyperparameters. The categorical distribution is a natural choice as our prior, as the corresponding factors are mostly discrete; e.g, background: water vs. trees; shape: duck vs. seagull shape; texture: duck color palette vs. seagull color palette. We perform stage-wise image generation, similar to [25, 40, 48], as each stage can be designed to uniquely control one factor of variation.

Background stage. PartGAN generates the background image conditioned on latent background code b and random noise z; see Fig. 2, *background stage*. b learns to control the background type (category) while z adds small variations to the background.

Shape stage. A one-hot shape code s and random noise z together generate a shape mask and shape foreground image which capture the object's shape and pose information. s learns to control the shape type (category) while z learns to control pose. The masked shape foreground image is stitched to the background image from the background stage to form the shape image. The computed shape feature F_s from generator G_{p1} is used in the ensuing stages; see Fig. 2, *shape stage*.

Texture stage. A one-hot texture code t is combined with the shape feature F_s to generate a texture mask and texture foreground image. t learns to control the texture type (category). The texture mask is used to stitch the texture foreground image onto the shape image from the shape stage to form the final generated image; see Fig. 2, *texture stage*.

Part stage. In this stage, PartGAN takes in the shape feature F_s and the masked texture foreground image, and learns to discover different parts. For each part, it generates a part

mask so that it can control the texture at the part-level during image generation.

Next, in Sec. 3.1, we explain how PartGAN achieves object-level disentanglement in the background, shape, and texture stages. Then, in Sec. 3.2, we describe how it can further decompose an object into parts.

3.1 Object-level disentanglement

To disentangle background, we apply adversarial training [14] to learn the distribution of background patches. For this stage, and this stage only, we assume access to object bounding boxes, so that we can sample patches outside the boxes to model the background distribution.

Next, to achieve disentanglement of shape and texture without supervision, we use information theory [7, 40]. Specifically, in the shape and texture stages, we maximize the mutual information between the shape code s and masked shape foreground image, and texture code t and the masked texture foreground image, respectively, as demonstrated in Fig. 2. The masked image in each stage is obtained by performing an element-wise multiplication between the generated mask and generated foreground image. Following FineGAN [40], we impose constraints on the relationships between the latent codes to induce the desired disentanglement: we (1) constrain the sampled texture codes into disjoint groups so that each group shares the same shape code, and (2) enforce the sampled background and texture codes to be the same. These two constraints model two aspects of real-world data: (1) some object instances share a common shape even if they have different textures (e.g., different seagull species with different texture details share the same seagull shape); (2) a specific object type often appears with correlated background (e.g., ducks typically have water as background). And finally, by setting the number of latent shape categories N_s to be much smaller than all possible pose variations of shape in the data, the continuous code z can be made to control the object's pose; i.e., so that each particular shape instance (e.g., a left-facing duck) can be described as a combination of general shape plus pose (e.g., duck shape code s plus left-facing pose z). The losses used to train the background, shape, and texture stages are identical to FineGAN [40]. Due to limited space, we provide more details in the supp.

3.2 Part-level decomposition

We design a new part stage to discover and control different object parts. In this stage, the part generator G_{pt} takes in the shape feature F_s from the shape stage and outputs k part masks M^i_{pt}, where $i = 1, 2, ..., k$. Each part mask has pixel values in $[0, 1]$. We use the shape feature F_s from the shape stage because it already contains spatially rich part information due to texture stage training. Specifically, since it is used to generate different texture details conditioned on different texture codes, F_s must understand different object parts in order to accurately generate part-level texture details. Nonetheless, despite F_s having part-level information, the part details are entangled together as one feature, which means that we cannot control the texture of each part separately. In order to achieve *part decomposition without supervision*, we apply the following constraints.

Merge constraint. Recall that our model generates an object-level texture mask in the texture stage to stitch the generated texture foreground image onto the shape image from the shape stage. Thus, to make sure the generated parts together focus on the entire object, we enforce the sum of part masks to be equal to the object-level texture mask M_t:

$$\mathcal{L}_{merge} = |\sum_{i=1}^{k} M^i_{pt} - M_t|. \tag{1}$$

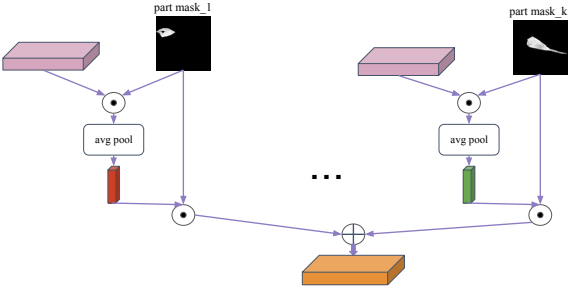

Figure 3: **Details of the part stage.** We first use our model's predicted part masks to pool the foreground texture feature (pink block). The pooled features (red and green vertical bars) are then used with the part masks to reconstruct the full foreground texture feature (orange block). This operation forces the model to learn more meaningful parts that are conditioned on the object, which are usually more homogeneous compared to arbitrary spatial partitions.

Concentration constraint. We desire each part to be concentrated in a spatially local region. To achieve this, we leverage the concentration loss [53], where we consider each part mask as the density of a bivariate (x,y) distribution in image coordinate space:

$$\mathcal{L}_{concentration} = \frac{1}{k}\sum_{i=1}^{k} 2\pi e(\sigma_{x_i}^2 + \sigma_{y_i}^2) \tag{2}$$

where $\sigma_{x_i}^2$ and $\sigma_{y_i}^2$ are variances along the two axes for part mask M_{pt}^i. Importantly, we use these statistics only for computing this loss while our part representations are still pixel-level masks. In contrast, [28] uses them (plus the mean) as the final representation of each part.

Partition constraint. Ideally, each part mask should depict a unique spatially disjoint region. To this end, we penalize any spatial overlap between any two part masks:

$$\mathcal{L}_{partition} = \frac{1}{C(k,2)}\sum_{i,j} M_{pt}^i \cdot M_{pt}^j, \text{ where } i \neq j. \tag{3}$$

Here $C(k,2)$ is the binomial coefficient (i.e., k choose 2) and $\cdot$ denotes the dot product.

Balancing constraint. We prefer each part to have reasonable size; i.e., so that no one part dominates the entire object or is represented only by a few pixels. Thus, we constrain the ratio of each part mask over the texture mask to be equal to $1/k$ within threshold t:

$$\mathcal{L}_{ratio} = \frac{1}{k}\sum_{i=1}^{k} \max(|r_i - \frac{1}{k}| - t, 0) \tag{4}$$

where r_i is ratio of total mass (sum over pixels) of i'th part mask M_{pt}^i to texture mask M_t.

Reconstruction constraint. Although the above constraints restrict each part mask to be non-overlapping, locally concentrated, and balanced in size, they do not enforce the parts to be meaningful (i.e., conditioned on the object). For example, the parts could be chosen to divide an image purely based on spatial location (e.g., horizontally divide the image into top, middle, and bottom regions for $k = 3$, as shown in the supp ablation studies) while ignoring the object. In order to push the part generator G_{pt} to learn consistent and meaningful part masks, we apply a novel reconstruction constraint on the generated part masks.

As shown in the part stage of Fig. 2 and Fig. 3, we first process the masked texture foreground image from the texture stage through an encoder E to get its feature representation (pink block). We next mask it with each generated part mask (via elementwise multiplication) and perform average pooling to get the feature representation vector for each part.

Figure 4: **Part segmentation on real images.** PartGAN can discover consistent parts, which are sometimes semantically meaningful, on diverse datasets without part supervision. Here the number of parts are $k = 5, 4, 10, 10, 4$, respectively.

We then paste the part feature back to all spatial locations, weighted by the part mask, and sum the result across all parts to get the reconstructed full feature representation (orange block). A decoder D takes the reconstructed feature representation along with channel-wise concatenated part masks, and tries to reconstruct the masked texture foreground image. Here we apply the L1 loss for the reconstruction task. Since the pooling operation destroys spatial information, it is easier for the model to learn meaningful parts, which are usually more homogeneous than arbitrary spatial partitions, for the reconstruction task.

3.3 Part based image generation

For unconditional generation, we first sample a background code b, shape code s, and a noise vector z to generate the background and shape image. In the texture stage, the shape feature F_s and a texture code t are used to fill in the foreground texture details to generate the complete image I. F_s is also given to the part stage for part decomposition and part mask generation. To change the texture of a specific part, say the i'th part, a different texture code can be input to the texture stage (with the same F_s as before) to generate a new foreground texture. Then the i'th part mask can be used to stitch the new texture foreground onto the existing image I to only change the texture of the i'th part.

For conditional generation, we train a set of encoders that take in a real image and predict the corresponding pose z, background b, shape s, and texture t codes. To make the encoders learn to extract those disentangled factors, without supervision, we use the technique proposed in MixNMatch [25]. Briefly, we train the encoders and generator using BiGAN [9] for paired image-code distribution matching, so that the encoded codes match the same prior distributions as the sampled latent codes in the unconditional setting, while simultaneously making the generated image look real. The extracted codes can then be fed into our PartGAN generator, and adjusted in the same way as in the unconditional setting, for changing the texture of specific parts during image generation.

4 Experiments

In this section, we qualitatively evaluate part segmentation and part-level texture control on image generation, and quantitatively evaluate the consistency of our learned parts. We do not quantitatively evaluate image generation quality directly using metrics like FID [16], since

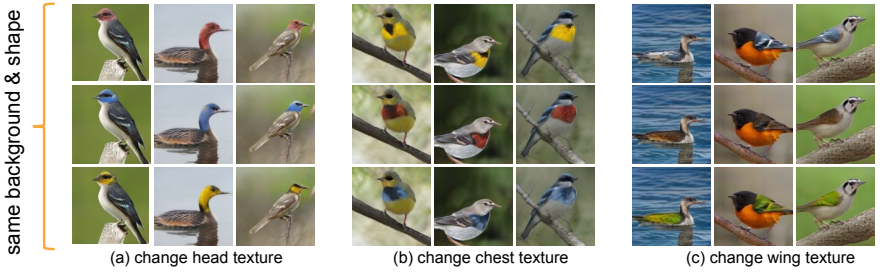

Figure 5: **Unconditional generation.** The images in each column share the same background, shape, and texture codes except for one part. We randomly sample different texture codes for the remaining part to change the texture of the head (a), chest (b), and wing (c).

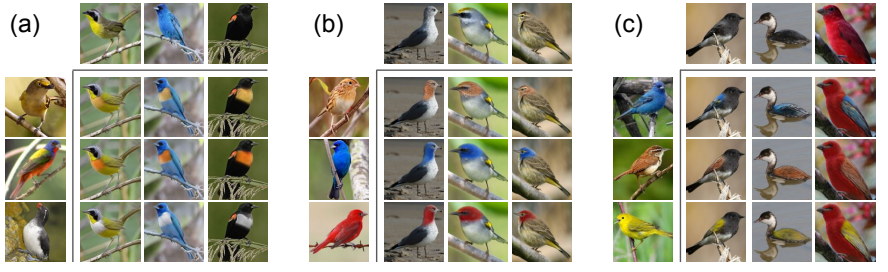

Figure 6: **Conditional generation.** For each sub-figure, the three real images on the left provide texture information for one part (chest (a), head (b), wing (c)), while the top three real images provide background, shape, and texture information for the rest of the parts.

the generated images with appearance changes made to one or more parts would not follow the real distribution. *Implementation details can be found in the supp.*

Datasets. (1) **CUB** [42]: 11,788 bird images from 200 classes. (2) **Stanford Cars** [24]: 8,144 car images from 196 classes. (3) **Deepfashion**: a clothes dataset [27], we follow [28] and select images of full bodies (all keypoints visible, measured by [54]) for quantitative comparison. (4) **BBC Pose** [6] contains videos of sign-language signers, we use the same train-test split as [28]. (5) **Human3.6M** [19] contains videos of human activities. We remove the background following [28, 53]. (6) **CelebA** [26] contains celebrity faces. For evaluation, we use the MAFL subset. (7) **Cat head** [52] has 9,997 images of cat heads.

4.1 Qualitative Results

Part segmentation. Fig. 4 shows PartGAN's discovered parts in the conditional image generation setting. PartGAN generates consistent part masks across different instances of an object category for a variety of datasets. Although our model does not use any part supervision, it can also sometimes discover semantic parts like the head, wing, and belly for birds, and lips, eyes, and forehead for human faces. We believe this is due to such semantic parts being spatially local and mostly consistent in appearance across instances, which are properties that our part constraints try to capture.

Unconditional part generation. Fig. 5 shows unconditional generation results from sampled latent codes. The images in each column share the same background, shape, pose codes, and same texture code for all but one part. We change the texture code for the (a) head, (b) chest, and (c) wing. The results show that our learned masks are meaningful

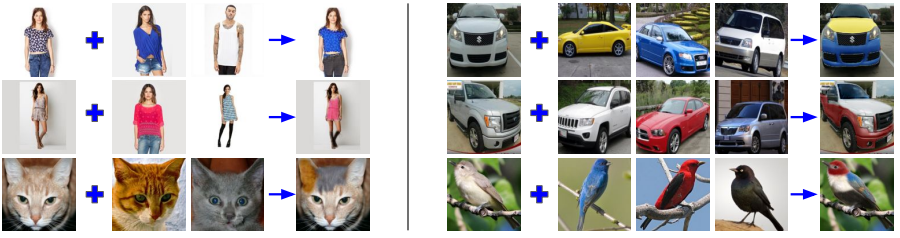

Figure 7: **Multi-part texture transfer.** Multiple parts texture can be transferred; e.g., (top right) car hood and body are made yellow and blue, and windows are darkened.

and consistent across instances, and that PartGAN can successfully change the texture of a specific region. *Note that our model does not learn with any part supervision, and we only provide semantic names for these parts to ease our explanation.*

Conditional part generation. We show part-level texture transfer given real reference images in Fig. 6. In each sub-figure, the real reference images on the top provide background, shape, and texture (for all but one part) information, while the reference images on the left provide the texture information of the part to be transferred: (a) chest, (b) head, and (c) wing. In Fig. 7, we show the results of transferring the texture of multiple parts. In the first row, we successfully transfer the texture of the shirt (2nd image) and pants (3rd image) to the first image. In the third row, we transfer the texture of the cat's ear and forehead. For the cars, we transfer the texture of the hood, body, and windows. These results demonstrate PartGAN's accurate part decomposition and texture transfer for image generation. In the supp, we also show an application for colorizing sketch images with different colors for different parts.

4.2 Quantitative Results

Landmark prediction. Following prior work [18, 28], we predict keypoints from the discovered parts as a proxy to evaluate their consistency across instances of a class. For each dataset, we train a convolutional landmark predictor (imp. details in supp) which takes the learned part masks as input and predicts 2D landmark points. We compare with baseline generative models [21, 28, 53] that can perform landmark discovery. Unlike our approach, which predicts pixel-level segmentation part masks, these methods predict hard/soft keypoints. Hence, they instead train a regressor to regress from their predicted 2D keypoints to ground-truth landmarks, as they cannot take advantage of more detailed part segmentation masks. Moreover, according to [21], these methods sometimes rely on ground truth keypoints to first select a subset of the best candidate parts before learning a regressor, likely due to their lack of ability to explicitly differentiate foreground from background.

Table 1 shows landmark prediction error results. For all datasets, we follow same train/test splits as [28]. We obtain state-of-the-art results on all but one dataset, which shows that PartGAN discovers more consistent parts than the baselines. This can be attributed to both PartGAN's prediction of detailed pixel-level part masks and its ability to disentangle foreground from background. The closest baseline to ours is [28], which also learns a generative model to perform part decomposition without part supervision. However, it only models rough part shape using a Gaussian distribution and cannot disentangle foreground from background. Hence, we have better landmark prediction, except for CUB. The worse result on CUB is in part due to PartGAN sometimes predicting the bird's tail as part of the branch that it sits on (e.g., when their colors are similar), which can confuse the tail landmark predictor.

	CAT	MAFL	CUB	BBC	Human3.6
Zhang [53]	15.35	3.46	5.36	-	4.14
Jakab [21]	-	3.19	-	68.4%	-
Lorenz [28]	9.88	3.24	**3.91**	74.5%	2.79
Ours	**9.34**	**3.08**	5.05	**77.0%**	**2.76**

Table 1: **Landmark prediction error results.** The error is in % of inter-ocular distance for Cat Head, MAFL and in % of the image edge length for CUB, Human3.6. Percentage of correct landmarks within 6 pixels reported for BBC pose.

	α=2.5%	α =5%	α =7.5%	α =10%
Lorenz [28]	85.6%	94.2%	96.5%	97.4%
Esser [11]	95.2%	98.4%	98.9%	99.1%
Ours	**96.6%**	**99.1%**	**99.6%**	**99.7%**

Table 2: **Shape consisency results.** We transfer texture from image B to image A to generate image C, and measure shape consistency between A and C. PCK measures the % of keypoints within α% (pixel distance / image diagonal) between images A and C.

Shape consistency. We next evaluate Percentage of Correct Keypoints (PCK) on DeepFashion dataset [27] following the same train/test split as [28]. We measure the consistency in shapes between the generated and real source image (from which shape is borrowed) during conditional image generation. As both our method and [28] try to reconstruct the full object shape from the predicted parts, their shape consistency would be reflective of the quality of the discovered parts. Specifically, for each image A, we assign its texture according to a random image B, to generate image C. We use [54] to predict keypoints of images A and C, and calculate PCK in image C according to image A.

Table 2 shows that we significantly outperform [28]. Since PartGAN generates pixel-level masks to represent a part, whereas [28] uses a Gaussian representation, PartGAN better preserves shape details of source image A when generating image C. We also compare to VU-Net [11], which relies on a supervised human pose detector [54] to transfer the texture. Still, PartGAN performs better, which again demonstrates that a mask representation can better preserve spatial details than keypoints.

Ablation studies. We conduct detailed ablation studies, where we show necessities of all losses. Please refer supp for details.

Comparison to [18] and failure cases. In the supp, we provide more analyses including a comparison to a discriminative part segmentation model [18] and discussing failure cases.

5 Conclusion and Limitations

We proposed PartGAN, a novel generative model that can decompose objects at the part-level without any mask or part annotations. On diverse datasets, we showed that it consistently discovers meaningful parts and can transfer part-level texture for image generation. One limitation is that the number of parts has to be set manually and when the number of parts is set too large, a meaningful part can get broken down into several not so relevant sub-parts (e.g., we obtain three different parts for the human forehead in Figure 4). How to automatically set the number of parts without part supervision is an interesting and important topic for future work. Nonetheless, we believe this work makes an important contribution in weakly-supervised part decomposition for generative image modeling.

Acknowledgments. This work was supported in part by a Sony Focused Research Award, NSF CAREER IIS-1751206, and IIS-1812850.

References

[1] Vittorio Ferrari Abel Gonzalez-Garcia, Davide Modolo. Do semantic parts emerge in convolutional neural networks? In *IJCV*, 2018.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[3] Guha Balakrishnan, Amy Zhao, Adrian Dalca, Fredo Durand, and John Guttag. Synthesizing images of humans in unseen poses. In *CVPR*, 2018.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019.

[5] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. In *arXiv*, 2019.

[6] James Charles, Tomas Pfister, Derek Magee, David Hogg, and Andrew Zisserman. Domain adaptation for upper body pose tracking in signed tv broadcasts. In *BMVC*, 2013.

[7] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NeurIPS*, 2016.

[8] Edo Collins, Radhakrishna Achanta, and Sabine Süsstrunk. Deep feature factorization for concept discovery. In *ECCV*, 2018.

[9] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *ICLR*, 2017.

[10] Martin Engelcke, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. In *ICLR*, 2020.

[11] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. In *CVPR*, 2018.

[12] Patrick Esser, Johannes Haux, and Björn Ommer. Unsupervised robust disentangling of latent characteristics for image synthesis. In *ICCV*, 2019.

[13] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. In *TPAMI*, 2010.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[15] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Chris Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *ICML*, 2019.

[16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Gunter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

[17] Xun Huang, Ming-Yu Liu, Serge J. Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018.

[18] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. Scops: Self-supervised co-part segmentation. In *CVPR*, 2019.

[19] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. In *TPAMI*, 2014.

[20] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial nets. *CVPR*, 2017.

[21] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *NeurIPS*, 2018.

[22] Donggyu Joo, Doyeon Kim, and Junmo Kim. Generating a fusion image: One's identity and another's shape. In *CVPR*, 2018.

[23] Hadi Kazemi, Seyed Mehdi Iranmanesh, and Nasser M. Nasrabadi. Style and content disentanglement in generative adversarial networks. In *WACV*, 2018.

[24] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, 2013.

[25] Yuheng Li, Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. Mixnmatch: Multifactor disentanglement and encoding for conditional image generation. In *CVPR*, 2020.

[26] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[27] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, 2016.

[28] Dominik Lorenz, Leonard Bereska, Timo Milbich, and Björn Ommer. Unsupervised part-based disentangling of object shape and appearance. In *CVPR*, 2019.

[29] Behrooz Mahasseni Michael Lam and Sinisa Todorovic. Fine-grained recognition as hsnet search for informative image parts. In *CVPR*, 2017.

[30] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. In *ICLR*, 2018.

[31] David Novotny, Diane Larlus, and Andrea Vedaldi. Anchornet: A weakly supervised network to learn geometry-sensitive features for semantic matching. In *CVPR*, 2017.

[32] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, 2017.

[33] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019.

[34] Yuxin Peng, Xiangteng He, and Junjie Zhao. Object-part attention model for fine-grained image classification. *TIP*, 2017.

[35] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *ICLR*, 2016.

[36] Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, 2016.

[37] Zhixin Shu, Mihir Sahasrabudhe, Riza Alp Güler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *ECCV*, 2018.

[38] Aliaksandr Siarohin, Enver Sangineto, Stephane Lathuiliere, and Nicu Sebe. Deformable gans for pose-based human image generation. In *CVPR*, 2018.

[39] Aliaksandr Siarohin, Subhankar Roy, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Motion-supervised co-part segmentation. In *arXiv preprint arXiv:2004.03234*, 2020.

[40] Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. FineGAN: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In *CVPR*, 2019.

[41] Saurabh Singh, Abhinav Gupta, and Alexei A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012.

[42] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, 2011.

[43] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018.

[44] Fanyi Xiao, Haotian Liu, and Yong Jae Lee. Identity from here, pose from there: Self-supervised disentanglement and generation of objects using unlabeled videos. In *ICCV*, 2019.

[45] Xianglei Xing, Ruiqi Gao, Tian Han, Song-Chun Zhu, and Ying Nian Wu. Deformable generator network: Unsupervised disentanglement of appearance and geometry. In *CVPR*, 2018.

[46] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, 2018.

[47] Zhenjia Xu, Zhijian Liu, Chen Sun, Kevin Murphy, William T. Freeman, Joshua B. Tenenbaum, and Jiajun Wu. Unsupervised discovery of parts, structure, and dynamics. In *ICLR*, 2019.

[48] Jianwei Yang, Anitha Kannan, Dhruv Batra, and Devi Parikh. Lr-gan: Layered recursive generative adversarial networks for image generation. *ICLR*, 2017.

[49] Wei Yang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *CVPR*, 2016.

[50] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *arXiv: 1710.10916*, 2017.

[51] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. In *ECCV*, 2014.

[52] Weiwei Zhang, Jian Sun, and Xiaoou Tang. Cat head detection - how to effectively exploit shape and texture features. In *ECCV*, 2008.

[53] Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In *CVPR*, 2018.

[54] Cao Zhe, Simon Tomas, Wei Shih-En, and Sheikh Yaser. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[55] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *ICCV*, 2017.

[56] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *CVPR*, 2020.