Linear Semantics in Generative Adversarial Networks

Jianjin Xu *
Columbia University
jx2386@columbia.edu

Changxi Zheng
Columbia University
cxz@cs.columbia.edu

Abstract

Generative Adversarial Networks (GANs) are able to generate high-quality images, but it remains difficult to explicitly specify the semantics of synthesized images. In this work, we aim to better understand the semantic representation of GANs, and thereby enable semantic control in GAN's generation process. Interestingly, we find that a well-trained GAN encodes image semantics in its internal feature maps in a surprisingly simple way: a linear transformation of feature maps suffices to extract the generated image semantics. To verify this simplicity, we conduct extensive experiments on various GANs and datasets; and thanks to this simplicity, we are able to learn a semantic segmentation model for a trained GAN from a small number (e.g., 8) of labeled images. Last but not least, leveraging our finding, we propose two few-shot image editing approaches, namely Semantic-Conditional Sampling and Semantic Image Editing. Given a trained GAN and as few as eight semantic annotations, the user is able to generate diverse images subject to a user-provided semantic layout, and control the synthesized image semantics. We have made the code publicly available[1].

1. Introduction

Recent years have witnessed the striking success of Generative Adversarial Networks (GANs) [15] in various image synthesis tasks: to generate human faces, animals, cars, and interior scenes [28, 12, 17, 21]. Apart from improving the generated image quality, recent research has been directed toward the *control* of GAN's image generation process—for example, to enforce the generated images having user specified attributes, colors, and layouts.

Toward this goal, a fundamental question remains unanswered: how does a well-trained GAN encodes image semantics—such as the layout of such semantic classes as hair, nose, and hats—in its image generation process? Motivated by this question, we aim to extract image semantics from a GAN's internal data, namely its feature maps. If we can well extract image semantics and understand the extraction process, we can develop insight on how the image semantics are encoded.

Our finding is surprisingly simple: a linear transformation on the GAN's internal feature maps suffices to extract the generated image semantics. In stark contrast to GAN's highly nonlinear image generation process, this simple linear transformation is easy to understand and has a clear geometric interpretation (see Sec. 3.1).

We refer to this linear transformation process as linear semantic extraction (LSE). To verify its performance, we conduct extensive experiments on various GANs and datasets, including PGGAN [20], StyleGAN [21] and StyleGAN2 [22] trained on FFHQ [21], CelebAHQ [25], and LSUN [37]'s bedroom and church dataset. We also compare the performance of LSE with other semantic extraction approaches which use learned nonlinear transformations. It turns out that LSE is highly comparable to those more complex, nonlinear models, suggesting that image semantics are indeed represented in a linear fashion in GANs.

Related to our study of the linear encoding of image semantics in GANs is the work of GAN Dissection [5]. It identifies feature maps that have causal manipulation ability for image semantics. Yet, most feature maps in that approach come from middle-level layers in the GAN, often having much lower resolution than the output image. Instead, we examine the GAN's internal feature maps collectively. We upsample all feature maps to the resolution of final output image and stack them into a tensor. This approach allows us to study per-pixel feature vectors, that is, feature values corresponding to a particular pixel across all internal layers, and we are able to classify every output pixel into a specific semantic class.

The linear transformation in our proposed LSE is learned under supervision. Its training requires image semantic annotations, which are automatically generated using a pretrained segmentation model (such as UNet [29]). Interestingly, thanks to the linearity of LSE, even a small number of annotations suffice to train LSE well. For example, the LSE trained with 16 annotated images on StyleGAN2 (which itself is trained on FFHQ dataset) achieves 88.1% performance

*Jianjin Xu is currently a research assistant at Panzhihua University.
[1]https://github.com/AtlantixJJ/LinearGAN

relative to a fully trained LSE model. Not only does this result further support our finding about the linear representation of semantics in GANs, it also inspires new approaches for controlling image generation through few-shot training.

In particular, we explore two controlled image generation tasks: (1) *Semantic Image Editing* (SIE) and (2) *Semantic-Conditional Sampling* (SCS). The former aims to update images based on the user's edit on the semantics of a GAN's output (e.g., generate images in which the hair region is reshaped); the latter is meant to generate images subject to a user specification of desired semantic layout (e.g., produce images of an interior room where the furnitures are laid out according to user specification). We demonstrate few-shot SIE and SCS models both trained with small number of annotated images.

Behind both SCS and SIE is the core idea of matching the generated image semantics with a target semantic specification. This is done by formulating an optimization problem, one that finds a proper latent vector for the GAN's image generation while respecting the user specification. We also consider baselines of both tasks, which are implemented by using carefully trained, off-the-shelf semantic segmentation models rather than our few-shot LSE. In comparison to the baselines, our approach with 8-shot LSE is able to generate comparable (and sometimes even better) image quality.

In summary, our technical contributions are twofold: **(i)** Through extensive experiments, we show that GANs represent the image's pixel-level semantics in a linear fashion. **(ii)** We propose an LSE with few-shot learning, which further enables two image synthesis applications with semantic control, namely SCS and SIE under few-shot settings.

2. Related work

Generative Adversarial Networks. GANs [15] have achieved tremendous success in image generation tasks, such as synthesizing photo-realistic facial images [20, 21, 22], cityscapes [35, 27] and ImageNet images [38, 6]. Among various GAN models, Progressively Grown GAN (PGGAN) [20], StyleGAN [21] and its improvement StyleGAN2 [22] are three of the most widely used GAN structures. PGGAN shares a similar architecture as the Deep Convolution GAN (DCGAN) [28], trained progressively. StyleGAN adopts the adaptive instance normalization [16] from neural stylization literatures to improve generation quality. Further improving on StyleGAN, StyleGAN2 is by far the state-of-the-art GAN model on various datasets. We therefore conduct experiments on the three types of GANs.

Interpreting GANs. Our study of image semantics in GAN models is related to the works toward interpreting and dissecting GANs. Along this research direction, existing methods can be grouped into two categories. First are those aiming to interpret a GAN's latent space. Prior works [30, 36] find that there exist linear boundaries in latent space that separate positive and negative attributes of image samples. Others works [31, 19, 34] propose to find linear trajectories of attributes in the latent space in an unsupervised way. Second, interpreting the feature maps of GANs. GAN Dissection [5] identifies convolution units that have causality with semantics in the generated images. Collins et al. [11] find that the clusters of semantics can be found by k-means and matrix factorization in GAN's feature maps. Our differences are two-fold. First, we study the high-resolution semantic masks extracted from the generator, which is rarely touched in existing works. Second, the SIE and SCS applications derived from our discoveries are novel in terms of their few-shot settings.

Controlling GANs. Methods to enable GAN's controllability can be divided into two streams. First, training new GANs with architectures specifically designed to enable controllability. Conditional GAN (cGAN) and its variant [26, 13, 9] are proposed to enable GAN's controllability for category. StackGAN [40] extends cGAN by using the embedding of natural language to control the synthesis. The image-to-image translation networks can map semantic masks to images [45, 18, 27, 46]. They allow explicit control of semantic structures but need expensive labeled data for training. Second, interpreting or devising auxilary architectures to exploit the controllability of pretrained GAN models. The controllability for *global attributes* is studied by many interpretation-based editing methods [30, 36, 31, 19, 34]. Besides interpretation, other methods propose auxilary networks for the controllability for attributes [3] or 3D characteristics [14, 33, 41]. The controllability for *local image editing* also receives much research attention. The latent code optimization methods [44, 7] can make the image resemble the color strokes drawn by users, but the precision of editing is limited. The feature map substitution methods [1, 32, 10] can edit a localized region of an image precisely, but the editing operation requires users to find a source image for reference. GAN Dissection [5] succeed in editing the semantics of images, but its resolution and diversity are limited. Bau et al. [4] rewrite the weight of a generator to change its generation pattern.

The semantic controllability studied in our work differs from previous works in two aspects. First, previous SCS models in the context of image-to-image translation require extensively labeled images, whereas our approach requires only a few annotations in its training. Second, previous SIE models (such as [5]) are mainly concerned with the control of semantic morphology, not the user's ability to fully specify semantic regions. As a result, our approach requires no reference image, and thereby eases the user editing process.

3. GAN's Linear Embedding of Semantics

We aim to decode a GAN's internal representation of image semantics in its image synthesis process. Our find-

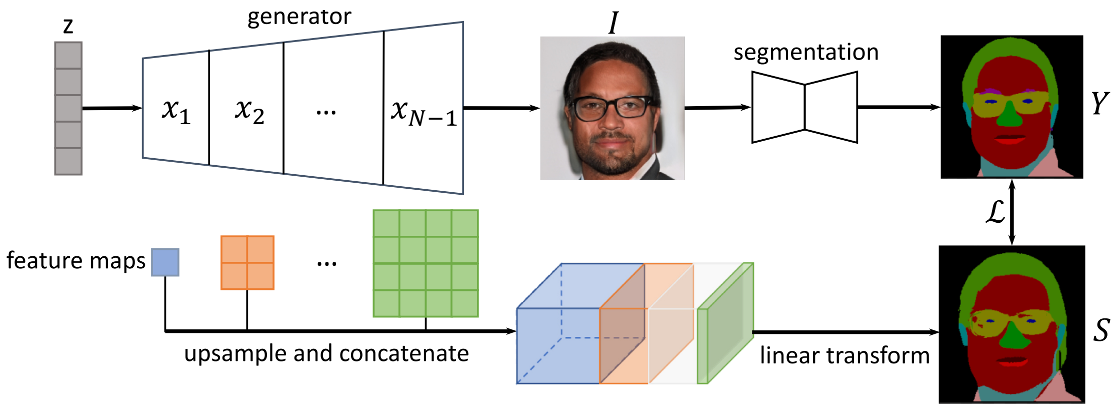


Figure 1. When synthesizing an image I from a latent vector z, the generator builds a series of internal feature maps. Provided a well-trained GAN model, we decode its feature maps $\{\mathbf{x}_i\}_{i=1}^{N-1}$ to extract the output image's semantic segmentation $\mathbf{S}$. This is done by learning a simple linear transformation applied on the feature maps. Learning the linear transformation is supervised by a pretrained segmentation model.

ing is surprisingly simple: a *linear transformation* on the GAN's feature maps suffices to reveal its synthesized image semantics. In this section, we first construct such a linear transformation (Sec. 3.1), and then conduct experiments (Sec. 3.2) to examine its competence for revealing image semantics (Sec. 3.3).

3.1. Linear Transformation on Feature Maps

A well-trained GAN model maps a randomly chosen latent vector to a realistic image. Structurally, a GAN model concatenates a series of network layers. Provided a latent vector, each layer i outputs a feature map $\mathbf{x}_i$, which is in turn fed into the next layer. We denote the width, height, and depth of $\mathbf{x}_i$ using w_i, h_i and c_i, respectively (i.e., $\mathbf{x}_i \in \mathbb{R}^{c_i \times w_i \times h_i}$).

It is unsurprising at all that one can deduce from the feature maps the generated image semantics. After all, feature maps represent the GAN's internal data flow that results in the final image. As images can be semantically segmented using pretrained networks, the feature map can also be segmented with appropriate networks. More interesting is the question of how *easily* we can learn from feature maps about the generated image semantics. A straightforward relation between feature maps and image semantics could be easy to understand, and inspire new theories and applications.

Objective. Consider a GAN model consisting of N layers and producing images with m semantic classes (such as hair, face, and cloth). We seek the simplest possible relation between its feature maps and output image semantics—a linear transformation matrix $\mathbf{T}_i$ applied to each feature map $\mathbf{x}_i$ to predict a semantic map of the layer i. By accumulating all the maps, we wish to predict a semantic segmentation $\mathbf{S}$ of the GAN's output image (see Fig. 1). Formally, $\mathbf{S}$ is just a linear transformation of all feature maps, defined as

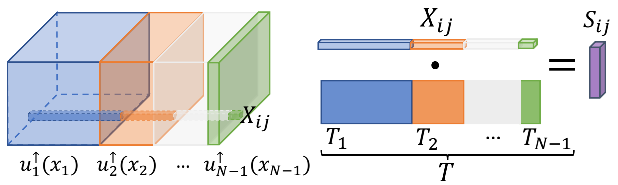
$$\mathbf{S} = \sum_{i=1}^{N-1} \mathsf{u}_i^{\uparrow}(\mathbf{T}_i \cdot \mathbf{x}_i), \tag{1}$$

where $\mathbf{T}_i \in \mathbb{R}^{m \times c_i}$ converts $\mathbf{x}_i \in \mathbb{R}^{c_i \times w_i \times h_i}$ into a semantic map $\mathbf{T}_i \cdot \mathbf{x}_i \in \mathbb{R}^{m \times w_i \times h_i}$ through a tensor contraction along the depth axis. The result from each layer is then upsampled (denoted by $\mathsf{u}_i^{\uparrow}$) to the output image resolution. The summation extends over all internal layers, excluding the last layer (layer N), which outputs the final image. The result $\mathbf{S} \in \mathbb{R}^{m \times w \times h}$ has the same spatial resolution $w \times h$ as the output image. Each pixel $\mathbf{S}_{ij}$ is a $m \times 1$ vector, indicating the pixel's unnormalized logarithmic probabilities representing each of the m semantic classes. We refer to this method as Linear Semantic Extractor (LSE).

Optimizing $\mathbf{T}_i$. The training process of LSE is supervised by pixel-level annotation of semantics. Yet, it is impractical to manually annotate a large set of images that are automatically generated by a GAN model. Instead, we leverage off-the-shelf pretrained segmentation models for semantic annotation. In practice, we use UNet [29] to segment facial images (into the nose, eye, ear, and other semantic regions), and DeepLabV3 [8] with ResNeSt backbone [39] for bedroom and church images.

Concretely, provided a well-trained GAN model, we randomly sample its latent space to produce a set $\mathbb{S}$ of synthetic images. When synthesizing every image in $\mathbb{S}$, we also record the model's feature maps $\{\mathbf{x}_i\}_{i=1}^{N-1}$. These feature maps are linearly transformed using (1) to predict a semantic mask of the image, which is then compared with the result from the pretrained semantic segmentation network to form the standard cross-entropy loss function:

$$\mathcal{L} = \frac{1}{w \cdot h} \sum_{\substack{1 \le i \le w \\ 1 \le j \le h}} \left[\log \left(\sum_{k=1}^{m} \exp\left(\mathbf{S}_{ij}[k]\right) \right) - \mathbf{S}_{ij}[Y_{ij}] \right],$$



(a) upsampled feature maps $\mathbf{X}$ (b) dot product $\mathbf{S}_{ij} = \mathbf{T} \cdot \mathbf{X}_{ij}$

Figure 2. The visualization of the linear transformation, with upsample placed before convolution.

where Y_{ij} is the semantic class at pixel (i, j) indicated by the supervisor network, and $\mathbf{S}_{ij}[k]$ is the corresponding unnormalized logarithmic probability for the k-th semantic class predicted by the LSE.

Lastly, the linear matrices $\mathbf{T}_i$ are optimized by minimizing the expected loss (estimated by taking the average loss over image batches in $\mathbb{S}$). Details of the training process are provided in Appx. D.

Geometric picture. The linear relation (1) allows us to draw an intuitive geometric picture of how image semantics are encoded in the generator's feature maps.

First, notice that $\mathbf{T}_i$ applied on $\mathbf{x}_i$ can be viewed as a 1×1 convolutional filter with stride 1. The filter operation is commutative with the upsample operation $\mathsf{u}_i^{\uparrow}(\cdot)$ (see Appx. B for a proof of this commutative property). As a result, we can rewrite the semantic prediction $\mathbf{S}$ in (1) as

$$\mathbf{S} = \sum_{i=1}^{N-1} \mathbf{T}_i \cdot \mathsf{u}_i^{\uparrow}(\mathbf{x}_i) = \mathbf{T} \cdot \mathbf{X}, \tag{2}$$

where $\mathbf{T} = \begin{bmatrix} \mathbf{T}_1 & \ldots & \mathbf{T}_{N-1} \end{bmatrix}$ is an $m \times n$ matrix with $n = \sum_{i=1}^{N-1} c_i$ being the total layer depth. $\mathbf{X} \in \mathbb{R}^{n \times w \times h}$ is a tensor concatenating all upsampled $\mathbf{x}_i$ (i.e., $\mathsf{u}_i^{\uparrow}(\mathbf{x}_i)$ with resolution $c_i \times w \times h$) along the depth axis (see Fig. 2a).

Now, consider a pixel (i, j) in the output image. To predict its semantic class, Equation (2) shows that we can take the corresponding $n \times 1$ vector $\mathbf{X}_{ij}$ that stacks the pixel's features resulted from all GAN layers, and dot product it with each row of $\mathbf{T}$ (see Fig. 2b): $\mathbf{S}_{ij} = \mathbf{T} \cdot \mathbf{X}_{ij}$. In other words, each row $\mathbf{T}^{(k)}$ of $\mathbf{T}$ defines a direction representing the semantic class k in the n-dimensional feature space.

If the linear transformation can classify features with high accuracy, then the feature vectors of different semantic classes are linearly separable. Define the set of all vectors that are classified into class k as

$$\mathcal{R}_k = \{\boldsymbol{x} | \mathbf{T}^{(k)}\boldsymbol{x} > \mathbf{T}^{(j)}\boldsymbol{x}, \forall j \neq k\}, \tag{3}$$

where $\mathbf{T}^{(k)}$ is the k-th row of the tensor $\mathbf{T}$. This definition shows that the subspace of each semantic class forms a hyper-cone originating from the origin.

An intuitive geometric picture is as follows. Consider a unit n-sphere at the origin. The intersection of a semantic class i's hyper-cone and the sphere surface encloses a convex area A_i. Then, take the feature values at a pixel and normalize it into a unit vector. If that vector falls into the convex area A_i, then the pixel is classified as the class i. In other words, the surface of n-sphere is divided into k convex areas, each representing a semantic class. From this geometric perspective, we can even infer a pixel's semantic class without training the linear model (1). Rather, we locate a semantic *center* c_i for each convex area A_i on the n-sphere surface. For example, the semantic centers can be estimated by a clustering algorithm (such as k-means clustering). A pixel is classified as class i if its feature vector is closest to c_i (among all semantic centers) on the n-sphere. In Sec. 3.3, we show that the class centers can segment images reasonably well, supporting our hyper-cone interpretation.

Nonlinear semantic extraction. If LSE can extract the semantics of generated images, a further question is to what extent the semantics can be better extracted by nonlinear models. The answer to this question provides further support on whether or not feature maps in GANs indeed encode image semantics linearly. If they do, then nonlinear models would perform *no* significantly better than our linear model.

We propose two nonlinear extraction models for this study. The first Nonlinear Semantic Extractor (which we referred to as NSE-1) transforms each layer's feature maps through three convolutional layers interleaved with ReLU activations. Each transformed feature map is upsampled using the same interpolation $\mathsf{u}_i^{\uparrow}(\cdot)$ as in (1). The second model (NSE-2) transforms feature maps into hidden layers and refines them as the resolution increases, resembling the approach in DCGAN [28]. See Appx. C for details of both models.

There are other nonlinear models—for example, one that concatenates a generative model with a full-fledged semantic segmentation model (such as UNet [29]). However, such a model provides no clue about how feature maps encode image semantics. We therefore choose not to consider them in our studies.

3.2. Experiment Setup

We conduct experiments on various GANs and datasets to examine our LSE model. We choose PGGAN [20], StyleGAN [21], and StyleGAN2 [22] trained on three datasets. Specifically, we use StyleGAN trained on the facial image dataset CelebAHQ [25], and StyleGAN2 trained on FFHQ [21] and separately on a bedroom and church dataset from LSUN [37]. Instead of training those GAN models from scratch, we use the existing pretrained GANs[2].

Training LSE is supervised by pretrained semantic segmentation models. For facial images, we use a UNet trained on CelebAMask-HQ [24] with manually labeled semantic masks, and it segments a facial image into 15 semantic regions. For bedroom and church images, we use the publicly

[2]These pretained GANs are publicly available here.

available DeepLabV3 [8] trained on ADE20K [42] dataset. DeepLabV3 predicts 150 classes, most of which are not present in the GAN's output images. Thus, we consider a subset of classes for generated bedroom and church images. Our choice of the classes is described in Appx. E.

In each experiment, the training data of LSE consists of 51,200 images sampled by a GAN model (i.e., PGGAN, StyleGAN, or StyleGAN2), and the images are semantically labeled by a pretrained segmentation model. Meanwhile, we record the GAN model's feature maps in each image generation. The semantic masks together with the feature maps are then used to train the transformation matrix $\mathbf{T}_i$ for every GAN layer (see Appx. D for more details).

After training, we evaluate our LSE on a separate set of 10,000 generated images. During the generation of each image, we use LSE (and NSE-1 and NSE-2 for comparison) on the generator's feature maps to predict a semantic segmentation, which is in turn compared with the segmentation labels to compute an IoU score (defined in Appx. A).

3.3. Results

We now present empirical results to back our proposed linear semantic extraction (1).

Evaluation of LSE. Figure 3 compares qualitatively semantic segmentation of LSE to other methods. The quantitative results in terms of mIoU scores are reported in Table 1, from which it is evident that our simple LSE is comparable to more complex, nonlinear semantic classifiers. The relative performance gap between LSE and NSEs (NSE-1 and NSE-2) is within 3.5%. Results on StyleGAN-Bedroom and StyleGAN-Church have a slightly larger gap ($< 8\%$). We present additional qualitative results and IoU for each category in Appx. J.

Takeaway. Our experiments show that LSE is capable of extracting image semantics from the feature maps of the GANs. Further, the close performance of LSE to NSEs suggests that a well-trained GAN encodes the image semantics in its feature maps in a linear way.

Our approach differs from the prior GAN Dissection work [5], which identifies feature maps correlating with a specific semantic class. These feature maps are primarily found in middle-level feature maps, resulting in a lower resolution segmentation than the network output. Also, the per-pixel semantic classification remains unexplored. In contrast, the semantics extracted by LSE are of high resolution (the same as the output image) and have sharp boundaries.

Geometrical evidence. The geometric interpretation of (1) indicates that features of a semantic class fall into a convex surface area on an n-sphere. To verify this intuition, we test a stronger hypothesis—the features of individual pixels can be clustered around class centers. If the clusters are well formed, we should be able to find a convex hull to identify individual classes.

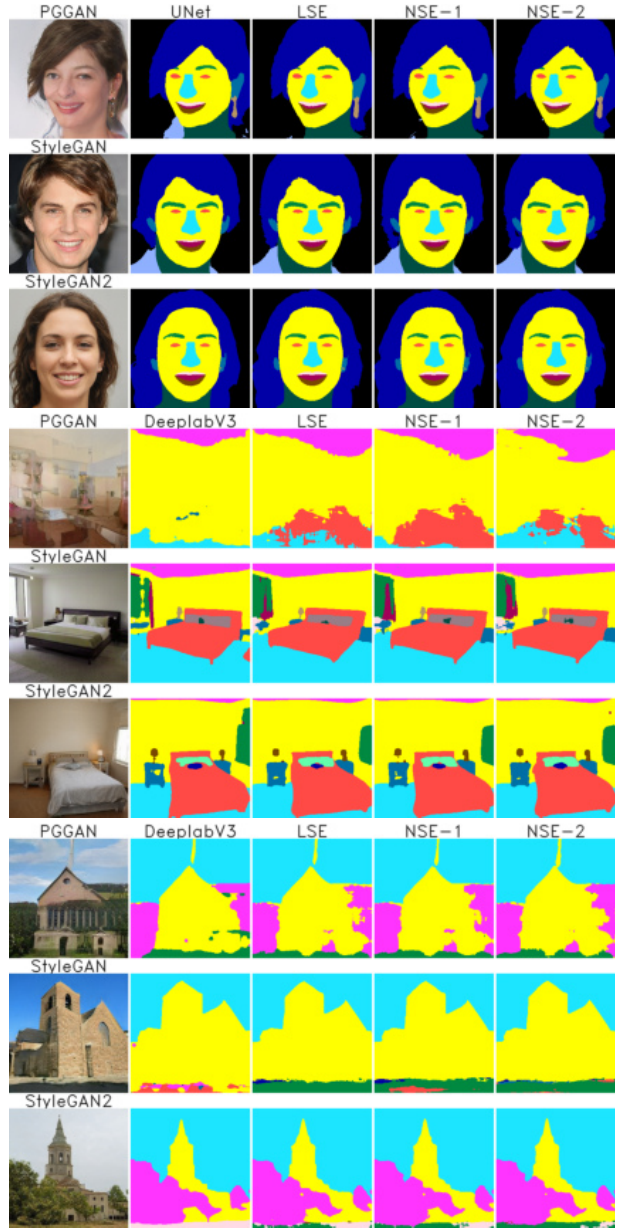


Figure 3. Qualitative comparison of LSE, NSE-1 and NSE-2. From top to bottom, every 3 rows are from GAN models trained on the same dataset (face, bedroom, church images, respectively). Images are sampled randomly rather than cherry-picked.

To estimate the class centers, we randomly generate 3000 images using StyleGAN-CelebAHQ, and obtain their semantic masks using UNet. All per-pixel feature vectors from the same semantic class are collected and normalized onto the unit n-sphere. The vectors are then averaged and renormalized on the n-sphere. The resulting vector is then treated as a class center to determine each pixel's semantic class. Some segmentation results are shown in Fig. 4, suggesting that this approach indeed segments images reasonably. The segmentation error (e.g., in Fig. 4) may be attributed to the inaccurate

Dataset	PGGAN CelebAHQ	PGGAN Bedroom	PGGAN Church	StyleGAN CelebAHQ	StyleGAN Bedroom	StyleGAN Church	StyleGAN2 FFHQ	StyleGAN2 Bedroom	StyleGAN2 Church
LSE	65.5 (-1.6)	33.2 (-3.2)	51.3 (-3.2)	69.1 (-1.9)	39.9 (-7.8)	35.4 (-6.3)	79.7 (-1.7)	53.9 (-3.4)	37.7 (-2.6)
NSE-1	**66.5**	**34.3**	**53.0**	**70.5**	**43.3**	**37.8**	**81.0**	**55.8**	**38.7**
NSE-2	65.9 (-0.9)	30.7 (-10.5)	49.5 (-6.6)	70.1 (-0.5)	38.9 (-10.2)	34.0 (-10.1)	80.2 (-1.1)	52.1 (-6.8)	35.3 (-8.8)

Table 1. The mIoU (%) of LSE, NSE-1, and NSE-2 trained with off-the-shelf semantic segmentation models (UNet for CelebAHQ and FFHQ, DeepLabV3 for bedroom and church dataset). "Bedroom" and "Church" images are subsets of the LSUN [37] dataset. The numbers in brakets are the performance difference relative to the best model highlighted in bold.

Figure 4. Forging LSE's parameter $\mathbf{T}$ using the statistical centers of features. Experiment is done on StyleGAN-CelebAHQ.

boundaries between classes, as they are not explicitly trained to separate different semantic classes. Nevertheless, this experiment confirms our geometric intuition about the feature maps' linear embedding of semantics.

We further compute the statistics of cosine similarities of feature vectors that are within the same semantic class and that are in different classes. We show that feature vector's cosine similarities between pixels within the same class are indeed higher. Details are reported in Appx. F.

Takeaway. Statistical centers of feature vectors can segment images reasonably well, suggesting that feature vectors from different classes are well separated on the n-sphere. The relatively large cosine similarities between different classes also backs our intuition. These are further evidence indicating linear encoding of image semantics in the feature maps of GANs.

Few-shot LSEs. Thanks to the linearity of LSE, we can also train it using a small number of examples. We refer to this approach as the *few-shot* LSE. Here, we experiment the training of LSE with only 1, 4, 8, and 16 annotated images, respectively. For each few-shot LSE setup, the training is repeated for five times, and we report the average performance of the five trained models.

Table 2 reports the quantitative evaluation results. First, the extreme case, one-shot LSE, already shows plausible performance, achiving 69.8%, 39.8%, and 52.5% mIoU scores

N	FFHQ	Bedroom	Church
1	55.6 (69.8) $\pm$ 5.2	21.5 (39.8) $\pm$ 3.7	19.7 (52.2) $\pm$ 3.4
4	64.8 (81.4) $\pm$ 1.0	36.5 (67.8) $\pm$ 2.7	24.2 (64.3) $\pm$ 1.4
8	68.4 (85.8) $\pm$ 2.6	38.6 (71.6) $\pm$ 2.4	26.3 (69.7) $\pm$ 0.8
16	70.2 (88.1) $\pm$ 3.0	42.2 (78.3) $\pm$ 1.1	27.7 (73.5) $\pm$ 0.8
full	79.7%	53.9%	37.7%

Table 2. The evaluation of few-shot LSEs for StyleGAN2. Each model is trained 5 times. Both the mean and maximum deviation of the 5 repeats are shown. The numbers in parentheses indicate the ratio of the mean performance over the fully trained model's performance listed in the last row.

relative to the fully trained model. The 16-shot LSE further improves the mIoU scores to 88.1%, 78.3%, and 73.5% relative to the fully trained model.

Takeaway. The few-shot LSEs has already the performance comparable to fully supervised LSEs. Not only do they enable a low-cost way of extracting semantics from GANs, the results further support our hypothesis that image semantics are linearly embedded in feature maps.

4. Applications

In this section, we leverage the simplicity of LSE to control image semantics of GAN's generation process.

4.1. Few-shot Semantic Editing

In many cases, the user may want to control a GAN's image generation process. For example, they might want to adjust the hair color of a generated facial image from blond to red; and the user may draw a red stroke on the hair to easily specify their intent. Existing approaches, such as color space editing [44, 7, 2, 43], aim to find a latent vector that generates an image better matching the user specification. The latent vector is often found by minimizing a distance measure between the generated image and the user's strokes in color space.

However, without explicit notion of semantics, the minimization process may not respect image semantics, leading to undesired changes of shapes and textures. For example, in the 2nd row and 2nd and 3rd columns of Fig. 5, the user wishes to remove the hair in generated images, but the color space editing methods tend to just lighten the hair color rather than removing it.

Leveraging LSE, we propose an approached called Semantic Image Editing (SIE) to enable semantic-aware image generation. We define a semantic edit loss $L_s = \mathcal{L}(P(G(z)), Y)$, where $\mathcal{L}(\cdot)$ is the cross-entropy loss, Y is the target semantic mask, and G is the generator that takes the latent vector z as input. P is a pretrained segmentation model such as our LSE. Starting from an image's latent vector z, we find an output image's latent vector z' by minimizing the loss. The details are presented in Appx. G.

Here, we compare the results of the method using different segmentation models, including UNet, our 8-shot LSE, and fully trained LSE. The qualitative results are shown in Fig. 5. For each instance, we include both the results of

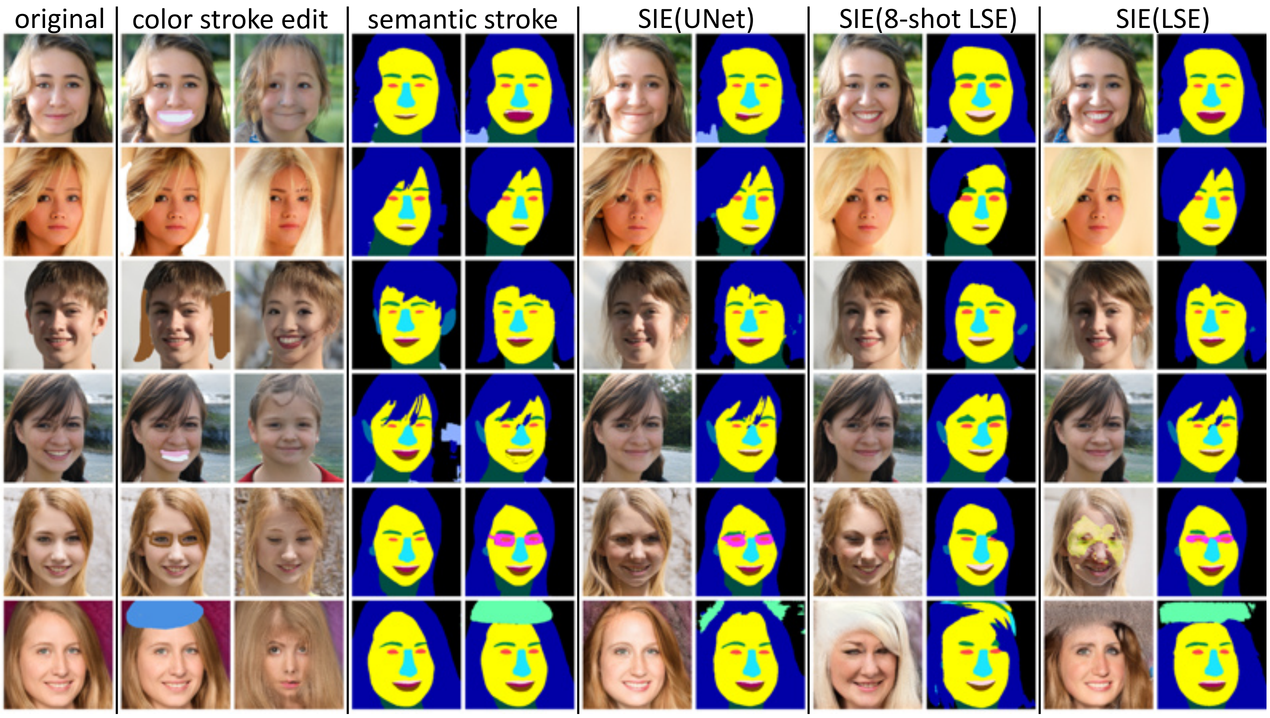


Figure 5. Results of Semantic Image Editing (SIE) on StyleGAN2-FFHQ. Images edited with color strokes are shown in col 2 and 3. Col 4 and 5 show LSE's original segmentation mask and the user-editied semantic masks. The rest of columns show the results of SIE(UNet), SIE(8-shot LSE), and SIE(LSE), respectively.

color-space editing and semantic editing, and more results are presented in Appx. J.

First, SIE(UNet) controls image generation and better preserves semantics than color-space editing. In comparison to the results of color-space editing, the undesired changes in output images are greatly reduced, although SIE(UNet) may still fail to transform the image's semantics: for instance, in the 1st and 2nd row of Fig. 5, SIE(UNet) barely changes the original image. We speculate that this is because the gradient from L_s is carried through the entire UNet, making the optimization process more difficult.

SIE(8-shot LSE) edits the semantics better than SIE(UNet): it preserves the semantic regions not intended by the user. However, in 5th and 6th row, SIE(8-shot LSE) produces lower image quality. We conjecture that this is due to highly unbalanced data distribution: the semantic classes "hat" and "eyeglasses" occur sparsely in the training dataset. As a result, those classes can not be well represented in the GAN model—leading to the well-known mode collapse problem. Lastly, SIE(8-shot LSE) has a similar performance to SIE(LSE), although its LSE is trained with much fewer annotations.

4.2. Few-shot Conditional Generation

Semantic-Conditional Sampling (SCS) aims to synthesize an image subject to a semantic mask. It offers the user more control over the image generation process. SCS has been explored [27, 46], but most previous works rely on large annotated datasets to train their models. Thanks to its simplicity, our LSE can be trained with a small set of annotated images (recall our few-shot LSE). Here we leverage it to build a few-shot SCS model. It is the need of only a few labeled images that differs our method from existing image-to-image translation methods [18, 45, 46, 27, 35].

Our few-shot SCS finds output latent vector by formulating a minimization problem similar to SIE discussed in Sec. 4.1. But unlike SIE, which takes the input image's latent vector, it needs to choose a proper initial latent vector. The details of the minimization process are presented in Appx. H.

Qualitative results. Figure 6 shows conditionally sampled images using 8-shot LSEs. The eight image labels are produced by a pretained semantic segmentation model: for facial images, we use UNet; and for bedroom and church images, we use DeepLabV3.

While the generated images are diverse, they all respect well the provided semantic targets (first column of Fig. 6). For facial image generation (1st to 4th row), faces are well matched to the provided semantic masks. For bedroom images (5th to 8th row), the location and orientation of beds, windows, and walls all match well to the target semantic layout. For church images (9th to 12th row), the church geometries are mostly matched. For instance, in the first group of church images, three of the five samples have two tall towers and one short tower in between. In the second

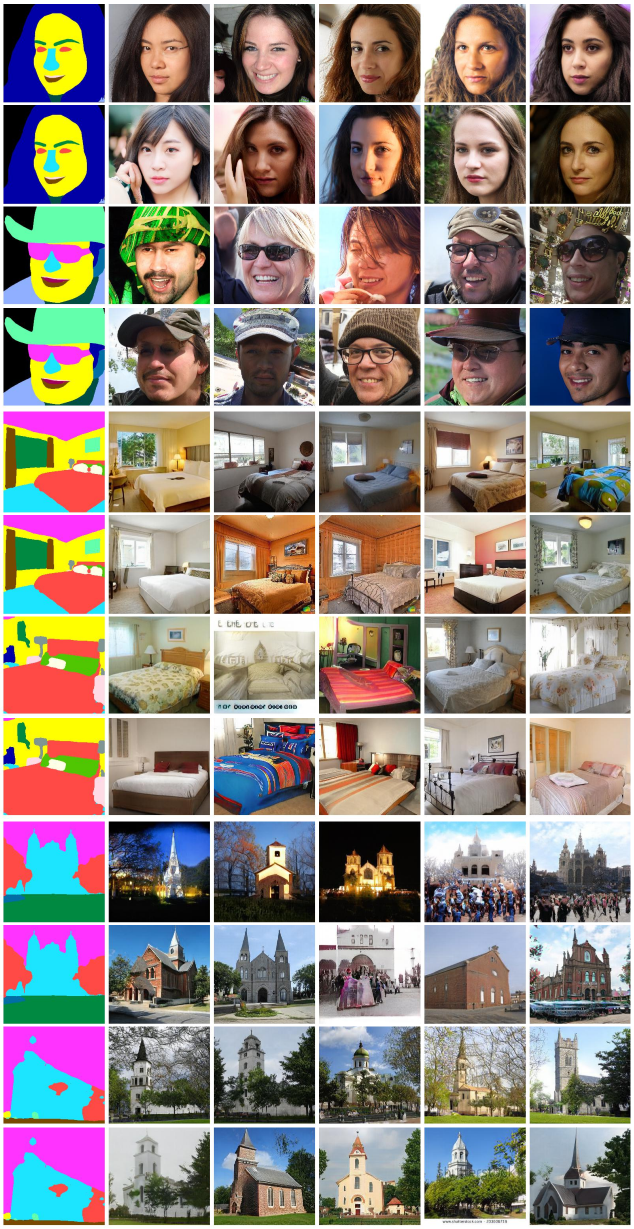


Figure 6. SCS results on StyleGAN2. Every pair of rows are to compare SCS(8-shot LSE) results (shown in the first row of each pair) with SCS using a pretrained segmentation model (shown in the second row of each pair).

group, all the samples have a tall tower on the left side, matching the provided semantic mask.

The SCS(8-shot LSE) and the baseline SCS(UNet) have comparable semantic quality in generated images. We notice that in 3rd and 4th rows, SCS(8-shot LSE) appears less successful than SCS(UNet) to respect the provided semantic targets. We believe that this is again due to the data imbalance: "hat" and "eyeglasses" occur must less frequently in the training dataset than other semantic classes.

N	Church	Bedroom	FFHQ
1	16.0 ± 1.4	17.5 ± 2.0	37.2 ± 0.8
4	18.0 ± 1.3	21.6 ± 0.9	39.1 ± 0.5
8	19.6 ± 0.5	21.7 ± 0.8	39.4 ± 0.9
16	20.4 ± 0.6	22.3 ± 0.4	40.0 ± 0.2
baseline	23.1	17.3	34.3

Table 3. The semantic accuracy measures the semantic agreement between generated images and targets. For SCS with few-shot LSEs, each model is trained for 5 times with different training data to account for the training data variance. The numbers before $\pm$ sign are the average results of the 5 repeats, and the numbers following $\pm$ indicate the maximum deviations from the average. Experiments are done on StyleGAN2.

Quantitative results. We compute the semantic accuracy of SCS, which measures the discrepancy between the semantic target and the segmentation of a generated image. We present the formal definition of the accuracy in (7) of the appendix, and report the results in Table 3. On the church dataset, the SCS(few-shot LSE) performs slightly worse than SCS(UNet), while on the bedroom and face datasets, our method with 8-shot (and 16-shot) LSE is even better than SCS(UNet).

5. Conclusions

In this work, we study how the image semantics are embedded in GAN's feature maps. We propose a Linear Semantic Extractor (LSE) to extract image semantics modeled by GANs. Experiments on various GANs show that LSE can indeed reveal the semantics from feature maps. We also study the class centers and cosine similarities between different classes to provide geometric interpretation of our LSE. Therefore, it is well-backed that GANs use a linear notion to encode semantics. Then, we successfully train LSEs in few-shot settings. Using only 16 training annotations, we obtain 73.5%, 78.3%, and 88.1% performance relative to fully supervised LSEs on the church, bedroom, and face images. Finally, we build two novel applications based on few-shot LSEs: the few-shot Semantic-Conditional Sampling and the few-shot Semantic Image Editing. Our methods can match or surpass the baselines using fully supervised segmentation networks. Using the proposed methods, users can exert precise and diverse spatial semantic controllability over pretrained GAN models with only a few annotations.

Acknowledgements. This work was supported in part by the National Science Foundation (1910839, 1717178, 1816041). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or others.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? *CoRR*, abs/1911.11544, 2019. 2

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? 2019. 6

[3] Rameen Abdal, Peihao Zhu, Niloy Mitra, and Peter Wonka. StyleFlow: Attribute-conditioned Exploration of StyleGAN-Generated Images using Conditional Continuous Normalizing Flows. *arXiv e-prints*, page arXiv:2008.02401, Aug. 2020. 2

[4] David Bau, Steven Liu, Tongzhou Wang, Jun-Yan Zhu, and Antonio Torralba. Rewriting a deep generative model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[5] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B Tenenbaum, William T Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. *arXiv preprint arXiv:1811.10597*, 2018. 1, 2, 5

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 2

[7] A. Brock, T. Lim, J. M. Ritchie, and N. Weston. Neural photo editing with introspective adversarial networks. *ArXiv*, abs/1609.07093, 2017. 2, 6

[8] Liang Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. 2017. 3, 5, 1

[9] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. *arXiv e-prints*, page arXiv:1606.03657, June 2016. 2

[10] Edo Collins, Radhakrishna Achanta, and Sabine Susstrunk. Deep feature factorization for concept discovery. In *The European Conference on Computer Vision (ECCV)*, 2018. 2

[11] Edo Collins, Raja Bala, Bob Price, and Sabine Süsstrunk. Editing in style: Uncovering the local semantics of GANs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[12] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015. 1

[13] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016. 2

[14] Partha Ghosh, Pravir Singh Gupta, Roy Uziel, Anurag Ranjan, Michael Black, and Timo Bolkart. Gif: Generative interpretable faces, 2020. 2

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1, 2

[16] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. 2017. 2

[17] Xun Huang, Yixuan Li, Omid Poursaeed, John Hopcroft, and Serge Belongie. Stacked generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5077–5086, 2017. 1

[18] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017. 2, 7

[19] Ali Jahanian, Lucy Chai, and Phillip Isola. On the"steerability" of generative adversarial networks. *arXiv preprint arXiv:1907.07171*, 2019. 2

[20] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 1, 2, 4

[21] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 1, 2, 4, 6

[22] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *arXiv preprint arXiv:1912.04958*, 2019. 1, 2, 4

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1, 2

[24] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: towards diverse and interactive facial image manipulation. *arXiv preprint arXiv:1907.11922*, 2019. 4, 1, 6

[25] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 1, 4

[26] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional Image Synthesis With Auxiliary Classifier GANs. *arXiv e-prints*, page arXiv:1610.09585, Oct. 2016. 2

[27] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Gaugan: semantic image synthesis with spatially adaptive normalization. In *ACM SIGGRAPH 2019 Real-Time Live!*, pages 1–1. 2019. 2, 7

[28] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 1, 2, 4

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1, 3, 4

[30] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. *arXiv preprint arXiv:1907.10786*, 2019. 2

[31] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. 2020. 2

[32] Ryohei Suzuki, Masanori Koyama, Takeru Miyato, Taizan Yonetsuji, and Huachun Zhu. Spatially Controllable Image Synthesis with Internal Representation Collaging. *arXiv e-prints*, page arXiv:1811.10153, Nov. 2018. 2

[33] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images, 2020. 2

[34] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. *arXiv preprint arXiv:2002.03754*, 2020. 2

[35] 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. 2017. 2, 7

[36] Ceyuan Yang, Yujun Shen, and Bolei Zhou. Semantic hierarchy emerges in deep generative representations for scene synthesis. *arXiv preprint arXiv:1911.09267*, 2019. 2

[37] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 1, 4, 6

[38] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018. 2

[39] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Mueller, and R and Manmatha. Resnest: Split-attention networks. 2020. 3, 2

[40] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915, 2017. 2

[41] Yuxuan Zhang, Wenzheng Chen, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. Image gans meet differentiable rendering for inverse graphics and interpretable 3d neural rendering, 2020. 2

[42] Bolei Zhou, Hang Zhao, Francesco Xavier Puig Fernandez, Sanja Fidler, and Antonio Torralba. Scene parsing through ade20k dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5, 2

[43] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. 2020. 6

[44] Jun Yan Zhu, Philipp Krhenbühl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold. 2016. 2, 6

[45] Jun Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. 2017. 2, 7

[46] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. 2019. 2, 7