Cluster-guided Image Synthesis with Unconditional Models

Markos Georgopoulos
Imperial College London
m.georgopoulos@imperial.ac.uk

James Oldfield
Queen Mary University of London
j.a.oldfield@qmul.ac.uk

Grigorios G Chrysos
EPFL
grigorios.chrysos@epfl.ch

Yannis Panagakis
University of Athens
yannisp@di.uoa.gr

Abstract

Generative Adversarial Networks (GANs) are the driving force behind the state-of-the-art in image generation. Despite their ability to synthesize high-resolution photo-realistic images, generating content with on-demand conditioning of different granularity remains a challenge. This challenge is usually tackled by annotating massive datasets with the attributes of interest, a laborious task that is not always a viable option. Therefore, it is vital to introduce control into the generation process of unsupervised generative models. In this work, we focus on controllable image generation by leveraging GANs that are well-trained in an unsupervised fashion. To this end, we discover that the representation space of intermediate layers of the generator forms a number of clusters that separate the data according to semantically meaningful attributes (e.g., hair color and pose). By conditioning on the cluster assignments, the proposed method is able to control the semantic class of the generated image. Our approach enables sampling from each cluster by Implicit Maximum Likelihood Estimation (IMLE). We showcase the efficacy of our approach on faces (CelebA-HQ and FFHQ), animals (Imagenet) and objects (LSUN) using different pre-trained generative models. The results highlight the ability of our approach to condition image generation on attributes like gender, pose and hair style on faces, as well as a variety of features on different object classes.

1. Introduction

Generative Adversarial Nets (GANs) [8] have demonstrated photo-realistic generation quality by utilizing the rich corpus of available image datasets. Despite their success, the value they can add as data generation tools is currently limited by the lack of control in the synthesized content. In the typical GAN setting an image is synthesized by sampling a vector from a latent distribution and performing a forward pass through a generator network. However, random sampling from the latent distribution provides no control over semantic attributes in the image space. Such control over the generated characteristics is vital for tasks like autonomous driving [39] or (inverse) reinforcement learning [12].

A common solution to the problem of controllable generation is to introduce supervision in the form of class labels [1, 5, 28]. This process requires the annotation of the training set, which can be a resource-intensive task, in addition to being impractical for a continually-growing number of attributes of interest. Additionally, even with a rich and diverse annotated dataset, training a conditional generative model that can balance control and photo-realism is a non-trivial task that requires tailored engineering tricks (e.g., truncation trick [1]).

In this work, we introduce a method that can be implemented on top of any pretrained GAN to introduce control without the need for labels and supervision. The method relies on the clusters that are formed in the intermediate representation space of a generator. We posit that the representational capacity of the network allows for semantic attributes, like hair color and pose, to be disentangled in this representation space. Hence, each of the formed clusters corresponds to a different semantic attribute. This assumption enables us to control image generation by conditioning on the cluster assignment. Latent sampling from these clusters is achieved via Implicit Maximum Likelihood Estimation (IMLE). The proposed framework is summarized in Figure 1. We benchmark the method against GANs that learn clustering in the latent space as well as methods for interpretable directions in pretrained GANs. The results highlight the efficacy of our method in consistently generating images of desired attributes.

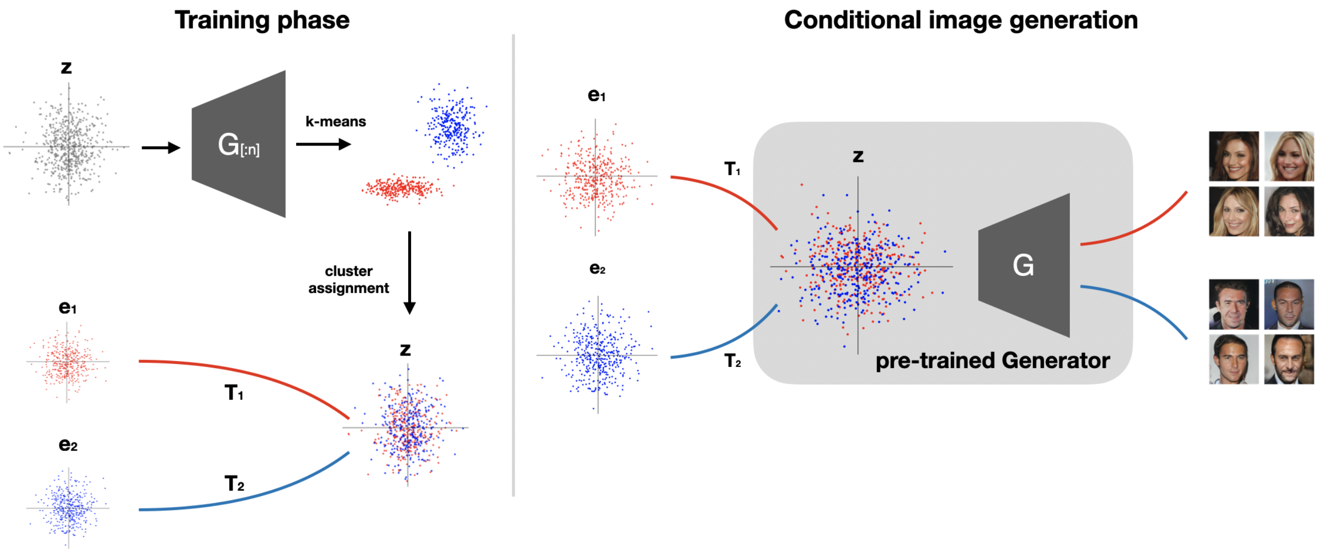

Figure 1. Conditional generation using an unsupervised generator. The training phase (depicted on the left) includes the following steps: (a) latent codes are sampled from the latent space of the generator, and then (b) passed through the first n layers of the generator. The resulting representations are then clustered using k-means. Thus, we can assign each sampled latent code to a cluster (in the representation space). (c) Sequentially, we can learn a mapping from an auxiliary distribution e_c to the subspace of each cluster in the latent space of the generator. In the testing phase (depicted on the right), we can sample from the auxiliary distributions and use the corresponding mappings (T_1 or T_2) to synthesize images that have specific semantic attributes, e.g., male or female.

2. Related work

Generative Adversarial Nets (GANs) [8] are able to synthesize diverse and photo-realistic images [1, 17, 18]. Introducing structure in the latent space of the generator is an active area of research. Different distributions have been proposed to enforce this structure. Specifically, a Cauchy distribution [22], a mixture model [9], a parametric distribution based on tensor decompositions [7, 20], or non-parametric distributions [37] have been used in this context. The goal is to primarily improve either the training of GANs [9] or the synthesized image quality [22, 37]. Our method relies on a trained generator instead, hence it could utilize any of the aforementioned modifications on the latent space.

Disentanglement of the latent space: The topic of disentangling the factors of variation in the latent space has sparked the interest of the community. InfoGAN [4] is the first effort to augment GAN to achieve unsupervised disentanglement. InfoGAN uses auxiliary codes $\boldsymbol{\psi}$ and maximizes the mutual information of the codes with the synthesized image. The idea has since been extended in [25, 26]. However, in [25] the authors explain why the success of disentanglement relies heavily on design choices and inductive biases in the network, making ideas such as InfoGAN sensitive to the choice of the architecture. The works of [15, 21, 24] also rely on modifying the GAN architecture with codes $\boldsymbol{\psi}$. In [24], they rely on pairwise differences between elements ψ_i; in [21] a beta-VAE [11] provides the codes $\boldsymbol{\psi}$, while in [15] the codes are provided by a tree-structure.

Due to the challenging nature of unsupervised disentanglement often some (weak) labeling is used. In [36], the authors use bounding boxes as a weak supervision signal to disentangle the background from the foreground information in synthesized images. Supervised disentanglement has also been used in various tasks [38, 45]. However, in our work we do not utilize any type of labels for training.

Interpretable directions in GANs: Beyond the aforementioned methods, the discovery of interpretable latent directions in a pretrained generator is an active area of study. A dataset of trajectories in the latent space is created in [33]. Such trajectories correspond to known transformations in the data space; the method searches for simple transformations encoded. In [41], they use two latent codes (one is shifted version of the other); the authors synthesize the two images and then learn a dense layer to predict the shift in the codes. Harkonen et al. [10] and Shen et al. [35] find the principal directions of variation using Principal Component Analysis (PCA), either in the latent space or the weights of the first layer. Similarly, Tensor Component Analysis is utilized in [31] to better separate style and geometry. The drawbacks of such methods is that they provide interpretable directions relative to the input image; that is given a single latent code, they can find some directions that transform (e.g., rotate) the generated image. Similarly, Jahanian et al. [13] propose to use self-supervision to learn directions for simple automatically obtained transformations (e.g., shifts, zoom, rotation), while Collins et al. [6] utilize clustering across only the channel dimension in order to discover and edit semantic regions in the pixel-space. On the contrary, our method learns to directly sample an image with a desired attribute from noise without editing.

Clustering in the latent space: A long line of research takes advantage of the clusters that are formed in the feature space of convolutional representations. The majority of such works focus on unsupervised/self-supervised techniques for discriminative tasks, e.g., [2,3,42,43,46], while a number of works apply similar techniques to generative modelling [7,27,29]. More closely related to this work, clustering has been utilized in the latent space of GANs [27,29] to generate diverse images. Mukherjee et al. [29] utilize an auxiliary encoder to predict the cluster assignments. On the other hand, Liu et al. [27] use the features from the discriminator to cluster the images. The proposed approach is orthogonal to these methods since it works on a pretrained generator and is not trained end-to-end.

3. Method

In this section, we motivate our approach and present the proposed method that enables conditional generation using GANs that are not trained with attribute supervision. Firstly, we introduce briefly image synthesis using GANs. We continue by motivating our assumption regarding the clustering that occurs in the representation space. Lastly, we present our method for performing conditional image generation using IMLE.

3.1. Image synthesis with GANs

Generative Adversarial Networks (GANs) [8] utilize a set of images $\mathcal{S} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N]$ to synthesize new images that resemble the data in $\mathcal{S}$. Image synthesis is enabled by sampling a latent code $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, where $\boldsymbol{I}$ is the identity matrix, and passing the code though a generator network G. In GANs, in addition to the generator, an auxiliary "discriminator" network is used for the optimization. Specifically, GANs are trained using a minimax formulation between the generator G and the discriminator D. The discriminator is trained to distinguish the synthetic images $\tilde{\boldsymbol{x}} = G(\boldsymbol{z})$ from the real images in $\mathcal{S}$. Concretely, we denote with $p_{\mathcal{S}}$ the data distribution and with $p_{\boldsymbol{z}}$ the distribution for sampling the latent codes (e.g., a normal distribution). Then the learning objective can be formulated as:

$$\min_{\boldsymbol{w}_G} \max_{\boldsymbol{w}_D} \mathcal{L}_{GAN}(G, D) = \mathbb{E}_{\boldsymbol{x} \sim p_{\mathcal{S}}}[\log D(\boldsymbol{x}; \boldsymbol{w}_D)] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}}[\log(1 - D(G(\boldsymbol{z}; \boldsymbol{w}_G); \boldsymbol{w}_D))] \tag{1}$$

where the $\boldsymbol{w}_G$ and the $\boldsymbol{w}_D$ are the learnable parameters of the generator and the discriminator networks respectively. After training, image synthesis is performed by sampling from $p_{\boldsymbol{z}}$ and passing the code through G, which progressively increases the higher frequency content at each layer.

3.2. Clustering in the representation space

The core operation behind state-of-the-art GANs is convolution. This is due to the inductive bias of the operator that allows for great generalization power of these networks in the image domain. This inductive bias is so effective that even randomized convolutional neural networks (CNNs) can produce useful image representations (classification accuracy of 12% on Imagenet in [30], while random chance is at 0.1%). A number of works [2,3] focus on leveraging the clustering that occurs in the representation space of CNNs for downstream tasks. Contrary to this line of research, this work focuses on the clustering that separates the representation space of intermediate layers of generative CNNs (i.e., GANs).

In particular, we posit that images are clustered in the representation space of the generator according to semantic attributes, e.g., geometric features. In the same vein, for a generator G with a hierarchical architecture (e.g., Progressive GAN [16]) different layers should capture different attributes. To this end, we perform clustering on the representation $G_{[:n]}(\boldsymbol{z})$ of the n^{th} layer and separate the space into $C \in \mathbb{N}$ clusters. In this work, we use k-means, although any clustering technique would perform in a similar manner. By manually assigning a semantic attribute to each cluster, we can perform synthesis of a specific attribute by conditioning image generation on a specific cluster.

3.3. Implicit Maximum Likelihood Estimation

Given the latent vector $\boldsymbol{z}$, the transformation $G_{[:n]}(\boldsymbol{z})$ to the clusters in the representation space is deterministic, thus we have a direct assignment of each latent vector $\boldsymbol{z}$ into a cluster c with $c \in \{1, 2, \ldots, C\}$. We denote the latent vector corresponding to cluster c as $\boldsymbol{z}^c$. The codes $\boldsymbol{z}^c$ do not form clear clusters in the latent space nor do they follow a known probability distribution. Hence, sampling from $\boldsymbol{z}^c$ to perform conditional generation is non-trivial. This step is crucial in order to effectively sample from the observed clusters in the representation space, and proceed with the forward pass to an image of defined attributes. To this end, we utilize an auxiliary (normal) distribution $\boldsymbol{e}^c$ and obtain a mapping $\boldsymbol{e}^c \mapsto \boldsymbol{z}^c$ using Implicit Maximum Likelihood Estimation (IMLE).

IMLE is a non-adversarial method that learns a mapping T between two distributions. Li et al. [23] show that the method is equivalent to maximizing the likelihood under some assumptions. We utilize IMLE to learn a mapping from the auxiliary distribution (which is known, e.g., a Gaus-

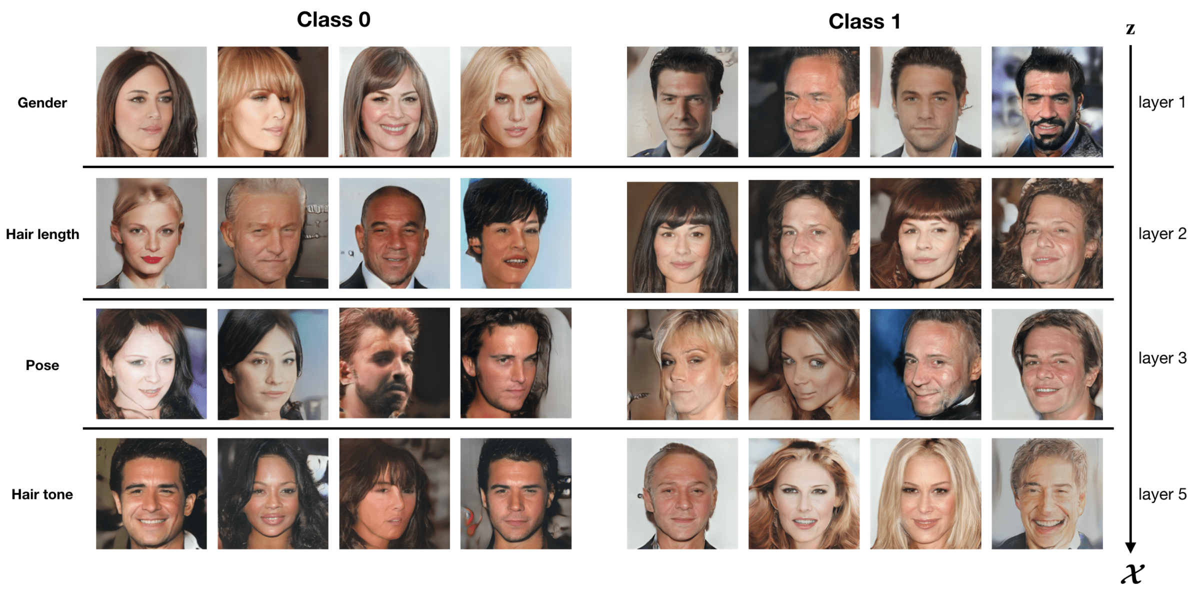

Figure 2. The mappings that are learned for clusters in different layers correspond to different semantic attributes. At each level of the PGAN generator we learn the two depicted clusters. Naturally, some of the semantic attributes are entangled, e.g., hair tone and background.

sian distribution) to the subspace spanned by all the latent vectors $\boldsymbol{z}^c$ corresponding to a specific cluster c. By learning a mapping T_c for every cluster c, we are able to sample from the auxiliary Gaussian distribution, and obtain a synthesized image with the semantic attribute of cluster c.

Next, we elaborate on the training procedure for the mappings T_c, which includes the following steps:

1. Firstly, we sample $\Gamma \in \mathbb{N}$ vectors $\boldsymbol{e}^c$ from the auxiliary distribution for cluster c and apply the transformation T_c to obtain the latent codes $\tilde{\boldsymbol{z}}^c$, i.e., $\tilde{\boldsymbol{z}}^c_\gamma = T_c(\boldsymbol{e}^c_\gamma)$ for $\gamma = 1, 2, \dots, \Gamma$.

2. For each latent vector $\boldsymbol{z}^c_i$, we aim to minimize the Euclidean distance of $\boldsymbol{z}^c_i$ and $\tilde{\boldsymbol{z}}^c_\gamma$, i.e., we perform a nearest neighbor search on the vectors $\boldsymbol{e}^c_\gamma$. That is expressed as:
$$\boldsymbol{e}^c_i = \underset{\boldsymbol{e}^c_\gamma, \gamma=1,2,\dots,\Gamma}{\arg\min} \|\boldsymbol{z}^c_i - T_c(\boldsymbol{e}^c_\gamma)\|^2_2. \tag{2}$$

3. The last step consists in optimizing the mappings T_c. The approximate matches of the last step are used to optimize the transformations. Concretely:
$$\tilde{T}_c = \underset{T_c}{\arg\min} \sum_i \|\boldsymbol{z}^{\boldsymbol{c}}_{\boldsymbol{i}} - T_c(\boldsymbol{e}^{\boldsymbol{c}}_{\boldsymbol{i}})\|^2_2. \tag{3}$$

The steps are repeated until convergence of all mappings T_c.

After training the mapping functions for each cluster, conditional sampling for each semantic attribute can be performed by utilizing the corresponding mapping, i.e., $G(\boldsymbol{z}, c) = G(T_c(\boldsymbol{e}^c))$. The training and testing phases of the proposed framework are summarized in Figure 1 and Algorithm 1.

Algorithm 1: Algorithm for the proposed method

```
Result: A set of mappings $\{T_1, \dots, T_C\}$
$\mathbf{z} \leftarrow$ Sample from the latent distribution of GAN
$\mathbf{y} \leftarrow \mathbf{G}_{[:\mathbf{n}]}(\mathbf{z})$
Initialize parameters $\theta_c$ of $T_c, c \in \{1, \dots, C\}$
for c in 1...C do
    for number of epochs do
        $\boldsymbol{e}^c \leftarrow$ Sample from the normal distribution
            for cluster  c
        $\boldsymbol{z}^{\boldsymbol{c}} \leftarrow$ Latent codes belonging to cluster $c$
        for $\boldsymbol{z}^{\boldsymbol{c}}_{\boldsymbol{i}}$ in $\boldsymbol{z}^{\boldsymbol{c}}$ do
            $\boldsymbol{e}^{\boldsymbol{c}}_{\boldsymbol{i}} \leftarrow \arg\min_{\boldsymbol{e}^c} \|\boldsymbol{z}^c_i - T_c(\boldsymbol{e}^c)\|^2_2$
        end
        for number of batches do
            // SGD
            $\theta_c = \theta_c - \lambda_t \nabla_\theta \mathrm{MSE}(T_c(\boldsymbol{e}^c_i), \boldsymbol{z}^c_i)$.
        end
    end
end
```

Using IMLE to learn the mapping from the auxiliary distribution to the latent codes of the generator yields a number of benefits. For example, using a GAN for this task would suffer from unstable training as well as mode-collapse. On the other hand, IMLE ensures support for every point in the training set. However, using IMLE to generate images directly with an L2 loss would result in blurry images. We

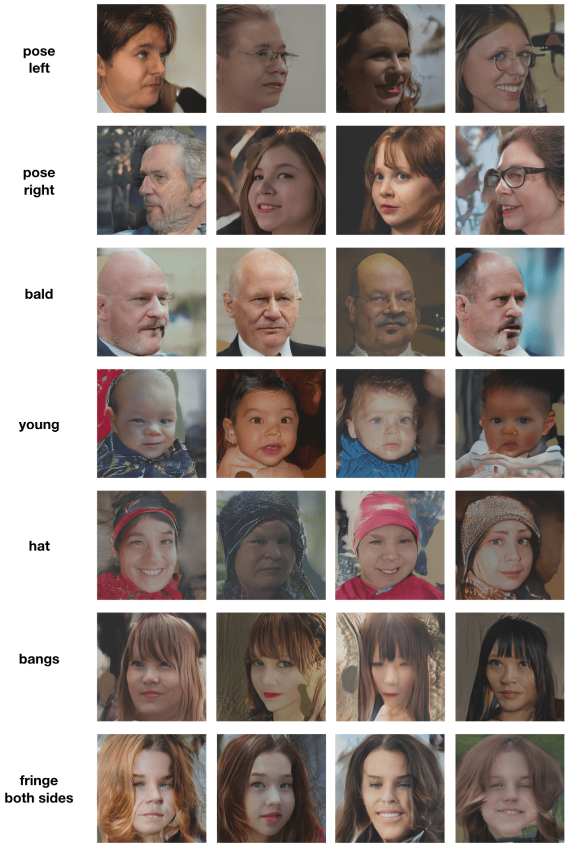

Figure 3. Each row depicts images synthesized from a different cluster in the representations of StyleGAN. The semantic attribute of each cluster is denoted on the left of the image. Note that besides primitive features like pose, the method captures several high-level attributes such as hat or bald.

mitigate this by using the pretrained generator to synthesize high resolution photo-realistic images from the mapped latent codes.

4. Experimental evaluation

In this section, we validate the proposed method on a variety of architectures across different datasets. In particular, we utilize PGAN [16] on LSUN [44] and CelebA-HQ [16], StyleGAN [18] on FFHQ and BigGAN [1] on Imagenet [34]. Our evaluation demonstrates that the proposed clustering works when trained in different objects, such as faces or cars. We verify that these clusters contain semantically-relevant images by showcasing state-of-the-art attribute classification results compared to four baselines.

4.1. Experiments on faces

To highlight the effect of our method on representations of different layers, we utilize PGAN trained on CelebA-HQ. Conditional image synthesis for binary attributes is presented in Figure 2. For this experiment, the representation space was separated into 2 clusters for each layer. The results highlight that different semantic features are captured in different layers. Indicatively, the first layer captures gender, while geometric and color features are encoded in the later layers.

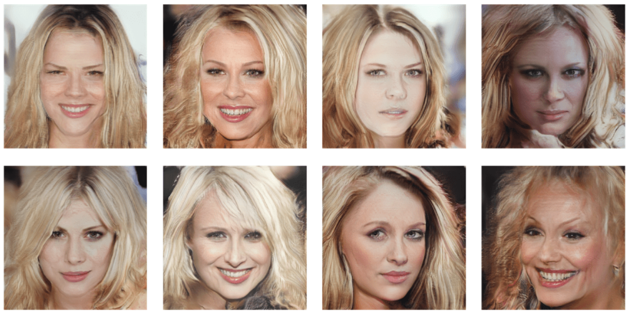

Figure 4. Fine-grained attribute synthesis for attributes female and blond hair.

The results of our method in Figure 3 showcase that the representation forms multiple clusters based on attributes like pose, hair style and age.

4.2. Experiments on objects

In addition to faces, we demonstrate in this section how our method generalizes on the object classes of LSUN. We notice that by using our method we obtain direct control of the rotation of cars, as well as other high level features of different classes. Most of the recovered clusters show considerable variation, e.g., one vs many chairs, the landscape behind a bridge and the architecture of the church. The results on Figure 5 show that our method can introduce control of significant modes of variation without loss of photo-realism.

4.3. Fine-grained attribute synthesis

As showcased in Figure 2, different semantic features are captured in different layers of PGAN. A logical extension would be to attempt to combine such features to form fine-grained attributes in a hierarchical manner. We indeed verify this assumption in Figure 4 where we sample blond female faces using PGAN on CelebA-HQ. Forming the cluster for fine-grained attributes requires two steps. First, we sample latent codes from the latent distribution, perform a forward pass and learn the clusters on the representation space of the first attribute (e.g., the first layer for gender). Then we perform a forward pass using only the samples of the specified cluster (e.g., female faces) onto the later layers (e.g., the fifth layer for hair tone). After clustering again, the resulting cluster will only correspond to blond female faces. The rest of the sampling procedure is trained using IMLE as discussed above.

4.4. Generalization across classes

We further explore the generalization of the mappings that are learned from IMLE across different classes of objects. In particular, we are interested in determining whether a mapping T_c that is responsible for a specific attribute for one class (e.g., pose of a dog) can be used on a different class (e.g., a cat) to facilitate the same attribute. To this end, we utilize BigGAN [1] trained in Imagenet. BigGAN is already a conditional model, however it only allows for object class labels (e.g., dog breed). However, when our method is used in conjunction with the model, we can control generalizable geometric attributes like pose. In Figure 6 we train the two mapping networks (one for each pose) on one object class (in the first row) and use them to sample images of different animals. The results highlight that the geometric features are encoded in the same layers for similar classes and hence, the same mapping can be used across different classes. This finding indicates that the generator learns to disentangle shape from appearance for classes with similar geometry (e.g., different cat breeds). In particular, the network learns generalisable low-level primitives for similar looking classes, e.g., pose. However, the same does not hold for higher-level class-specific attributes, such as type of car.

4.5. Comparison to end-to-end training

One of the advantages of the proposed method is that it can be used on any pretrained GAN, without the need for retraining the generator. However, we compare against SC-GAN [27] and ClusterGAN [29] that learn a clustering that separates the latent space during training. We showcase results on CelebA for 2 and 4 clusters respectively in Figure 7. In the case of 2 clusters, the method mostly learns to separate female from male faces, entangled with pose. In contrast, in the case of 4 clusters we do not notice a clear separation of semantic attributes other than image statistics (e.g., dark background) for SCGAN. We further quantify the inconsistency of attributes in each cluster in Table 1. In particular, we classify hair color and gender (using Microsoft Azure[1]) in each cluster and present the percentage corresponding to 'dark hair' and 'female'. In this setting, we binarize the hair color as the data form clusters based on light and dark hair tone. The results highlight that in some cases the distribution may even be almost uniform, indicating attribute inconsistency. Similarly, we calculate the yaw of the faces. The inconsistency in pose is demonstrated by the large standard deviation.

4.6. Quantitative comparison

To evaluate our method quantitatively, we generate samples for the attributes 'yaw' and 'gender' and evaluate them using Microsoft Azure. We compare against GANSpace [10] and SeFA [35] by identifying the attributes above in their basis of interpretable directions. Similarly, we also compare against ClusterGAN and SCGAN by identifying the clusters where each studied attribute is more prevalent. The results in Table 2 show that the images generated using the proposed method consistently contain the target attributes. On the other hand, ClusterGAN and SCGAN are not able to find clusters that separate the pose.

4.7. Implementation details

The mapping networks consist of 3 fully connected layers without biases, as well as batch-normalization between each layer [2]. The networks were optimized using Adam [19] on Pytorch [32]. We train each model for 400 epochs on a Titan X GPU with 12 GB in less than an hour. Both k-means and the nearest neighbour algorithms are implemented using

Model	Hair	Gender	Yaw (deg)
SCGAN, k=4, c=1	79%	94%	-6.79 ± 5.19
SCGAN, k=4, c=2	85%	45%	-13.4 ± 4.93
SCGAN, k=4, c=3	65%	86%	11.46 ± 7.51
SCGAN, k=4, c=4	73%	95%	-9.04 ± 5.72
SCGAN, k=2, c=1	72%	92%	-2.64 ± 5.86
SCGAN, k=2, c=2	82%	35%	-24.30 ± 7.59
ClusterGAN, k=4, c=1	69%	89%	-1.72 ± 5.88
ClusterGAN, k=4, c=2	54%	76%	-2.42 ± 6.49
ClusterGAN, k=4, c=3	67%	73%	-14.58 ± 7.81
ClusterGAN, k=4, c=4	67%	75%	7.50 ± 7.60
ClusterGAN, k=2, c=1	56%	80%	-11.59 ± 7.77
ClusterGAN, k=2, c=2	72%	69%	5.30 ± 9.5

Table 1. Attribute predictions for the images of each cluster for ClusterGAN [29] and SCGAN [27], trained for 2 and 4 clusters. For the attribute 'hair' we report the percentage of 'dark' hair (classified as 'brown' or 'black'). For the attribute of gender we report the percentage of faces classified as 'female'. For 'yaw', we report both the mean and standard deviation of the degrees.

	Gender	Yaw
ClusterGAN	51%	24%
SCGAN	95%	58%
GANSpace	98%	89%
SeFA	98%	94%
Ours	**100%**	**95%**

Table 2. Classification accuracy for multiple attributes using the baseline methods and ours. For gender, we sample images with the 'female' attribute and for yaw with the 'pose right' (at least 10 degrees from frontal). If a face is not found in the generated image, we deem it to be misclassified.

[1] https://azure.microsoft.com/en-gb/services/cognitive-services/face/

[2] https://www.math.ias.edu/~ke.li/projects/imle/

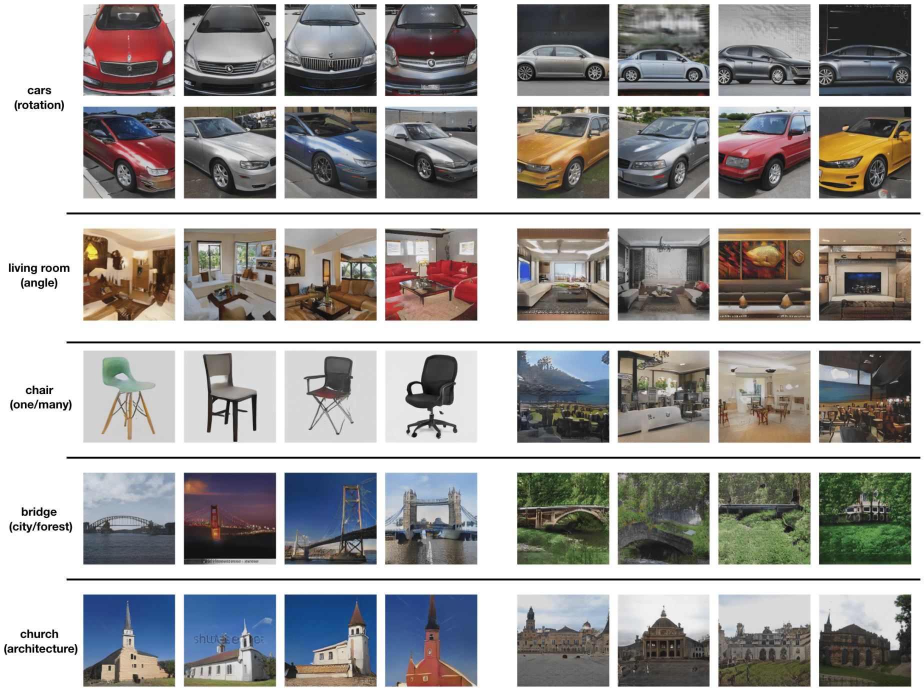

Figure 5. Synthesized images on LSUN with Progressive GAN (PGAN). The proposed approach can identify different sources of variation in each cluster, for instance rotation in the car, or multitude of objects in the chairs, background context in the bridges, and even architectural style in the churches.

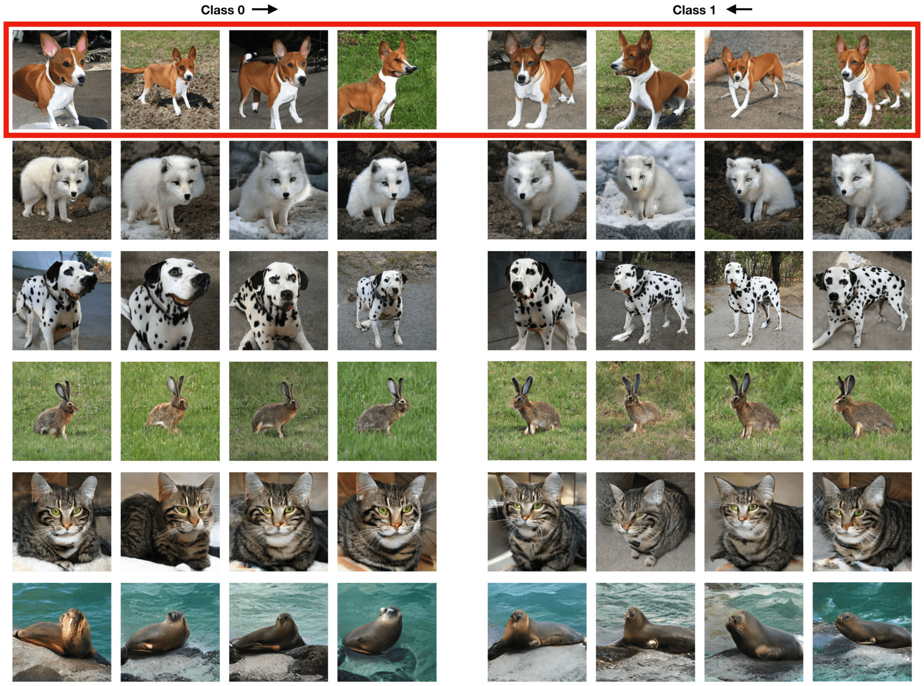

Figure 6. Generalization to different classes. The two mappings are learned in the representation space of the class depicted in the first row (i.e., red rectangle). Then, we demonstrate how the learned mappings transfer to other ImageNet classes without training them. Note that the transformation generalizes beyond other dog breeds, e.g., it applies to rabbits and seals.

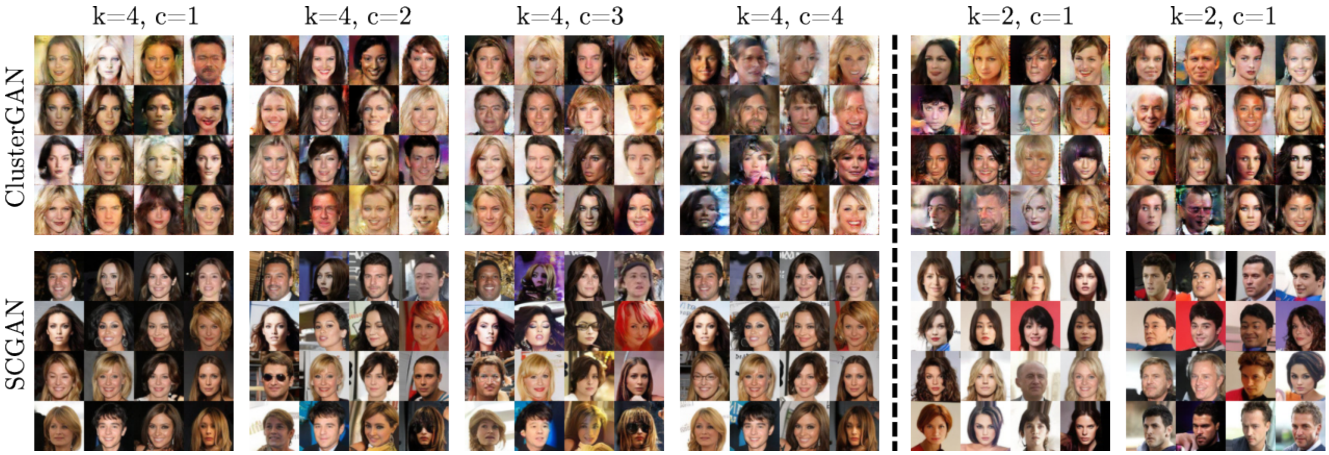

Figure 7. Results of SCGAN [27] and ClusterGAN [29] trained on CelebA for $k = 2$ and $k = 4$ total number of clusters. For SCGAN $k = 2$, we notice that the separated attribute is gender entangled with pose, while for SCGAN $k = 4$ there is no clear disentangled attributed except for the $c = 1$ cluster, which captures face with dark backgrounds. On the other hand, the clusters produced by ClusterGAN do not demonstrate consistent attributes.

FAISS [14]. To enable reproducibility of our work, we utilize open-source code for the pretrained GANs (the links can be found in the supplementary material).

5. Limitations and broader impact

Limitations: The qualitative results in the previous section highlight the efficacy of the proposed method in conditioning unconditional GANs. However, the generated attributes can be entangled and are dependent on the variation present in the training set (e.g., bald people are always male in Figure 3). It should be noted that similar limitations are faced by most unsupervised methods (e.g., [10]). Furthermore, the number of clusters used for k-means has an effect on the resulting attributes. In this work, we treat the number of clusters as a hyper-parameter but there are several heuristics in the literature that deal with this issue (e.g., elbow method or eigengap for subspace clustering [40]). However, calculating the number of clusters is beyond the exploratory purpose of this work. Lastly, since the mapping learned by IMLE is approximate, the generation quality of the samples is not always similar to the ones sampled from the latent distribution, which is a trade-off for controllable synthesis.

Broader impact: Our method is built on top of a pretrained GAN. As such, it inherits all the biases of the training data used to train the GAN, e.g., issues with CelebA-HQ. Our method can be used to control the low-level features (e.g., pose) for on-demand generation. If the dataset includes biases, those could be reflected in the clusters and invariably in the on-demand generation. On the other hand, our method can be viewed as a tool for investigating such biases, since the clusters will reflect the primary variations of the dataset. We also emphasize that the high-fidelity GANs we utilize [1, 16, 18] are publicly available. As the generation quality improves further, we believe methods like ours can be used as a 'semantic debugging tool' to uncover the biases of the GAN model. Thus, we believe our work aids towards transparency and explainability in generative models.

6. Conclusion

In this work, we introduce a method for controllable generation using unconditional GANs. The proposed method focuses on learning semantic attributes without supervision and conditioning the GAN generator using such attributes. Our method is lightweight and can work on top of any GAN generator as demonstrated by our experiments with three strong-performing generators, i.e., Progressive GAN, StyleGAN and BigGAN. We explore how those semantic attributes differ across classes and even illustrate how learned attributes in one class can transfer to different classes. A future step would be to explore automatic selection of the number of clusters.

References

[1] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations (ICLR)*, 2019.

[2] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning

of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

[4] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems (NeurIPS)*, pages 2172–2180, 2016.

[5] Grigorios Chrysos, Markos Georgopoulos, and Yannis Panagakis. Conditional generation using polynomial expansions. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[6] Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. Editing in style: Uncovering the local semantics of gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5771–5780, 2020.

[7] Markos Georgopoulos, Grigorios Chrysos, Maja Pantic, and Yannis Panagakis. Multilinear latent conditioning for generating unseen attribute combinations. In *International Conference on Machine Learning*, pages 3442–3451. PMLR, 2020.

[8] 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 (NeurIPS)*, 2014.

[9] Swaminathan Gurumurthy, Ravi Kiran Sarvadevabhatla, and R Venkatesh Babu. Deligan: Generative adversarial networks for diverse and limited data. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 166–174, 2017.

[10] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546*, 2020.

[11] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations (ICLR)*, 2(5):6, 2017.

[12] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in neural information processing systems (NeurIPS)*, 2016.

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

[14] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017.

[15] Takuhiro Kaneko, Kaoru Hiramatsu, and Kunio Kashino. Generative adversarial image synthesis with decision tree latent controller. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6606–6615, 2018.

[16] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, 2018.

[17] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. 2020.

[18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[20] Maxim Kuznetsov, Daniil Polykovskiy, Dmitry P Vetrov, and Alex Zhebrak. A prior of a googol gaussians: a tensor ring induced prior for generative models. In *Advances in neural information processing systems (NeurIPS)*, pages 4104–4114, 2019.

[21] Wonkwang Lee, Donggyun Kim, Seunghoon Hong, and Honglak Lee. High-fidelity synthesis with disentangled representation. *arXiv preprint arXiv:2001.04296*, 2020.

[22] Damian Leśniak, Igor Sieradzki, and Igor Podolak. Distribution-interpolation trade off in generative models. In *International Conference on Learning Representations (ICLR)*, 2019.

[23] Ke Li and Jitendra Malik. Implicit maximum likelihood estimation. *arXiv preprint arXiv:1809.09087*, 2018.

[24] Zejian Li, Yongchuan Tang, and Yongxing He. Unsupervised disentangled representation learning with analogical relations. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2018.

[25] Zinan Lin, Kiran Thekumparampil, Giulia Fanti, and Sewoong Oh. Infogan-cr and modelcentrality: Self-supervised model training and selection for disentangling gans. In *International Conference on Machine Learning (ICML)*, pages 6127–6139, 2020.

[26] Bingchen Liu, Yizhe Zhu, Zuohui Fu, Gerard de Melo, and Ahmed Elgammal. Oogan: Disentangling gan with one-hot sampling and orthogonal regularization. In *AAAI Conference on Artificial Intelligence*, 2020.

[27] Steven Liu, Tongzhou Wang, David Bau, Jun-Yan Zhu, and Antonio Torralba. Diverse image generation via self-conditioned gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14286–14295, 2020.

[28] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. In *International Conference on Learning Representations (ICLR)*, 2018.

[29] Sudipto Mukherjee, Himanshu Asnani, Eugene Lin, and Sreeram Kannan. Clustergan: Latent space clustering in generative adversarial networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4610–4617, 2019.

[30] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

[31] James Oldfield, Markos Georgopoulos, Yannis Panagakis, Mihalis A Nicolaou, and Ioannis Patras. Tensor component analysis for interpreting the latent space of gans. *arXiv preprint arXiv:2111.11736*, 2021.

[32] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NeurIPS-W*, 2017.

[33] Antoine Plumerault, Hervé Le Borgne, and Céline Hudelot. Controlling generative models with continuous factors of variations. In *International Conference on Learning Representations (ICLR)*, 2020.

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[35] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. *arXiv preprint arXiv:2007.06600*, 2020.

[36] Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6490–6499, 2019.

[37] Rajhans Singh, Pavan Turaga, Suren Jayasuriya, Ravi Garg, and Martin Braun. Non-parametric priors for generative adversarial networks. In *International Conference on Machine Learning (ICML)*, pages 5838–5847, 2019.

[38] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 3, page 7, 2017.

[39] Michal Uřičář, Pavel Křížek, David Hurych, Ibrahim Sobh, Senthil Yogamani, and Patrick Denny. Yes, we gan: Applying adversarial techniques for autonomous driving. *Electronic Imaging*, 2019(15):48–1, 2019.

[40] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

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

[42] Xueting Yan, Ishan Misra, Abhinav Gupta, Deepti Ghadiyaram, and Dhruv Mahajan. Clusterfit: Improving generalization of visual representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6509–6518, 2020.

[43] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5147–5156, 2016.

[44] 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.

[45] Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Joshua B Tenenbaum, and William T Freeman. Visual object networks: Image generation with disentangled 3d representation. In *Advances in neural information processing systems (NeurIPS)*, 2018.

[46] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6002–6012, 2019.