Supplementary Material for OpenGAN: Open Set Generative Adversarial Networks

A Additional Implementation Details

A.1 Network Architecture

The generator and discriminator architectures are shown in Tables 1a and 1b, respectively. Spectral normalisation [1] 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 1a and 1b, 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 [2] is used for the feature extractor, with the softmax layer and class-specific fully connected layer removed.

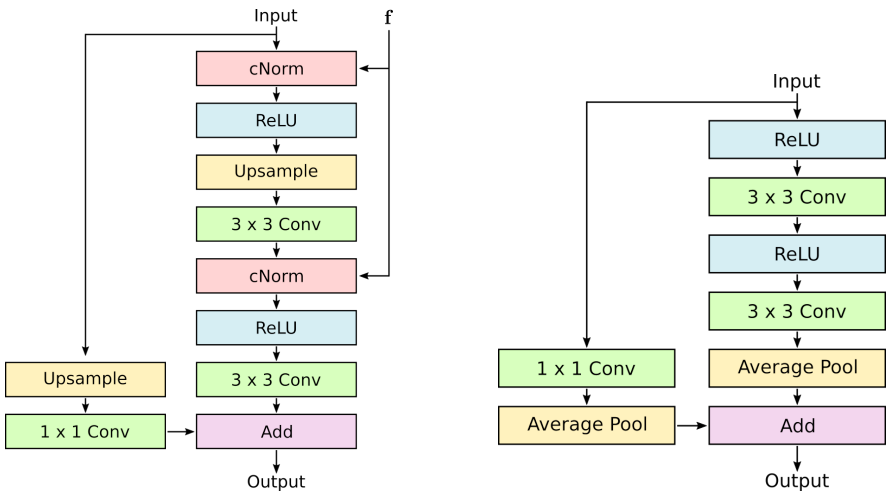
$\mathbf{z} \in \mathbb{R}^{128} \sim \mathcal{N}(0,1)$ $\mathbf{f} \in \mathbb{R}^{512}, \mathbf{f} = F(\mathbf{x}), \mathbf{x} \sim p_d$
Linear $128 \rightarrow 512 \times 4 \times 4$
ResBlock Up $512 \rightarrow 512$
ResBlock Up $512 \rightarrow 256$
ResBlock Up $256 \rightarrow 256$
ResBlock Up $256 \rightarrow 128$
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 1: Network architectures to generate 256×256 samples. The real data distribution is denoted as p_d.



(a) ResBlock Up (b) ResBlock Down

Fig. 1: Structure of residual blocks. Upsampling layers (nearest neighbour) and downsampling layers (average pooling) change the scale by a factor of two.

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 the Celeba dataset [3]. 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 [2] 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.

Fig. 2: 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.

One-Shot Image Generation Figures 2 (Flowers102 [4]) and 3 (Celeba [3]) show samples generated when the generator is conditioned on feature embeddings extracted from novel class source images. Despite the classes being from 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 4. 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.

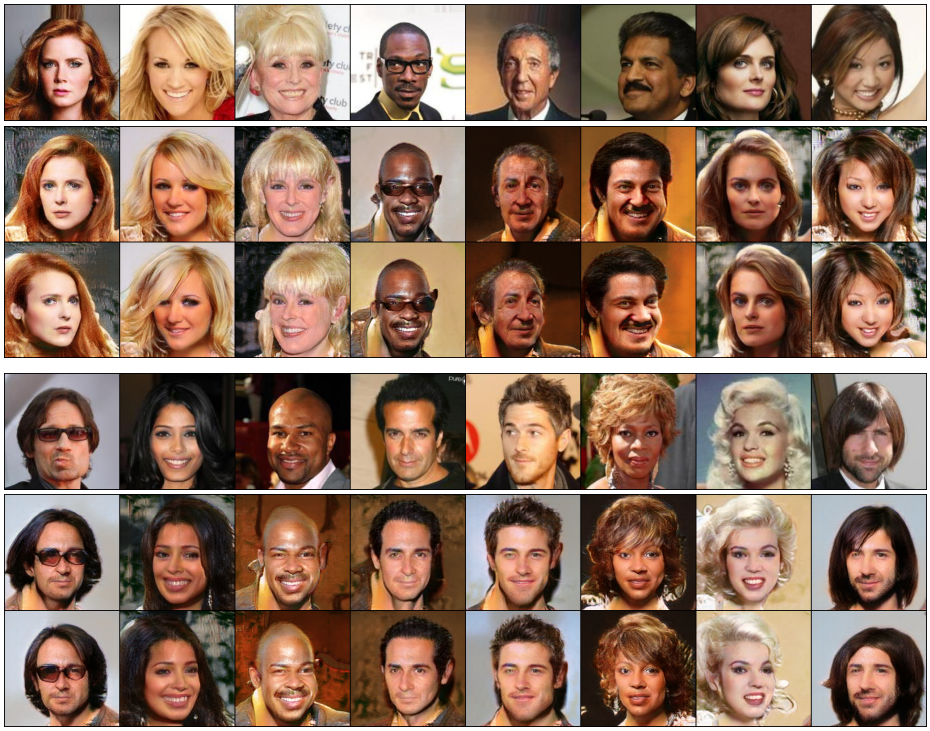


Fig. 3: Celeba novel class real source images (top row of each section) and resultant generated images (bottom two rows of each section). Although the identities are not present during training, the fake images match the features of the real source images.

Random Interpolation Figures 5 and 6 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 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 7. Features are sampled using a single mean and standard deviation across all embedding dimensions. No class-level information or other labels are utilised.

Fig. 4: Pose is encoded only in the latent space, while age, hairstyle (bangs) and gender are encoded only in the feature space.

Fig. 5: Interpolation between two latent vectors (horizontal) and two feature embeddings (vertical). Feature embeddings are from novel classes.

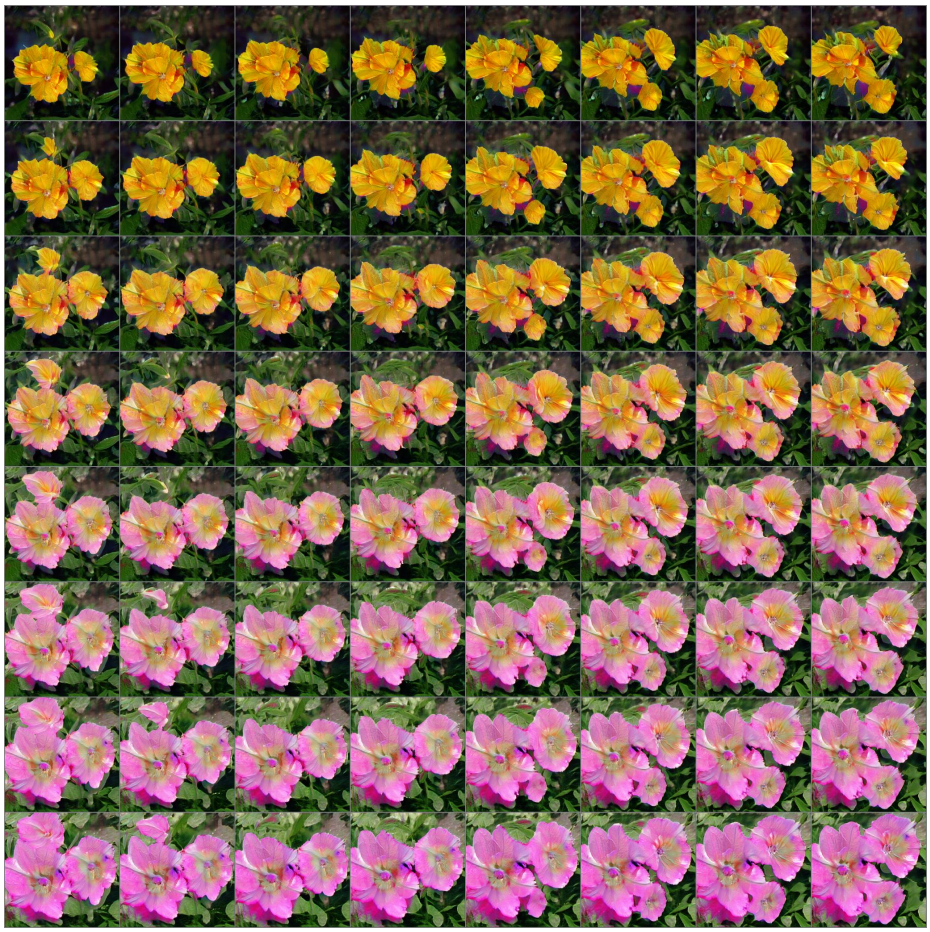


Fig. 6: Interpolation between two latent vectors (horizontal) and two feature embeddings (vertical). Feature embeddings are from novel classes.

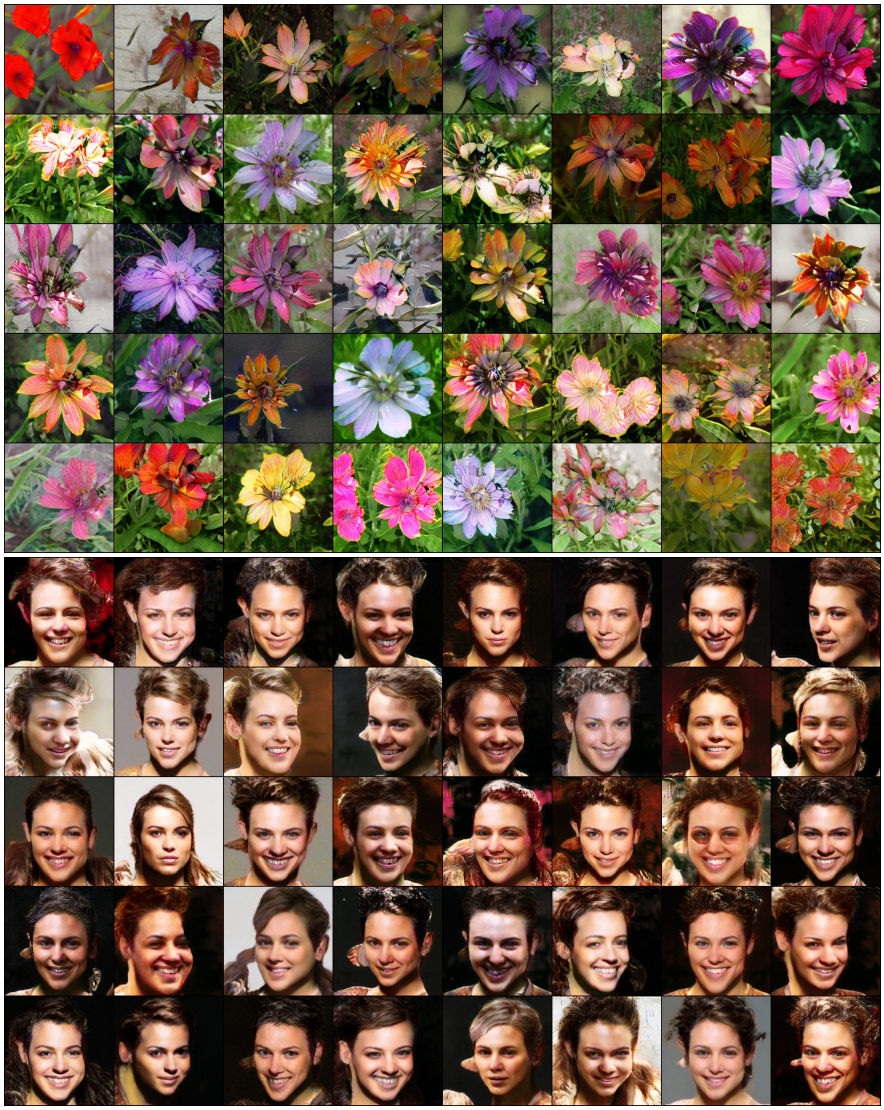


Fig. 7: Uncurated images generated by randomly sampling the metric feature space and latent space. No class-level information or other labels are used to generate these samples.

References

1. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: International Conference on Learning Representations (ICLR). (2018)
2. 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). (2016) 770–778
3. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: IEEE International Conference on Computer Vision (ICCV). (2015)
4. 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)