OpenGAN: Open Set Generative Adversarial Networks

Luke Ditria*, Benjamin J. Meyer*, and Tom Drummond

ARC Centre of Excellence for Robotic Vision, Monash University
{luke.ditria,benjamin.meyer,tom.drummond}@monash.edu

Abstract. Many existing conditional Generative Adversarial Networks (cGANs) are limited to conditioning on pre-defined and fixed class-level semantic labels or attributes. We propose an open set GAN architecture (OpenGAN) that is conditioned per-input sample with a feature embedding drawn from a metric space. Using a state-of-the-art metric learning model that encodes both class-level and fine-grained semantic information, we are able to generate samples that are semantically similar to a given source image. The semantic information extracted by the metric learning model transfers to out-of-distribution novel classes, allowing the generative model to produce samples that are outside of the training distribution. We show that our proposed method is able to generate 256×256 resolution images from novel classes that are of similar visual quality to those from the training classes. In lieu of a source image, we demonstrate that random sampling of the metric space also results in high-quality samples. We show that interpolation in the feature space and latent space results in semantically and visually plausible transformations in the image space. Finally, the usefulness of the generated samples to the downstream task of data augmentation is demonstrated. We show that classifier performance can be significantly improved by augmenting the training data with OpenGAN samples on classes that are outside of the GAN training distribution.

1 Introduction

Generating new data that matches a target distribution is a challenging problem with applications including image-to-image translation [25,68], data augmentation [11,63] and video prediction [33,30]. A popular approach to this problem is Generative Adversarial Networks (GANs) [15], which train a generator and discriminator network in an adversarial manner. However, such networks have issues with training instability, especially for complicated and multi-modal data, and often result in a lack of diversity in the generated samples, particularly when training data is limited [18]. Conditional GANs (cGANs) [41] achieve greater control over the generated samples by conditioning the model on information including class labels [49,43,65,2], attributes [12,27,36], textual attributes [52,53,38,66,7,50] or object pose [61,13,64]. However, class conditional GANs are

* Authors contributed equally.

Fig. 1: Given novel class source images (top row of each section), our approach is able to generate 256×256 samples (bottom two rows of each section) that closely match the features of the source. OpenGAN was not trained on the classes shown.

unable to generate novel class samples, attribute conditional GANs are limited to a fixed set of pre-defined attributes and pose conditional GANs require hand-labelled and pre-defined pose codes or object landmarks. While some existing methods condition on image-level features using an encoder-decoder architecture [1,61,67], these approaches train the encoder concurrently with the generator, enforcing no restrictions on the information encoded in the features. This can undesirably result in significant variation of the discriminative semantic information in samples generated from the same source image (see Section 5.6).

In this work, we propose an open set GAN (OpenGAN) that conditions the model on per-image features drawn from a metric space. Deep metric learning approaches have been shown to learn metric spaces that encode both class-level and fine-grained semantic information, and also have the ability to transfer to novel, out-of-distribution classes [59,57,56,20,58,54,40,59]. By conditioning on per-image metric features, our proposed model is not limited to closed-set problems, but can also generate samples from novel classes in the open-set domain (see Figure 1). Further, this conditioning method results in high intra-class diversity, where that is desirable. Unlike many existing methods, our approach is not conditioned on class-level information alone or pre-defined attributes and poses, but rather on the semantic information extracted by a state-of-the-art deep metric learning model. Additionally, the proposed approach differs from existing feature matching GANs [55,16,17], as it does not attempt to match feature moments over the entire dataset, but conditions the model on a per-feature basis. During testing, data can be generated by conditioning on specific source images, or by randomly sampling the metric feature space.

Given a metric feature extracted from a real source image, our model generates images that visually and semantically match the source, as shown in Figure 1. The generator should not simply reconstruct the source image, but produce images with features that are similar to the source, when passed through the metric learning model. Conditioning the generator on semantically rich features not only allows for the generation of both in-distribution and novel class images, but also for transfer between source domains (Section 5.8). Further, OpenGAN samples can be successfully utilised for data augmentation (Section 5.9).

The use of a metric learning model is an important design decision. Metric features describe only discriminative semantic information, ignoring all contex-

tual and structural information, such as pose, the quantity and arrangement of objects and other non-discriminative intra-class and inter-class variations. As a result, the generator relies on a latent space noise vector to map this information (and only this information), meaning that the structural and contextual information can be modified without any variation occurring between the semantic content of the source image and generated image. This is unlike in encoder-decoder GAN architectures that learn to extract image features concurrently with the GAN [1,61,67]. For our approach, the content information is cleanly split into two distinct spaces, without the need for the pre-defined, hand-labelled pose and landmark information that is required in previous work [61,13,64].

2 Related Work

Conditional Generative Models. The two most commonly used generative models in recent times are Generative Adversarial Networks [15] and Variational Auto-Encoders [29]. In this work, we focus on deep convolutional GANs [51]. Several methods have been proposed to achieve greater control over the generated images. Mirza and Osindero [41] condition on class-level labels by supplying one-hot class vectors to both the generator and discriminator. Such an approach can improve both generated image quality and inter-class diversity in the generator distribution. Incorporating class-level information by treating the discriminator as a multi-label classifier has also been shown to improve the quality of generated samples [48,55]. Odena *et al.* [49] extend this by tasking the discriminator with estimating both the probability distribution over class labels and over the source distribution (i.e. real or fake). Conditional information can also be incorporated by way of conditional normalisation layers [9,6,43,65,2], which learn the batch normalisation [24] or instance normalisation [62] scale and bias terms as a function of some input.

Beyond class-level conditional information, data generation can also guided by conditioning the model on pre-defined attributes [12,27,36], such as hair colour and style for face generation. Similarly, generative models can be conditioned on attributes in a textual form by text-to-image synthesis methods [52,53,38,66,7,50]. GANs can be conditioned on structural information, allowing direct control over the object pose in the generated image. Such methods require hand-labelled and pre-defined pose codes or object landmarks [61,13,64].

Methods including DAGAN [1], MetaGAN [67] and DR-GAN [61] use an encoder-decoder structure, allowing the generator to be conditioned on image-level features. As such, these approaches are not limited to in-distribution classes by their design, unlike class conditional GANs. However, the encoder and generator are trained simultaneously with no constraints on the information that is represented in the encoder features. Unlike these methods, our approach leverages metric features extracted from a deep metric learning model that is trained prior to the GAN. Metric learning models have a demonstrated efficacy for open set problems [39], as such, our approach is explicitly designed for the open set domain. Further unlike the encoder-decoder GANs, our method results in no

semantic variation when changing only the latent vector. This is because all discriminative semantic information is encoded in the metric features and the generator is constrained to produce images with features that match those of the source image. Consequently, the latent vector can only encode non-discriminative information, such as the object pose, the background and the number of objects in the image. Encoder-decoder GANs do not enforce these feature constraints.

Nguyen *et al.* [46,45] condition the generator using an auxiliary classifier network by finding the latent vector that results in generated data that strongly activates neurons in the auxiliary network. These so called Plug and Play Generative Networks can generate data that is outside of the generator's training distribution, but is inside the auxiliary network's training distribution. For our approach, the generator and feature extractor training distributions are the same, and the generator can be conditioned on data that is outside of that distribution.

Matching Networks. Training stability of GANs can be improved by performing feature matching [55]. The generator is trained such that the expected value of features extracted from generated data by a given layer of the discriminator matches that of the real data. Similar to feature matching networks are moment matching networks [32,10,31], which generally try to match all moments of the distributions using maximum mean discrepancy [16,17]. Unlike feature matching networks, our approach attempts to match per-sample source features individually, rather than the expected value. Our generator is also directly conditioned on per-sample features, such that the generated samples match the semantic content of the source features. Further, we use an auxiliary metric learning model to extract features, rather than the discriminator. Our feature matching is also related to perceptual loss functions [26], which use a pre-trained classifier network to match the low and high level features of input and target images for problems such as style transfer.

Metric Learning. Rich visual features can be extracted from images by using a deep convolutional neural network to learn a distance metric over the images [59,57,56,20,58,54,40,59]. Many of these so-called deep metric learning methods are based on Siamese [3,4,19] and triplet networks [23], which perform distance comparisons in the feature space. Research in this area often focuses on the generalisation of triplet loss, such that multiple pairwise distance comparisons can be made for a given example within a training batch [59,57]. Other work focuses on the selection of informative triplets via mining techniques [56,20]. Beyond triplet loss, the work by Song *et al.* [58] directly minimises a clustering measure. Rippel *et al.* propose Magnet loss [54], which explicitly models class distributions in the feature space and penalises class overlap. Other approaches minimise Neighbourhood Component Analysis (NCA) loss over the set of training features [40] or per-class proxy features [44]. Metric learning has been combined with generative models to improve the stability of GAN training [8,5], as well as to improve the training of a metric learning model [69]. Unlike these methods, we use a metric learning model to condition a GAN on image features.

Fig. 2: Visualisation of the metric feature space for 20 novel classes, represented by colour. The feature extractor is not trained on any of the shown flower species, yet examples are co-located based on class. Example images are selected to show similar classes being located nearby. Best viewed zoomed-in.

3 Background

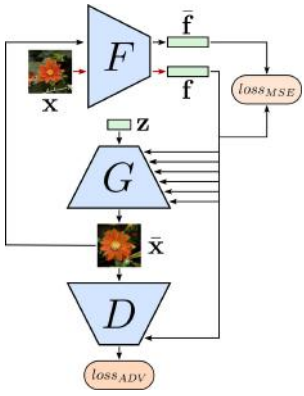
3.1 Generative Adversarial Networks

Let G be a *generator* network that attempts to learn a mapping from a latent space to a target data space. Specifically, an image is generated as $\bar{\mathbf{x}} = G(\mathbf{z})$, where $\mathbf{z}$ is a latent vector sampled from the distribution $p_z = \mathcal{N}(0, 1)$. Further, let D be a *discriminator* network that takes as input an image and attempts to distinguish between the generator distribution and the real data distribution p_d. The two networks are trained in an adversarial fashion, with improvement in one network driving improvement in the other. Greater control over the generated image can be achieved by conditioning both networks on a label $y \in p_d$.

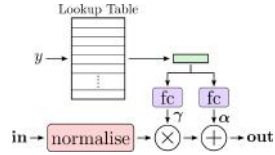
3.2 Deep Metric Learning

Our proposed method is agnostic in terms of the metric learning model used to extract image features. Here, we detail the metric learning algorithm that is used for all experiments in Section 5. This particular approach [40] is selected due to its ability to extract semantically rich features that both transfer well to novel classes and encode fine-grained intra-class and inter-class variations. This is shown in the t-SNE visualisation [37] of novel class examples in Figure 2. Despite being from outside of the training set distribution, examples are well clustered based on class, with semantically similar classes located nearby.

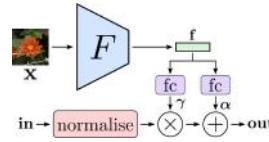
For a given input image, the network F extracts a d-dimensional feature $\mathbf{f} = [f^{(i)}, ..., f^{(d)}]$. For training, a set of n Gaussian kernel centres are defined in the feature space as $\mathcal{C} = \{\mathbf{c}_1, ..., \mathbf{c}_n\}$, where $\mathbf{c}_i = [c_i^{(1)}, ..., c_i^{(d)}]$ is the i-th kernel centre. The centres are defined to be the locations of the n training set features, with the weights of F updated during training by minimising the NCA loss [14]. To make training feasible, a cached version of the kernel centres $\hat{\mathcal{C}}$ is stored and updated periodically during training, avoiding the need to do so at every



(a) Overview of method.



(b) Class conditional normalisation.



(c) Feature embedding conditional normalisation.

Fig. 3: (a) Overview of our approach. During training, features are extracted from training images and used to condition the GAN via conditional normalisation layers. During testing, features may be extracted from source images or randomly sampled from the metric space. (b) Class conditional normalisation compared to the feature conditional normalisation (c) used in our model.

training iteration. The loss minimised during training is shown in Equation 1, where σ is a hyperparameter and ℓ_i is the class label of the i-th training example. If necessary, approximate nearest neighbour search can leveraged to make the approach scalable both in terms of the number of classes and training examples.

$$loss_F = -\sum_{\mathbf{c}_i \in \mathcal{C}} \ln \left(\frac{\sum_{\hat{\mathbf{c}}_j \in \hat{\mathcal{C}}, i \neq j, \ell_i = \ell_j} \exp\left(\frac{-\|\mathbf{c}_i - \hat{\mathbf{c}}_j\|^2}{2\sigma^2} \right)}{\sum_{\hat{\mathbf{c}}_k \in \hat{\mathcal{C}}, i \neq k} \exp\left(\frac{-\|\mathbf{c}_i - \hat{\mathbf{c}}_k\|^2}{2\sigma^2} \right)} \right) \tag{1}$$

4 Feature Embedding Conditional GANs

4.1 Overview

Our proposed method consists of three convolutional neural networks: a generator G, a discriminator D and a metric feature extractor F. Network F extracts a feature $\mathbf{f}$ for a sampled training image, which is fed into networks G and D. The generator attempts to produce an image that both fools the discriminator and results in a feature $\bar{\mathbf{f}}$ that closely matches the real feature, when the fake image is passed through network F. The former is achieved via an adversarial (ADV) loss, while the latter is achieved by a mean squared error (MSE) loss term in the metric feature space. This is illustrated in Figure 3a. During testing, examples can be generated by conditioning on features from specific images or by simply sampling the feature space.

Algorithm 1 Training algorithm for OpenGAN.

Require:
Models F, G, D with parameters $\boldsymbol{\theta}_F$, $\boldsymbol{\theta}_G$, $\boldsymbol{\theta}_D$
Scale term for feature loss λ
Pre-train $\boldsymbol{\theta}_F$ (Section 3.2)
while $\boldsymbol{\theta}_G$ is not converged **do**
  Sample $\mathbf{x} \sim p_d$
  $\mathbf{f} \leftarrow F(\mathbf{x})$
  Sample $\mathbf{z} \sim \mathcal{N}(0, 1)$
  $\mathcal{L}_D \leftarrow \min(0, 1 - D(\mathbf{x}, \mathbf{f}))$
  $+ \min(0, 1 + D(G(\mathbf{z}, \mathbf{f}), \mathbf{f}))$
  $\boldsymbol{\theta}_D \leftarrow \boldsymbol{\theta}_D - \text{Adam}(\nabla \mathcal{L}_D)$
  Sample $\mathbf{z} \sim \mathcal{N}(0, 1)$
  $\bar{\mathbf{x}} \leftarrow G(\mathbf{z}, \mathbf{f})$
  $\mathcal{L}_G \leftarrow -D(\bar{\mathbf{x}}, \mathbf{f}) + \lambda \, \|F(\bar{\mathbf{x}}) - \mathbf{f}\|^2$
  $\boldsymbol{\theta}_G \leftarrow \boldsymbol{\theta}_G - \text{Adam}(\nabla \mathcal{L}_G)$
end while

Features are incorporated into the generator and discriminator by way of feature conditional normalisation layers, described in detail in Section 4.3. The normalisation scale and bias terms are learned as a continuous function of the conditioning features, as opposed to a discreet function of class labels in class conditional normalisation. This continuity means that during testing, meaningful interpolation between features can occur. Further, out-of-distribution images can be generated by sampling a desired point in the metric feature space or by conditioning on the feature extracted from a specific novel image.

In a conventional GAN or class conditional GAN framework, generating an image that visually and semantically matches a given source image can be challenging. Additionally, there is no mechanism in the generator training that encourages the ability to transfer to data outside of the training distribution. Conversely, the training of our generator is guided by a feature extractor that transfers to novel classes (see Section 3.2) and the ability to condition the generator on a specific source image is built-in to the framework.

4.2 Training Procedure

Network optimisation is outlined in Algorithm 1. The feature extractor is pre-trained, with the weights subsequently frozen. The loss functions minimised by the discriminator and the generator, respectively, are:

$$loss_D = \mathbb{E}_{\mathbf{x} \sim p_d} \left[\min(0, 1 - D(\mathbf{x}, \mathbf{f})) \right] + \mathbb{E}_{\mathbf{x} \sim p_d, \mathbf{z} \sim p_z} \left[\min(0, 1 + D(G(\mathbf{z}, \mathbf{f}), \mathbf{f})) \right], \tag{2}$$

$$loss_G = \mathbb{E}_{\mathbf{x} \sim p_d, \mathbf{z} \sim p_z} \left[-D(G(\mathbf{z}, \mathbf{f}), \mathbf{f}) + \lambda \, \|F(G(\mathbf{z}, \mathbf{f})) - \mathbf{f}\|^2 \right], \tag{3}$$

where λ is a scaling term for the feature loss component and $\mathbf{f} = F(\mathbf{x})$. Hinge loss [34,60] is used for the adversarial component of the losses, while mean squared error is used for the feature loss component. Model parameters are updated by gradient descent with Adam optimisation [28].

Table 1: Comparison of FID and intra-class FID scores. Lower scores indicate better sample quality.

	FID	Intra FID
U-SAGAN [65]	161.74	-
C-SAGAN [65]	66.12	179.67
Ours: T-SM	22.05	103.18
Ours: N-SM	39.51	110.04
Ours: N-RF	31.89	104.90

(a) U-SAGAN. (b) C-SAGAN. (c) Ours: T-SM. (d) Ours: N-SM. (e) Ours: N-RF.

Fig. 4: Uncurated and randomly selected images on the Flowers102 dataset.

4.3 Feature Conditional Normalisation

Intermediate neural network layer activations can be forced to have similar distributions by including layers that normalise over the entire batch [24] or over each instance individually [62]. Normalisation of activations can lead to faster and more stable training, as well as better overall model performance. Such layers perform the following normalisation on an activation:

$$\hat{m}_i = \gamma \frac{m_i - \mu}{\sqrt{v + \epsilon}} + \alpha \tag{4}$$

where m_i is the input, $\hat{m}_i$ is the normalised output, μ is the mean, v is the variance and ϵ is a small constant. In conventional normalisation layers, the scale γ and bias α terms are learned model parameters, while for conditional normalisation layers, they are learned as a function of some input. Class conditional normalisation (Figure 3b) learns a feature per-class that is often input to two fully connected (FC) layers to produce the scale and bias terms. This limits the network to produce only images from the training distribution or to an interpolation between training classes.

We propose metric feature embedding conditional normalisation (Figure 3c), which learns the scale and bias terms as a function of a feature embedding drawn from a metric space. This allows conditioning on specific images or features, compared to in-distribution class-level conditioning.

(a) Ours: T-SM. (b) Ours: N-SM. (c) Ours: N-RF.

Fig. 5: Uncurated and randomly selected images on the CelebA dataset.

Fig. 6: Novel class real source images (top row) and resultant generated images (bottom two rows). Although the identities are not present during training, the fake images match the features of the real source images.

5 Experiments

5.1 Implementation Details

The datasets used for evaluation are Oxford Flowers102 [47] and CelebA Faces [35]. Each dataset is split into training and novel classes. The first 82 classes from Flowers102 are used for training, resulting in 6433 training images. For CelebA, identities are used as class labels with the 3300 identities containing the most samples used for training, resulting in 97262 training images. The attribute and pose labels of CelebA are used for attribute and pose interpolation, however, the networks are not trained on this information.

The generator and discriminator follow a similar architecture to Self-Attention GAN (SAGAN) [65], but we replace the projection layer in the discriminator with a single feature embedding conditional normalisation layer and a fully connected layer. We also generate images twice the resolution of SAGAN at 256×256 pixels and use a channel width multiplier of 32. Six residual blocks [21] are used in each network, along with spectral normalisation [42] and a single self-attention block [65]. Feature conditional normalisation is used in all residual blocks in the generator but only in the final discriminator block. We find batch normalisation on Flowers102 and instance normalisation on CelebA performs best in practice. A latent space dimension of 128 is used for the generator. For training, a base learning rate of 10^{-4}, batch size of 48 and Adam optimiser [28] with $\beta_1 = 0$ and $\beta_2 = 0.999$ are used. The value of λ is set to 0.01. The GAN is trained for up to 60000 iterations on four Nvidia 1080 Ti GPUs, taking approximately 15 hours.

(a) (b)

Fig. 7: (a) Fake samples (right) are generated from a fixed feature, extracted from the novel class real samples (left). (b) Examples from three novel classes with a fixed latent vector for each class. The features of the real images (top rows of each section) are used to condition the GAN to produce the fake images (bottom rows of each section).

A ResNet18 architecture [21] with the class-dependent fully connected layer removed is used for the feature extractor, producing 512-dimensional features. The model is trained with a base learning rate of 10^{-5}, Gaussian σ of 10 and an Adam optimiser [28] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The stored Gaussian centres are updated every 5 epochs.

5.2 Comparison to Baselines

As the proposed method can use any suitable network architecture, the aim of this work is not to improve on state-of-the-art methods in terms of sample quality. Here, we aim to show that our method results in samples of at least comparable quality to appropriate baselines. We compare to two baselines: Unconditional SAGAN (*U-SAGAN*) and Class Conditional SAGAN (*C-SAGAN*) [65]. For fair comparison, these baselines have the same structure as our model, differing only in terms of the normalisation layers. U-SAGAN uses non-conditional normalisation, while C-SAGAN uses a single conditioning feature per-class (Figure 3b).

Uncurated qualitative results on the Flowers102 dataset can be seen in Figure 4 and a quantitative comparison, in terms of the FID and intra-class FID scores [22], is shown in Table 1. For our approach, we investigate sampling features from both the training and novel distributions, as well as two methods of feature sampling: random sampling from normal distributions the centred on the class means, and extracting features from sampled real images. Our methods are:

- *Ours: T-SM*: Training distribution, sample means.
- *Ours: N-SM*: Novel distribution, sample means.
- *Ours: N-RF*: Novel distribution, real image features.

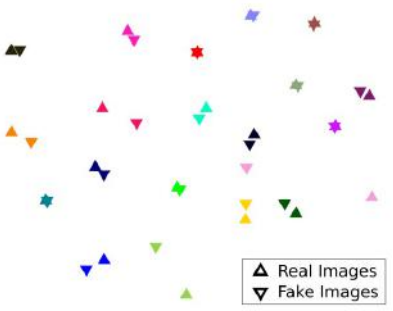


Fig. 8: Novel per-class mean feature embeddings for real and fake images. Colour represents class.

Fig. 9: Interpolation between two latent vectors (horizontal) and two feature embeddings (vertical).

Both qualitatively and quantitatively, there is little difference in quality between sampling training and novel distributions, or between the two feature sampling methods. This is also observed on CelebA in Figure 5. Compared to the baselines, our approach results in both higher quality images and better sample diversity.

5.3 One-Shot Image Generation

In this section, we show that our method is able to generate samples that match the semantic features of source images sampled from the novel distribution. We name this problem "one-shot image generation", however, it is important to note that no updates are made to the network weights using the novel source images; the source images are simply used to condition the generator. Figure 1 demonstrates this ability on both datasets, while further CelebA samples are shown in Figure 6. Additional Flowers102 samples are shown in Figures 7a and 7b, with discussion in Section 5.4.

Figure 8 shows a t-SNE visualisation [37] of the novel per-class mean features of the real and fake samples when passed through network F. In the majority of cases, the fake mean feature is co-located with the real mean feature.

5.4 Single Source and Intra-Class Diversity

Our method is able to generate a range of samples from a single source image by randomly sampling the latent vector. This single source diversity is demonstrated in Figure 7a. The generated samples match the semantic features of the source image, but varying the latent vector results in structural changes, such as the pose and number of flowers present. Intra-class diversity is demonstrated in Figure 7b by fixing the latent vector and sampling various features from the same class. Due to the fixed latent vector, the structural information is consistent, while the sampling of different features results in fine-grained intra-class differences, such as colour. Again, all source images are from novel classes.

(a) Pose interpolation in latent space.

(c) Attribute (age, bangs, gender) interpolation in feature space.

(b) Pose interpolation in feature space.

(d) Attribute (age) interpolation in latent space.

Fig. 10: Pose and attribute interpolation.

Source. DAGAN [1]. Ours: OpenGAN.

Fig. 11: Fixed source image per row with random latent vectors. DAGAN samples show significant semantic variation, while OpenGAN samples do not.

5.5 Latent and Feature Space Interpolation

A two-dimensional interpolation between two latent vectors (horizontal direction) and two feature embeddings (vertical direction) is shown in Figure 9. The generated samples are required to contain the semantic information encoded in the given feature embedding. As such, interpolation in latent space with a fixed feature results in plausible transformations in the image space, without changes in the fine-grained semantic content. This is unlike latent space interpolation in conventional cGANs, which by design results in intra-class semantic variations.

By training a classifier to predict the binary pose and attribute labels of CelebA, we are able to compute pose/attribute mean latent and feature vectors. If a given attribute is encoded, traversing the line that connects the mean positive and negative vectors will vary that attribute in the image space. As seen in Figure 10, pose information is encoded only in the latent space, with no pose change seen when interpolating between the mean feature vectors. Conversely, attributes such as age, gender and hair style are encoded only in feature space.

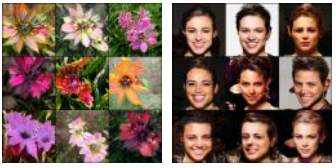


Fig. 12: Random sampling of the feature space.

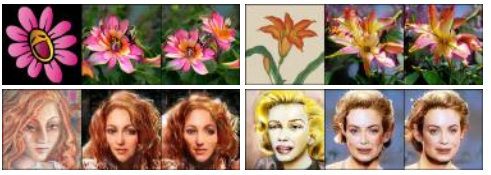


Fig. 13: Generating from out-of-domain source images.

5.6 Split of Information in Latent and Feature Spaces

As seen in Figures 7 and 10, all discriminative semantic information is encoded in the feature space and only non-discriminative structural information is encoded in the latent space. In Figure 11, we show that this clean split does not exist in encoder-decoder style image-conditional GANs, such as DAGAN [1]. We train a DAGAN model using the official implementation on Flowers102. It can be seen that for a fixed source image, DAGAN samples undesirably show significant semantic variation (e.g. the colour of the flower) when varying only the latent space, while OpenGAN samples show no discriminative semantic variation.

5.7 Random Feature Space Sampling

Conventional GANs are able to generate data by randomly sampling the latent space without any external inputs. Our generator is trained not only with latent space sampling, but also feature space sampling. Figure 12 shows that new data can be generated by randomly selecting both the latent and feature vectors. The generated samples are diverse, as well as visually and semantically plausible.

5.8 Out-of-Domain Source Images

We investigate the use of out-of-domain source images, such as paintings and digital art, that have similar semantic content as the training images. As seen in Figure 13, the fake samples match the semantic features of the source images. This shows that the metric learning model is able to extract relevant information, despite the domain shift.

5.9 OpenGAN for Data Augmentation

In this section, we demonstrate the usefulness of samples generated by OpenGAN to the downstream application of data augmentation for classification. As a baseline, we train a Resnet18 [21] classifier on 500 novel (i.e. outside of the OpenGAN training distribution) CelebA classes, using 1, 2, 5 and 10 training examples per class. The same test set is used for all experiments. To train the classifier with data augmentation, we first sample a batch of real images from

Table 2: CelebA data augmentation using OpenGAN samples.

	Real Per Class	Fake Per Real	η	Test Acc. (%)
Baseline	1	0	-	2.71
With Data Aug.	1	5	2	**12.13**
Baseline	2	0	-	7.47
With Data Aug.	2	4	1.5	**22.70**
Baseline	5	0	-	25.98
With Data Aug.	5	3	1.5	**51.81**
Baseline	10	0	-	52.69
With Data Aug.	10	2	1.5	**71.98**

the training data set and perform an optimisation step on the classifier. Using the metric features extracted from the sampled real images, a batch of fake images is generated, which is used to perform another optimisation step on the classifier. The randomised generation of fake images and classifier optimisation step is repeated using the same batch of real features until the desired ratio of fake-to-real data is achieved. A new batch of real images is then sampled and the process repeats. We find that adding small random perturbations to the real features before generating fake data can be beneficial. The perturbations are Gaussian noise with a zero mean and standard deviation of $\eta\,\sigma_F$, where η is a scaling term and σ_F is the standard deviation of the real features across all examples and dimensions.

For each number of real samples per class, we experiment with fake-to-real data ratios of 1 through 5 and η values of 0, 1.5 and 2. The best performing experiments for each number of real samples per class are shown in Table 2. Data augmentation results in a significant improvement in classification performance, despite the classes being from outside of the OpenGAN training distribution.

6 Conclusion

In this paper, we proposed a generative adversarial network that is conditioned on per-sample feature embeddings drawn from a metric space. Such an approach allows the generation of samples that are semantically similar to a given source image. Our method is able to generate data from novel classes that are outside of the training distribution. We demonstrated that interpolation in the feature and latent spaces results in semantically plausible samples, with the feature space encoding fine-grained semantic information and the latent space encoding structural information. Finally, generated samples can be used to significantly improve classification performance through data augmentation.

7 Acknowledgements

This research was supported by the Australian Research Council Centre of Excellence for Robotic Vision (project number CE140100016).

References

1. Antoniou, A., Storkey, A., Edwards, H.: Data augmentation generative adversarial networks. In: International Conference on Learning Representations Workshops (ICLRw) (2017)
2. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. In: International Conference on Learning Representations (ICLR) (2019)
3. Bromley, J., Guyon, I., Lecun, Y., Sackinger, E., Shah, R.: Signature verification using a Siamese time delay neural network. In: Advances in Neural Information Processing Systems (NeurIPS) (1993)
4. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 1, pp. 539–546 (2005)
5. Dai, G., Xie, J., Fang, Y.: Metric-based generative adversarial network. In: Proceedings of the 25th ACM international conference on Multimedia. pp. 672–680. ACM (2017)
6. De Vries, H., Strub, F., Mary, J., Larochelle, H., Pietquin, O., Courville, A.C.: Modulating early visual processing by language. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 6594–6604 (2017)
7. Dong, H., Yu, S., Wu, C., Guo, Y.: Semantic image synthesis via adversarial learning. In: IEEE International Conference on Computer Vision (ICCV). pp. 5706–5714 (2017)
8. Dou, Z.Y.: Metric learning-based generative adversarial network. arXiv preprint arXiv:1711.02792 (2017)
9. Dumoulin, V., Shlens, J., Kudlur, M.: A learned representation for artistic style. In: International Conference on Learning Representations (ICLR) (2017)
10. Dziugaite, G.K., Roy, D.M., Ghahramani, Z.: Training generative neural networks via maximum mean discrepancy optimization. In: Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence. pp. 258–267 (2015)
11. Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., Greenspan, H.: Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. Neurocomputing **321**, 321–331 (2018)
12. Gauthier, J.: Conditional generative adversarial nets for convolutional face generation. Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester **2014**(5), 2 (2014)
13. Ge, Y., Li, Z., Zhao, H., Yin, G., Yi, S., Wang, X., et al.: Fd-gan: Pose-guided feature distilling gan for robust person re-identification. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 1222–1233 (2018)
14. Goldberger, J., Hinton, G.E., Roweis, S.T., Salakhutdinov, R.R.: Neighbourhood components analysis. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 513–520 (2005)
15. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 2672–2680 (2014)

16. Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., Smola, A.J.: A kernel method for the two-sample-problem. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 513–520 (2007)
17. Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. J. Mach. Learn. Res. **13**, 723–773 (Mar 2012)
18. Gurumurthy, S., Kiran Sarvadevabhatla, R., Venkatesh Babu, R.: Deligan: Generative adversarial networks for diverse and limited data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 166–174 (2017)
19. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality Reduction by Learning an Invariant Mapping. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 2, pp. 1735–1742 (2006)
20. Harwood, B., Kumar, B., Carneiro, G., Reid, I., Drummond, T., et al.: Smart mining for deep metric learning. In: IEEE International Conference on Computer Vision (ICCV). pp. 2821–2829 (2017)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
22. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 6629–6640 (2017)
23. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: International Workshop on Similarity-Based Pattern Recognition. pp. 84–92 (2015)
24. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning (ICML). pp. 448–456 (2015)
25. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1125–1134 (2017)
26. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision (ECCV) (2016)
27. Kaneko, T., Hiramatsu, K., Kashino, K.: Generative attribute controller with conditional filtered generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7006–7015 (2017)
28. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)
29. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
30. Kwon, Y.H., Park, M.G.: Predicting future frames using retrospective cycle gan. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1811–1820 (2019)
31. Li, C.L., Chang, W.C., Cheng, Y., Yang, Y., Póczos, B.: Mmd gan: Towards deeper understanding of moment matching network. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 2203–2213 (2017)
32. Li, Y., Swersky, K., Zemel, R.: Generative moment matching networks. In: International Conference on Machine Learning (ICML). pp. 1718–1727 (2015)
33. Liang, X., Lee, L., Dai, W., Xing, E.P.: Dual motion gan for future-flow embedded video prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1744–1752 (2017)

34. Lim, J.H., Ye, J.C.: Geometric gan. arXiv preprint arXiv:1705.02894 (2017)
35. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: IEEE International Conference on Computer Vision (ICCV) (2015)
36. Lu, Y., Tai, Y.W., Tang, C.K.: Attribute-guided face generation using conditional cyclegan. In: European Conference on Computer Vision (ECCV). pp. 282–297 (2018)
37. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. Journal of Machine Learning Research **9**(Nov), 2579–2605 (2008)
38. Mansimov, E., Parisotto, E., Ba, J.L., Salakhutdinov, R.: Generating images from captions with attention. In: International Conference on Learning Representations (ICLR) (2016)
39. Meyer, B.J., Drummond, T.: The importance of metric learning for robotic vision: Open set recognition and active learning. In: International Conference on Robotics and Automation (ICRA). pp. 2924–2931 (2019)
40. Meyer, B.J., Harwood, B., Drummond, T.: Deep metric learning and image classification with nearest neighbour gaussian kernels. In: IEEE International Conference on Image Processing (ICIP). pp. 151–155 (2018)
41. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
42. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: International Conference on Learning Representations (ICLR) (2018)
43. Miyato, T., Koyama, M.: cgans with projection discriminator. In: International Conference on Learning Representations (ICLR) (2018)
44. Movshovitz-Attias, Y., Toshev, A., Leung, T.K., Ioffe, S., Singh, S.: No fuss distance metric learning using proxies. In: IEEE International Conference on Computer Vision (ICCV). pp. 360–368 (2017)
45. Nguyen, A., Clune, J., Bengio, Y., Dosovitskiy, A., Yosinski, J.: Plug & play generative networks: Conditional iterative generation of images in latent space. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
46. Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., Clune, J.: Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 3395–3403 (2016)
47. Nilsback, M.E., Zisserman, A.: Automated Flower Classification over a Large Number of Classes. In: Indian Conference on Computer Vision, Graphics and Image Processing (2008)
48. Odena, A.: Semi-supervised learning with generative adversarial networks. arXiv preprint arXiv:1606.01583 (2016)
49. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans. In: International Conference on Machine Learning (ICML). pp. 2642–2651 (2017)
50. Park, H., Yoo, Y., Kwak, N.: Mc-gan: Multi-conditional generative adversarial network for image synthesis. In: British Machine Vision Conference (BMVC) (2018)
51. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: International Conference on Learning Representations (ICLR) (2016)
52. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. In: International Conference on Machine Learning (ICML). pp. 1060–1069 (2016)

53. Reed, S.E., Akata, Z., Mohan, S., Tenka, S., Schiele, B., Lee, H.: Learning what and where to draw. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 217–225 (2016)
54. Rippel, O., Paluri, M., Dollar, P., Bourdev, L.: Metric learning with adaptive density discrimination. In: International Conference on Learning Representations (ICLR) (2016)
55. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., Chen, X.: Improved Techniques for Training GANs. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 2234–2242 (2016)
56. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 815–823 (2015)
57. Sohn, K.: Improved Deep Metric Learning with Multi-class N-pair Loss Objective. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 1857–1865 (2016)
58. Song, H.O., Jegelka, S., Rathod, V., Murphy, K.: Deep Metric Learning via Facility Location. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2206–2214 (2017)
59. Song, H.O., Xiang, Y., Jegelka, S., Savarese, S.: Deep Metric Learning via Lifted Structured Feature Embedding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4004–4012 (2016)
60. Tran, D., Ranganath, R., Blei, D.: Hierarchical implicit models and likelihood-free variational inference. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 5523–5533 (2017)
61. Tran, L., Yin, X., Liu, X.: Disentangled representation learning gan for pose-invariant face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1415–1424 (2017)
62. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
63. Xing, Y., Ge, Z., Zeng, R., Mahapatra, D., Seah, J., Law, M., Drummond, T.: Adversarial pulmonary pathology translation for pairwise chest x-ray data augmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). pp. 757–765 (2019)
64. Zakharov, E., Shysheya, A., Burkov, E., Lempitsky, V.: Few-shot adversarial learning of realistic neural talking head models. In: IEEE International Conference on Computer Vision (ICCV) (2019)
65. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks. In: International Conference on Machine Learning (ICML) (2019)
66. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.: Stackgan++: Realistic image synthesis with stacked generative adversarial networks. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2018)
67. Zhang, R., Che, T., Ghahramani, Z., Bengio, Y., Song, Y.: Metagan: An adversarial approach to few-shot learning. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 2365–2374 (2018)
68. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2223–2232 (2017)
69. Zieba, M., Wang, L.: Training triplet networks with gan. arXiv preprint arXiv:1704.02227 (2017)

A Additional Implementation Details

A.1 Network Architecture

The generator and discriminator architectures are shown in Tables 3a and 3b, respectively. Spectral normalisation [42] is used on all weights, except in the feature embedding conditional normalisation (*cNorm*) blocks. The structure of the up-sampling and down-sampling residual blocks (*ResBlocks*) are shown in Figures 14a and 14b, respectively. The baseline models follow the same architecture, with the only difference being the calculation of the normalisation layer scale and bias terms. The non-conditional baseline has no normalisation layers, while the class-conditional baseline uses a single conditioning embedding per class. A standard Resnet18 network [21] is used for the feature extractor, with the softmax layer and class-specific fully connected layer removed.

$\mathbf{z} \in \mathbb{R}^{128} \sim \mathcal{N}(0, 1)$ $\mathbf{f} \in \mathbb{R}^{512}, \mathbf{f} = F(\mathbf{x}), \mathbf{x} \sim p_d$
Linear $128 \rightarrow 512 \times 4 \times 4$
ResBlock Up $512 \rightarrow 512$
ResBlock Up $512 \rightarrow 256$
ResBlock Up $256 \rightarrow 256$
ResBlock Up $256 \rightarrow 128$
Self-Attention Block
ResBlock Up $128 \rightarrow 64$
ResBlock Up $64 \rightarrow 32$
Normalisation, ReLU
3×3 Conv $32 \rightarrow 3$
Tanh

(a) Generator.

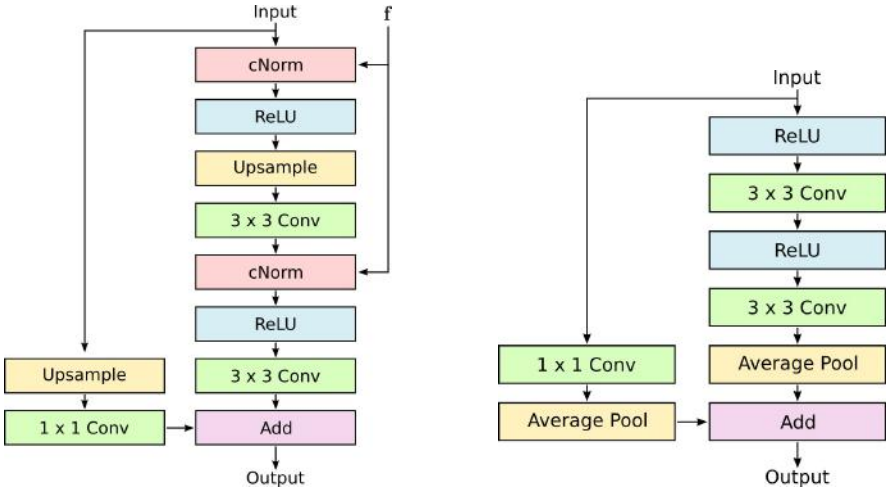
$\mathbf{x} \in \mathbb{R}^{256 \times 256 \times 3}$
3×3 Conv $3 \rightarrow 32$
ResBlock Down $32 \rightarrow 64$
ResBlock Down $64 \rightarrow 128$
Self-Attention Block
ResBlock Down $128 \rightarrow 256$
ResBlock Down $256 \rightarrow 256$
ResBlock Down $256 \rightarrow 512$
ResBlock Down $512 \rightarrow 512$
cNorm, ReLU
4×4 Conv $512 \rightarrow 1$

(b) Discriminator.

Table 3: Network architectures to generate 256×256 samples. The real data distribution is denoted as p_d.

A.2 Attribute Interpolation

In this section, we describe the method used to perform attribute and pose interpolation in more detail. The binary attribute labels are taken directly from



(a) ResBlock Up (b) ResBlock Down

Fig. 14: Structure of residual blocks. Upsampling layers (nearest neighbour) and downsampling layers (average pooling) change the scale by a factor of two.

the Celeba dataset [35]. Pose labels, for example left facing, right facing and forward facing, are found by using the facial landmark locations of the data. A Resnet18 network [21] is trained as a multi-label classifier on the training data using binary cross-entropy loss. To perform the interpolation, the positive and negative mean latent vectors and feature embeddings are found for each attribute and pose. This is achieved by predicting the attribute and pose labels of generated samples and grouping the associated latent and feature vectors. The unit vector that points from the positive group to the negative group for all attributes and poses are found for both the latent and feature spaces. To perform interpolation, an image is first generated using a source image and randomly sampled latent vector. For interpolation of a given attribute in the feature space, for example, the scaled attribute unit vector is added to the starting feature embedding. Samples are generated across a range of unit vector scaling terms.

B Additional Results

In this section, we include additional results to further evaluate the performance of our proposed method.

One-Shot Image Generation Figures 15 (Flowers102 [47]) and 16 (Celeba [35]) show samples generated when the generator is conditioned on feature embeddings extracted from novel class source images. Despite the classes being from

Fig. 15: Flowers102 novel class real source images (top row of each section) and resultant generated images (bottom two rows of each section). Although the species are not present during training, the fake images match the features of the real source images.

outside of the training distribution, the generated samples match the semantic features found in the source images.

Attribute Interpolation The method detailed in Section A.2 is used to perform interpolation between poses and attributes in Figure 17. Interpolations are shown in both the feature space and latent space. It can be seen that pose interpolation in the feature space has no impact on the pose in the generated samples. This indicates that pose is encoded only in the latent space. By contrast, attributes (age, bangs and gender) are encoded only in the feature space. Further pose and attribute interpolation can be observed in the included videos.

Random Interpolation Figures 18 and 19 show interpolation between two random latent vectors (horizontal direction) and two sampled novel class feature embeddings (vertical direction). It can be seen that only structural information changes when the latent vector is varied, while semantic information changes

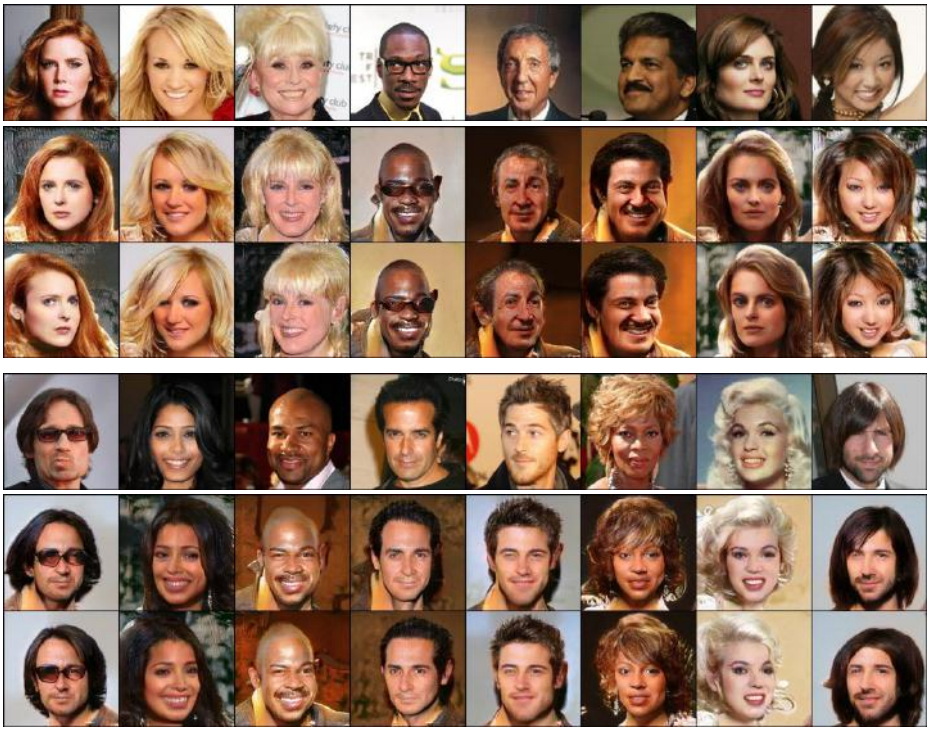
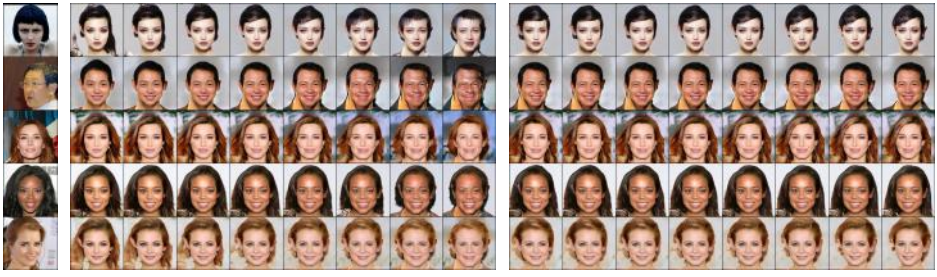


Fig. 16: Celeba novel class real source images (top row of each section) and resultant generated images (bottom two rows of each section). Although the identities are not present during training, the fake images match the features of the real source images.

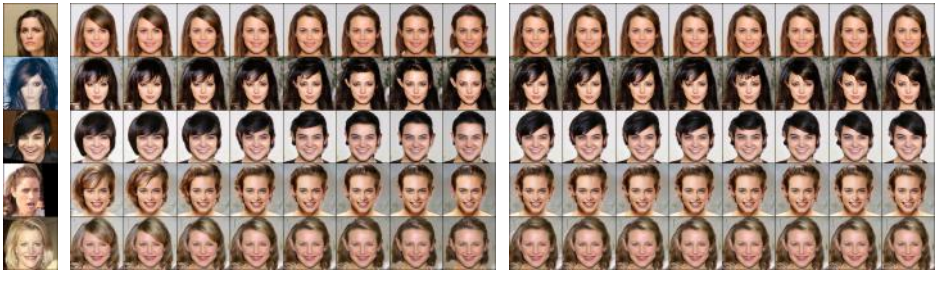
when the feature embedding is varied. Further random interpolation can be observed in the included videos.

Random Feature Sampling Further samples generated by randomly sampling the metric feature space are shown in Figure 20. Features are sampled using a single mean and standard deviation across all embedding dimensions. No class-level information or other labels are utilised.

Real. Feature space interpolation: pose. Latent space interpolation: pose.



Real. Feature space interpolation: age. Latent space interpolation: age.



Real. Feature space interpolation: bangs. Latent space interpolation: bangs.

Real. Feature space interpolation: gender. Latent space interpolation: gender.

Fig. 17: Pose is encoded only in the latent space, while age, hairstyle (bangs) and gender are encoded only in the feature space.

Fig. 18: Interpolation between two latent vectors (horizontal) and two feature embeddings (vertical). Feature embeddings are from novel classes.

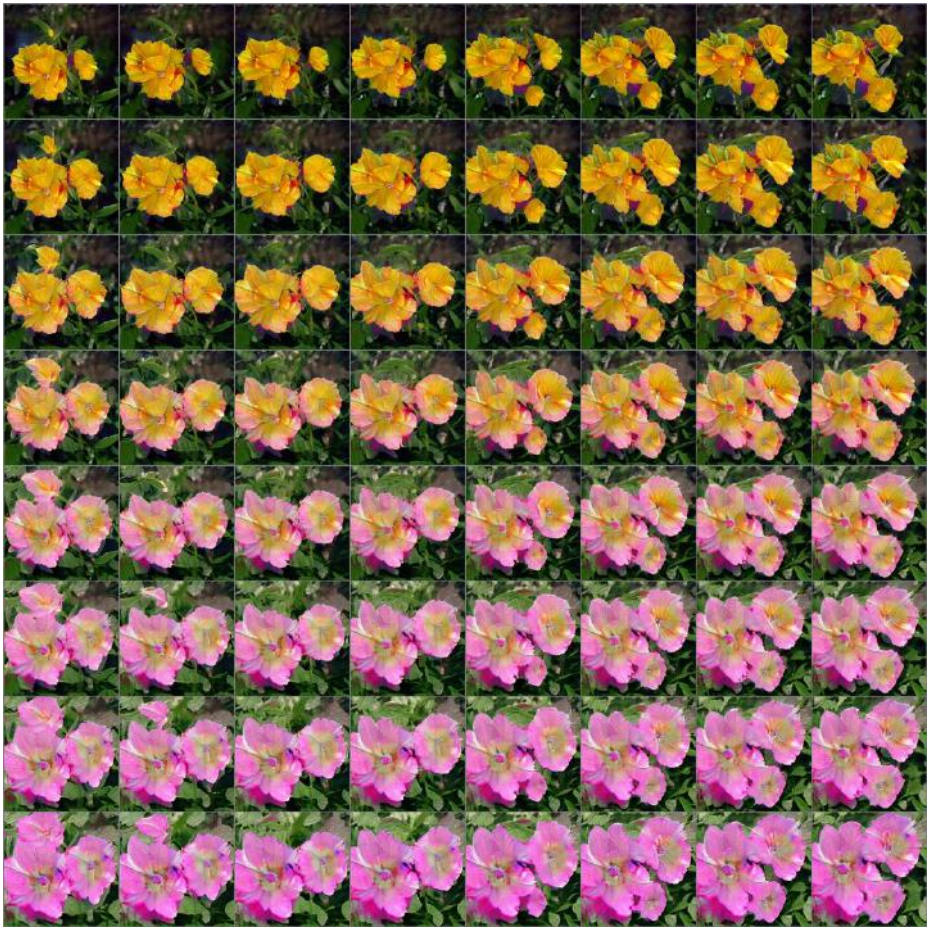


Fig. 19: Interpolation between two latent vectors (horizontal) and two feature embeddings (vertical). Feature embeddings are from novel classes.

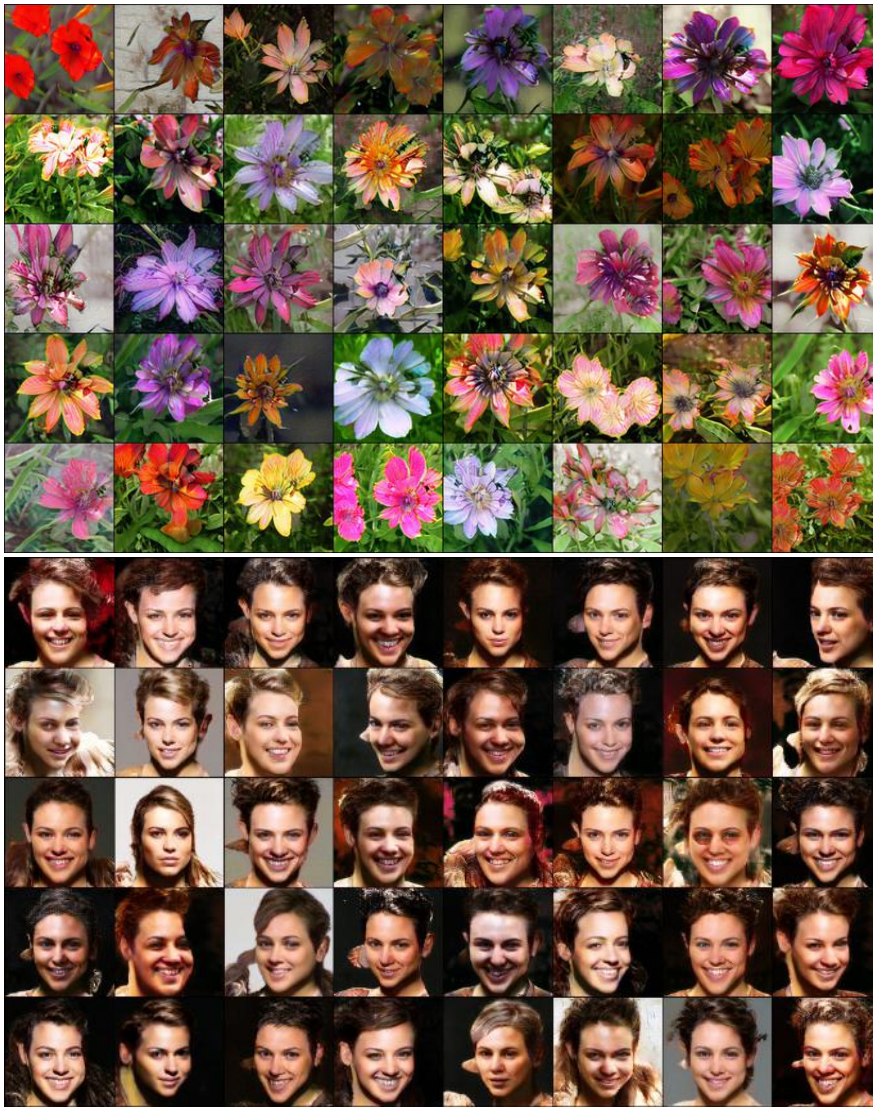


Fig. 20: Uncurated images generated by randomly sampling the metric feature space and latent space. No class-level information or other labels are used to generate these samples.