Few Shot Image Generation Using Conditional Set-Based GANs

Anonymous authors
Paper under double-blind review

Abstract

While there have been tremendous advances made in few-shot and zero-shot image generation in recent years, one area that remains comparatively underexplored is few-shot generation of images conditioned on sets of unseen images. Existing methods typically condition on a single image only and require strong assumptions about the similarity of the latent distribution of unseen classes relative to training classes. In contrast, we propose SetGAN - a conditional, set-based GAN that learns to generate sets of images conditioned on reference sets from unseen classes. SetGAN can combine information from multiple reference images, as well as generate diverse sets of images which mimic the factors of variation within the reference class. We also identify limitations of existing performance metrics for few-shot image generation, and discuss alternative performance metrics that can mitigate these problems.

1 Introduction

Few-shot and zero-shot learning has been an area of exploding interest in machine learning over the past few years. Transformer-based models such as GPT-4 (OpenAI, 2023) have achieved incredible leaps in performance in few-shot text generation, while diffusion-based Ho et al. (2020) image generation models such as DALL-E 3 (Betker et al., 2023) and Stable Diffusion (Rombach et al., 2022) have achieved remarkable success at zero-shot text-to-image generation. One area that remains comparatively underexplored, however, is few-shot or zero-shot image-to-image generation - particularly in the setting of generating images conditioned on *sets* of images.

We propose SetGAN - a novel image generation model that is trained to generate images conditioned on sets of reference images of unseen classes. The model learns to extract relevant features from the unseen reference sets, then generate high-quality, diverse images similar to the reference images at inference time. Once pretrained on a given image dataset, SetGAN can then generate any number of images for a variety of unseen reference classes, all without any further training or finetuning.

Conceptually, this model uses a similar framework to models such as DAGAN (Antoniou et al., 2017), following an adversarial learning approach where a "generator" model attempts to generate images conditioned on a given input image, and a "discriminator" model learns to distinguish between the generated images and other true images from the same class. The difference lies in the set-based nature of our model - SetGAN can condition its generations on multiple reference images rather than just a single image, and similarly generate multiple output images as well. This allows the model to better understand the variations within the reference class, as well as producing diverse sets of output images conditioned on that class. The discriminator is also able to compare the generated sets of images to the reference set, and judge the generated sets based not only on the individual images' similarity to the reference class, but also on the diversity and factors of variation within each set - leading to generations that more closely match the variations within the true reference class.

Existing works frequently rely on learning factors of variation within a typical reference class at training time, then applying those variations to a single image at inference time. This makes the assumption that the factors of variation within a given class at training time will be the same as for the unseen test classes - an assumption that does not always hold. Works that follow this methodology also have a tendency to produce

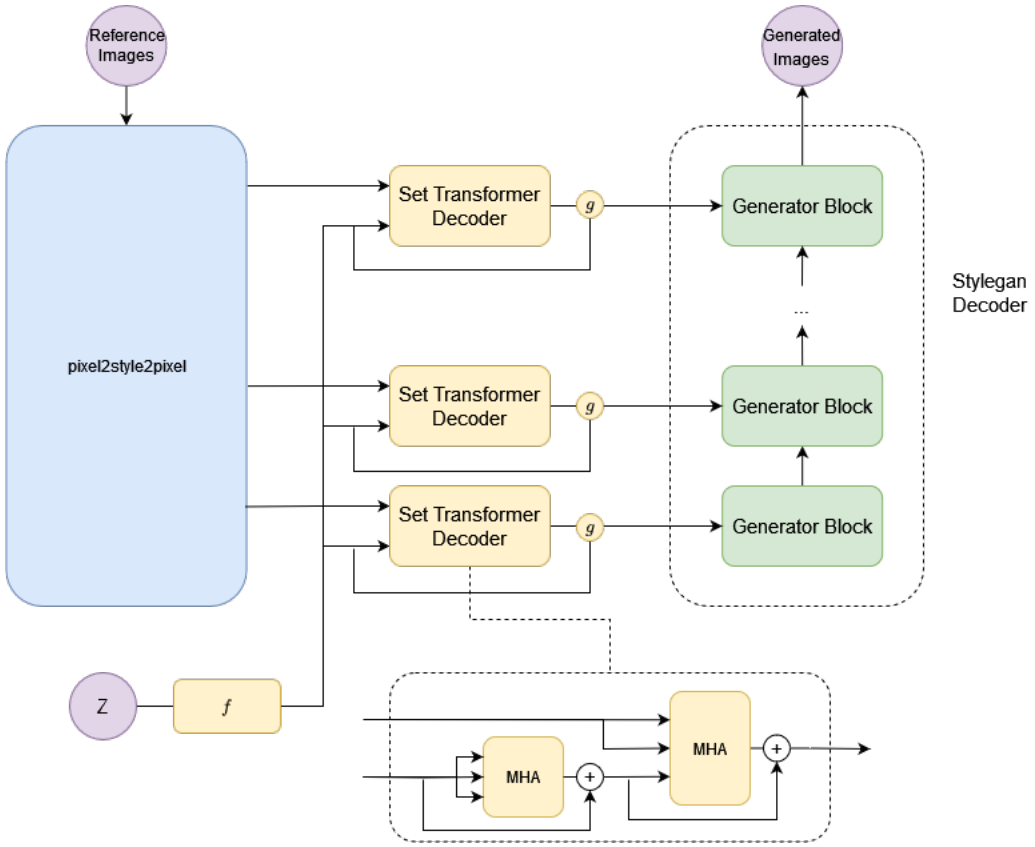


Figure 1: Diagram of the SetGAN generator. The pSp encoder maps each input image to the latent space $\mathcal{W}+$. The input style vectors are then passed through the StyleGAN2 mapping network, then passed to a series of conditioning networks which compute conditional styles for each layer of the decoder by attending to the appropriate output layer of the pSp encodings. These conditional styles then become the inputs to the StyleGAN2 generator, which decodes them into images.

generations that are highly similar to the reference image, limiting their diversity. Consider a training set of faces where each class consists of images of the same person under different poses and lighting conditions. If we want to generate faces similar to a reference set that contains images of different women that all have heavy eye shadow, those techniques will generate faces of the same women as the reference set with different poses and lighting conditions instead of different women with heavy eye shadow (see Section 5 and Figure 3). SetGAN does not suffer from these limitations, and can generate truly novel and diverse outputs that reflect the factors of variation of the reference set instead of the training set and without simply reproducing elements of a single input image.

2 Related Work

2.1 Few-shot GANs

Previous works on few-shot image generation using GANs generally fall into three categories: optimization-based methods, fusion-based methods, and transformation-based methods. Optimization-based methods (Clouâtre & Demers, 2019; Liang et al., 2020) use meta-learning techniques (Finn et al., 2017) to fine-tune their generative models on small amounts of data, but do not produce results competitive with other approaches. Fusion-based methods (Hong et al., 2020b; Gu et al., 2021; Yang et al., 2022) condition on several input images by starting with a single base image and incorporating local features from other reference images. These methods are highly dependent on the images they condition on and sometimes struggle

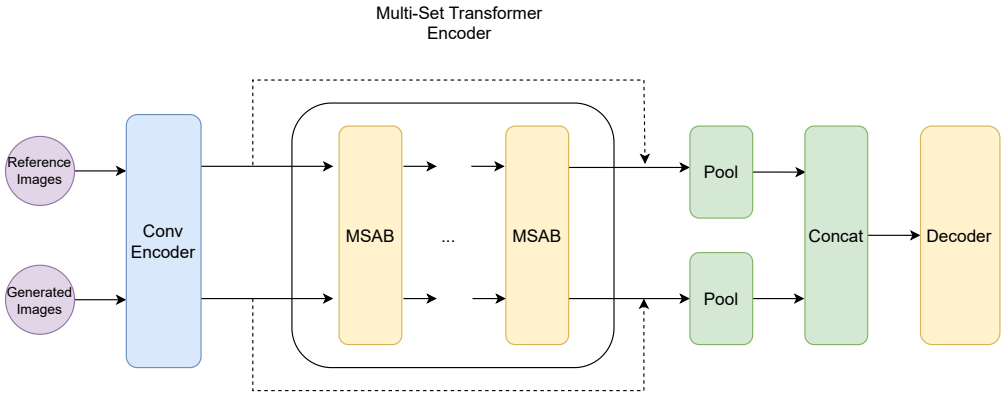


Figure 2: Diagram of the SetGAN discriminator. Sets of input images are encoded as fixed-size vectors using a convolutional network. These sets of vectors are then passed through a Multi-Set Transformer (Selby et al., 2022) consisting of several multi-set attention blocks, followed by a pooling operation performed on each set. These outputs are then concatenated and passed through a feedforward decoder layer to produce a scalar output.

to generalize beyond the features in the input images. Transformation-based methods (Ding et al., 2022; Hong et al., 2020a; Antoniou et al., 2017) learn transformations during training that mimic the typical factors of variation within each training class, then apply these learned transformations to a single test image. These methods can be highly successful at one-shot image generation, but make strong assumptions about the similarity in factors of variation between classes that may not generalize to more diverse datasets. Using only a single image to condition on can also limit diversity, as each generation may be only a slight transformation of the given input image.

2.2 Diffusion models

Many diffusion-based approaches such as DALL-E 3 (Betker et al., 2023) and Stable Diffusion (Rombach et al., 2022) have achieved incredible success at text-to-image generation, generating diverse high-resolution images from a wide variety of text-based prompts. While some limited equivalents exist for image-to-image generation such as inpainting Rombach et al. (2022) or image translation Saharia et al. (2022); Sasaki et al. (2021), no such large-equivalents exist for large-scale true few-shot generation of images conditioned on sets of unseen images. Giannone et al. (2022) do propose a framework for few-shot generation with diffusion models, however their model is tested only on very low-resolution datasets. Their model is also extremely slow to perform inference even at lower resolution scales, and this problem is compounded at higher resolutions.

2.3 Image translation

A closely related task to few-shot image generation is image-to-image translation. In this task, the goal is to translate images from one domain to a new domain, often in a few-shot setting. This frequently takes the form of adapting models pretrained on the source domain to the target domain via a minimal number of examples (Li et al., 2020; Ojha et al., 2021). While this approach to few-shot image translation is a distinct task from few-shot image *generation*, some approaches such as FUNIT (Liu et al., 2019b) have combined these approaches by seeking to translate images between different classes of the same dataset.

2.4 Set-based approaches in GANs

Ferrero et al. (2022) proposed an approach where the discriminator is allowed to make decisions based on a set of samples from either training data or the generator in order to increase stability and prevent mode collapse. While this work does also examine the idea of leveraging equivariances for generation, it focuses on improving the stability of unconditional generation rather than performing conditional set-based generation.

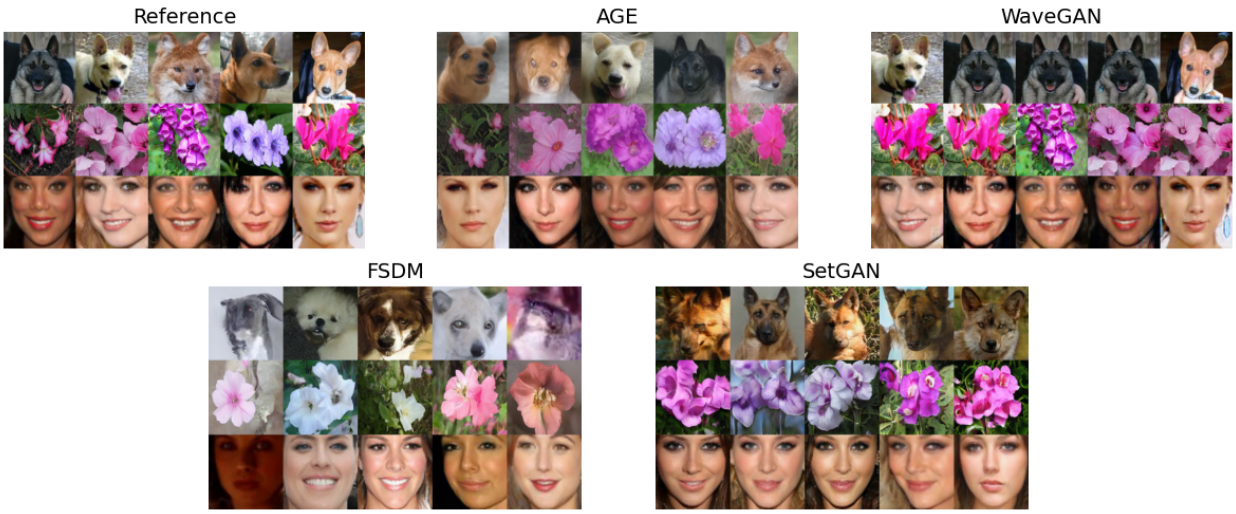


Figure 3: Examples of generations using images across many different test classes that share similarities according to other features - e.g. women with heavy eye makeup, animals with long upward-pointing ears, or clusters of pink and purple flowers. SetGAN generates diverse output images that faithfully reproduce these features, whereas other baselines either copy the reference images or generate images which are not faithful to the shared features.

3 Methods

3.1 Background

3.1.1 Few-shot image generation

Few-shot image generation consists of a dataset $\mathcal{D}$ divided into a number of classes $\{\mathcal{C}_i\}$, which are each composed of some $n_{\mathcal{C}_i}$ images. These classes are partitioned into a disjoint training set $\mathcal{D}_{\text{train}}$ and test set $\mathcal{D}_{\text{test}}$. At inference time, a class $\mathcal{C} \in \mathcal{D}_{\text{test}}$ is sampled from the dataset. From this class, k images are sampled to become the *reference set* $\mathcal{C}_{\text{ref}}$, with the rest forming the holdout *evaluation set* $\mathcal{C}_{\text{eval}}$. The goal is to generate additional images $\mathcal{C}_{\text{gen}}|\mathcal{C}_{\text{ref}}$ such that the difference between the $\mathcal{C}_{\text{gen}}$ and $\mathcal{C}_{\text{eval}}$ is minimized, according to some sort of distance metric (e.g. the Frechet Inception Distance, see Section 4.4).

3.1.2 Conditional GANs

Generative Adversarial Networks (or GANs) typically follow an adversarial training paradigm in which two networks are trained jointly: a "discriminator" D and a "generator" G. Given a dataset $\mathcal{D}$, these two networks train by playing a minimax game, often formulated approximately as follows:

$$\min_G \max_D \mathbb{E}_{x\sim\mathcal{D}} \log D(x) + \mathbb{E}_{z\sim p(z)} \log\left(1 - D(G(z))\right) \tag{1}$$

When applying a GAN to a conditional few-shot generation regime, this approach must be modified. A common way to proceed is to condition the generations on a single image. This means that the generator is no longer a mapping solely from the latent prior onto the data distribution $\mathcal{D}$, but rather a *conditional* mapping $G(z|x)$. The discriminator can then be viewed as a form of similarity function $D(x, y)$ between two images x and y. During training, a class $\mathcal{C} \sim \mathcal{D}_{train}$ is sampled, and from this class are drawn two images $x, y \sim \mathcal{C}$. A modified minimax game is then played, of the form:

$$\min_G \max_D \mathbb{E}_{\mathcal{C}\sim\mathcal{D}} \mathbb{E}_{x,y\sim\mathcal{C}} [\log D(x, y) + \mathbb{E}_{z\sim p(z)} \log\left(1 - D(G(z|x), y)\right)] \tag{2}$$

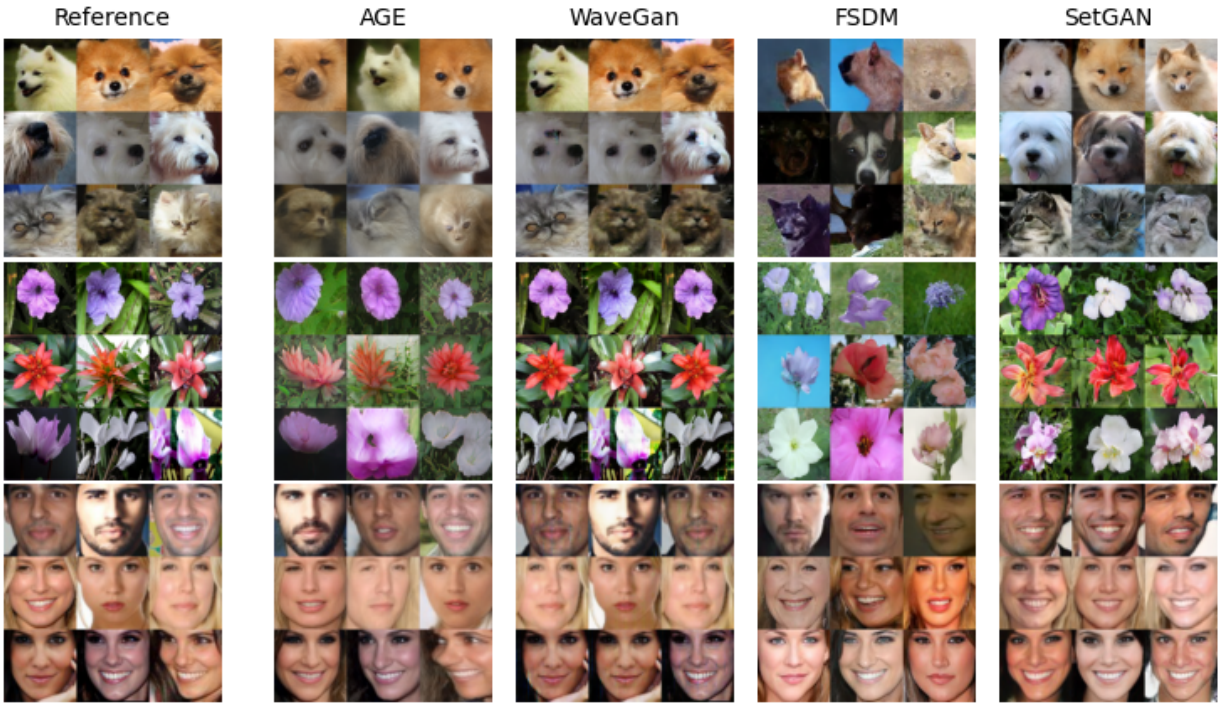


Figure 4: Generations from AGE, FSDM, WaveGAN and SetGAN conditioned on 3 reference images from unseen classes of each of the Animal Faces, Flowers and VGGFace datasets.

This is the training regime used by methods such as DAGAN (Antoniou et al., 2017), DeltaGAN (Hong et al., 2020a), and AGE (Ding et al., 2022). While this does provide a method to train a conditional GAN, it also has limitations. By conditioning on a single image only, it can be difficult for the model to generate images that are faithful to the reference class. Many of these methods assume that the latent factors of variation within the classes at inference time will follow the same distribution as those of the training classes, and thus seek to perform transformations on the given input image to follow these factors of variation (Ding et al., 2022; Hong et al., 2020a).

3.1.3 Set-based models

In an ideal case, the model should be able to incorporate information from all reference images in order to better understand the latent space and generate diverse, high quality samples in a generalizable way. In order to do this, the model must be conditioned on a *set* of input images, rather than a single image. As such, the model must obey the restriction of *permutation equivariance* - i.e. for all permutations π of the reference images R, $G(\pi(R)) = G(R)$.

The problem of constructing neural networks conditioned on sets of inputs while obeying restrictions of permutation-invariance or -equivariance has been discussed in previous works such as Zaheer et al. (2017), Lee et al. (2019) and Selby et al. (2022). As discussed in Lee et al. (2019), the simplest and most common model architecture which naturally conforms to these constraints is the transformer (Vaswani et al., 2017). The building block of the transformer is the so-called "attention mechanism", which takes the form

$$\mathrm{MHA}(X, Y) = \sigma\left((XW_Q)(YW_K)^T\right) YW_V W_O \tag{3}$$

This structure is naturally permutation-equivariant with respect to the queries X and permutation-invariant with respect to the keys Y, since for any permutation π, $\mathrm{MHA}(\pi(X), Y) = \pi(\mathrm{MHA}(X, Y))$ and $\mathrm{MHA}(X, \pi(Y)) = \mathrm{MHA}(X, Y)$. The "transformer decoder" architecture proposed by Vaswani et al. (2017) retains these properties, and constitutes a mapping $f : \mathbb{R}^{n\times d} \times \mathbb{R}^{m\times d} \to \mathbb{R}^{n\times d}$.

3.2 Set-GAN

Instead of conditioning on a single image only, SetGAN conditions its generations on a set of images from the same class, and seeks to generate a set of output images similar to these input images.

Formally, we again consider a setting where there is a dataset consisting of a number of classes $\{\mathcal{C}_i\}$, which are each composed of some $n_{\mathcal{C}_i}$ images. During training, a class $\mathcal{C} \sim \mathcal{D}_{train}$ is sampled, and from this class are drawn two (disjoint) sets of images: a *reference set* $R \in \mathcal{C}^n$, and a *candidate set* $C \in \mathcal{C}^m$. The generator G produces m generated images $G(R; m)$ conditioned on these reference images, which are then compared to the candidate set C by a discriminator D, which plays the following minimax game with the generator:

$$\min_G \max_D \mathbb{E}_{\mathcal{C}\sim\mathcal{D}_{train}} \mathbb{E}_{R\sim\mathcal{C}^n, C\sim\mathcal{C}^m} \left[\log D(R, C)\right] + \log\left(1 - D(R, G(R))\right) \tag{4}$$

3.3 Architecture

3.3.1 Generator

The generator model follows an encoder-decoder structure similar to that of a U-Net (Ronneberger et al., 2015). We take StyleGAN2's generator to be our base decoder architecture, which maps a series of $k = 18$[1] 512-dimensional *style vectors* to a single output image, with each style vector controlling the convolutions at a particular stage of the decoding. It has become common to refer to the extended latent space formed by the concatenation of these k vectors as $\mathcal{W}+$. Similar to Ding et al. (2022), we take the pixel2style2pixel (pSp) encoder proposed by Richardson et al. (2021) to be our encoder model, which maps a single input image into the space $\mathcal{W}+$ (although this could also be done with other similar encoders such as E4E (Tov et al., 2021) or ReStyle (Alaluf et al., 2021)). We augment this encoder-decoder model with a series of attention-based conditioning networks, consisting of a stack of 2 transformer decoder blocks for each of the k style vectors, each surrounded by a skip connection.

Given a set of n reference images, we encode each image R_i into a latent code: $C_i = \text{pSp}(R_i) = \{c_i^0, ..., c_i^k\}$, with the notation c_i^ℓ for style ℓ of the encoding of image i, and $C^\ell = \{c_i^\ell\}$. To generate a set of m candidate images, we then sample m noise vectors $Z = z_{1,...,m} \sim N(0, 1)$. These are then passed through the decoder's mapping network to generate the base style vectors $W = \{f(z_j)\}$, in the same fashion as StyleGAN. Now, the model takes the base style vectors W and transforms them by attending to the features of the reference encodings C. At each layer ℓ, the model computes the corresponding *conditional style vector*:

$$\omega^\ell = g^\ell(W, T^\ell(W, C^\ell)) \tag{5}$$

where T^ℓ is the transformer block associated with the ℓ-th style vector, and g^ℓ is a linear layer applied to the concatenation of the base style vector with the output of the attention blocks. These k conditional style vectors then form our conditional encoding $\omega \in \mathcal{W}+$, which becomes the input to the StyleGAN2 decoder.

3.3.2 Discriminator

The discriminator now takes the form $D : \mathbb{R}^{n\times d} \times \mathbb{R}^{m\times d} \to \mathbb{R}$, mapping an input reference set R and candidate set C to a single scalar output. To do this, we must use an architecture that can take as input multiple permutation-invariant sets. For this, we use the Multi-Set Transformer network proposed by (Selby et al., 2022). The input images are first passed through a convolutional encoder to encode each image within the two input sets as fixed-sized vectors, then passed through the multi-set transformer network. Finally, the two pooled output vectors are concatenated and fed to a linear output head. Skip connections are used to connect the outputs of the convolutional encoder to the latent vectors just before the pooling layer of the multi-set transformer. We use the convolutional architecture of the StyleGAN2 discriminator as the architecture for our discriminator encoder.

[1]Note that the default 18 style vectors correspond to a generation size of 1024x1024 px. Our experiments use a generation size of 256x256, and thus in practice use a truncated $\mathcal{W}+$ space of 14 vectors.

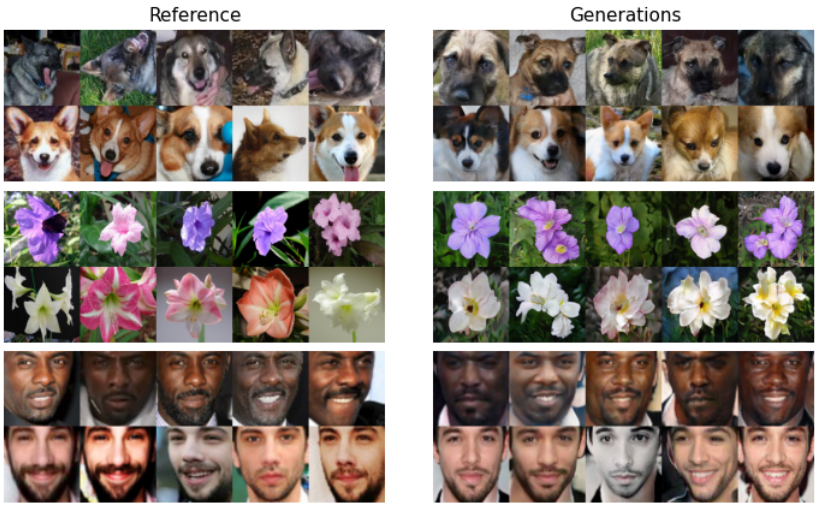


Figure 5: Additional generations from SetGAN using reference sets of 5 images.

4 Experiments

4.1 Setup

We first pretrain a StyleGAN2 model (Karras et al., 2020) on the given dataset at 256x256 resolution. Then, we train a pSp (Richardson et al., 2021) encoder to perform GAN inversion on the pretrained StyleGAN2 model to act as our encoder. These pretrained models are used to instantiate the encoder and decoder for our generator, and are then frozen. The discriminator from the StyleGAN2 model is also used to initialize the encoder for our multi-set discriminator model. These models are then trained following Eq. 4 until convergence. We use the base training scheme of StyleGAN2 (Karras et al., 2020) to train SetGAN, using a non-saturating loss with R1 gradient penalty ($\lambda = 10$) and path length regularization. Reference and candidate sizes are sampled uniformly from size 7-10 and 4-6 respectively, so that the model does not learn to assume a specific input size. Models are trained on NVIDIA A40 GPUs with the ADAM optimizer, with a batch size of 2 and learning rate 1e-3.

For inference, we follow similar models such as StyleGAN2 and apply latent space truncation, shifting the latent style vectors towards well-explored areas near the mean by a constant factor. Details of how this is applied are included in the supplementary material.

4.2 Datasets

In keeping with prior works (Hong et al., 2020a; Ding et al., 2022; Yang et al., 2022; Gu et al., 2021), we choose to report results on the Animal Faces (Liu et al., 2019a), Flowers (Nilsback & Zisserman, 2008) and VGGFace (Cao et al., 2018) datasets. We use the same train and evaluation splits proposed in Hong et al. (2020a) on Animal Faces and Flowers. For VGGFace, we restrict the evaluation set to the final 53 classes due to the computational requirements of inference for the FSDM baseline.

4.3 Baselines

Due to significant inconsistencies with existing results and methodologies (see appendix for details), we chose a selection of the highest performing models from the literature as baselines and computed metrics for each model ourselves by running the provided models under identical settings to ensure a fair comparison. Code and checkpoints provided by the authors were used wherever possible. We selected the AGE (Ding et al., 2022) and WaveGAN (Yang et al., 2022) models as representative of the highest-performing GAN-based approaches in the literature, as well as the diffusion-based approach FSDM (Giannone et al., 2022).

	$\text{MIFID}_{\text{Inc}}$			$\text{MIFID}_{\text{CLIP}}$			LPIPS		
	1	3	10	1	3	10	1	3	10
				Animal Faces					
AGE	71.35	62.23	56.55	14.09	12.77	11.74	0.4027	0.5095	0.5504
WaveGAN	2327.29	1057.39	529.08	603.44	242.81	136.09	0.0000	0.4211	0.5556
FSDM	75.68	73.93	77.37	8.78	8.59	10.38	0.6039	0.6076	0.6086
SetGAN	***61.51***	***52.34***	***47.18***	***6.56***	***5.84***	***5.28***	***0.6144***	***0.6154***	***0.6181***
				Flowers					
AGE	81.87	70.15	65.48	16.82	15.03	14.31	0.3790	0.5528	0.6078
WaveGAN	2653.56	1305.31	699.96	851.14	373.62	182.11	0.0000	0.4844	0.6345
FSDM	69.25	62.35	61.47	10.69	10.26	10.18	***0.6809***	***0.6985***	***0.7042***
SetGAN	***62.44***	***59.84***	***59.31***	**10.68**	**9.79**	***9.88***	0.6166	0.6240	0.6281
				VGGFace					
AGE	22.12	18.39	16.76	8.20	6.51	5.94	0.2604	0.3693	0.4063
WaveGAN	852.70	36.97	23.12	17.50	9.40	6.65	0.0000	0.3246	0.4301
FSDM	10.51	11.26	12.48	***3.28***	3.47	3.76	0.4509	0.4477	0.4471
SetGAN	***9.60***	***7.93***	***7.83***	4.16	***3.12***	***2.82***	***0.4633***	***0.4614***	***0.4712***

Table 1: Scores for conditional generation on the Animal Faces, Flowers and VGGFace datasets for each of the four baselines, conditioned on reference sets of size 1, 3 and 10. Results were averaged over three different random partitions of the test set into D_{eval} and D_{ref}. Lower scores are better for MIFID, higher is better for LPIPS. The best score in each category is bolded. Scores that exceed all others by at least one standard deviation are italicized.

4.4 Evaluation procedure and metrics

During evaluation, each test class $\mathcal{C} \in \mathcal{D}_{\text{test}}$ is partitioned into a reference set $\mathcal{C}_{\text{ref}}$ of size n_{ref} and evaluation set $\mathcal{C}_{\text{eval}}$ of size n_{eval}. For each such class, the model is used to generate n_{gen} new images, conditioned on images from $\mathcal{C}_{\text{ref}}$, to form $\mathcal{C}_{\text{gen}}$. For some metrics (such as FID or MIFID), these images are then aggregated into a single D_{eval} and D_{gen}. These image sets are then used to evaluate the generations using a variety of metrics. For our experiments, $n_{\text{eval}} = n_{\text{gen}} = 128$, and n_{ref} varied by experiment (see Section 5 for further details). If the number of images in a given evaluation set was lower than 128, all images were used. Each of these evaluations was performed three times with different randomly chosen partitions for each class.

The most common metrics used to evaluate the quality of models trained to perform few-shot image generation are the Frechet Inception Distance (FID) (Heusel et al., 2018), and Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018). FID measures the statistical similarity between distributions of embedded vectors corresponding to the evaluation set and generated sets respectively, and is often used as a measure of generation quality/fidelity. LPIPS is a metric used to measure perceptual similarity between pairs of images via the distance between their encodings under the pretrained VGG network. This is used as a metric for the diversity of generated images by computing the average pairwise distance between pairs of generated images within each class.

4.4.1 Limitations of existing metrics

While the aforementioned FID and LPIPS scores are the most widely-used metrics among existing literature, these metrics have significant flaws - particularly FID. Existing works such as Rangwani et al. (2023) and Kynkäänniemi et al. (2023) have already identified flaws in the FID metric related to its bias towards particular features specific to the ImageNet classes it was trained on, leading to arbitrary manipulation of scores via imperceptible changes in generated images. Rangwani et al. (2023) also demonstrate that traditional FID scores sometimes strongly emphasize fidelity over diversity in few-shot generation, and propose FID_{CLIP}

in order to address this issue - a modification to the FID method using the large multi-modal CLIP model (Radford et al., 2021) in place of the Inception backbone.

While FID_{CLIP} is an improvement over traditional Inception-based FID scores in some respects, it does not wholly solve this problem. In our experiments, we found that models that generate identical or nearly identical copies of the reference images achieved extremely low FID scores. To test this, we measured the FID scores between the evaluation sets and generated sets constructed solely by copying N random images sampled with replacement from the reference set (denoted as the "Copy" baseline). We also tried the same experiment if the copied images were subjected to a small, imperceptible level of Gaussian noise (denoted as the "Noisy" baseline). We compare these scores to the best scores among all trained models[2], as well as a theoretical maximum score given by comparing two partitions of the test set. As shown in Table 2, this baseline of simply copying the reference images achieves FID scores close to the theoretical maximum, and matches or exceeds the score of the best trained baseline in almost every case. As a result, we conclude that *traditional FID scores are not a reliable metric for measuring the performance of few-shot generation.*

	FID	FID_{CLIP}	MiFID	$\text{MiFID}_{\text{CLIP}}$
Animal Faces				
Best Model	46.20	5.02	46.20	5.02
Noisy	24.05	6.94	109.66	14.44
Copy	20.44	1.59	17714.97	1335.72
True	13.56	1.05	13.56	1.05
Flowers				
Best Model	57.12	9.30	57.75	9.30
Noisy	37.06	4.21	394.61	22.37
Copy	36.98	2.56	23408.14	1737.27
True	30.18	1.69	30.18	1.69
VGGFace				
Best Model	8.87	2.95	8.87	2.95
Noisy	47.96	12.20	56.51	12.65
Copy	9.54	0.77	4849.92	438.17
True	7.17	0.58	7.15	0.58

Table 2: Scores for synthetic baselines using a variety of performance metrics. Methods that simply copy the reference set ("Noisy" and "Copy") are disproportionately favored by many scoring methods, outperforming most trained models and even approaching the score for the true test set. MIFID scores are discussed in section 4.4.2.

4.4.2 Alternative metrics

These issues have also been identified in many other previous works, in the context of training set memorization. Works such as Gulrajani et al. (2020), Bai et al. (2021) and Jiralerspong et al. (2023) have discussed the tendency for traditional GAN evaluation metrics such as FID to overvalue fidelity and fail to penalize training set memorization. While these tools were proposed to measure generalization beyond the training set, they can also be equivalently applied here in order to measure generalization beyond the reference set. These works propose a wide range of different possible evaluation metrics as solutions to this problem, but most have key limitations that prevent them from being effective in this case. We choose to focus on MiFID (Bai et al., 2021), but a more detailed discussion of the other approaches and their unsuitability for our purposes is included in the supplementary material.

MIFID

MiFID uses the standard Frechet Inception Distance, scaled by a multiplicative penalty calculated from the similarities between the generations and the reference images:

$$\text{MiFID}(S_g, S_t) = m_\tau(S_g, S_t) \cdot \text{FID}(S_g, S_t) \tag{6}$$

wherein S_g is the generated set, S_t is the training set (or reference set, in the case of conditional generation), FID is the standard Frechet Inception Distance, and m_τ is the penalty factor. Specifically, m_τ is defined by:

[2]We exclude WaveGAN from this, given WaveGAN's propensity to also generate nearly-identical copies of the reference images.

$$s(S_g, S_t) = \frac{1}{|S_g|} \sum_{x_g \in S_g} \min_{x_t \in S_t} 1 - \frac{|\langle x_g, x_t \rangle|}{|x_g| \cdot |x_t|} \tag{7}$$

$$m_\tau = \begin{cases} \frac{1}{s(S_g, S_t) + \epsilon} & s(S_g, S_t) < \tau \\ 1 & \text{else} \end{cases} \tag{8}$$

This metric penalizes models that simply reproduce reference images by adding a multiplicative penalty based on the average cosine similarity between the generated images and the nearest reference image. As shown in Table 2, this metric successfully penalizes models that simply copy the inputs, while keeping the original FID scores otherwise intact.

We adopt this metric as a drop-in replacement for FID, with the threshold τ determined by the average scale of the test set S_{test}. For each dataset, we divide the test set into two partitions of equal size, S_1 and S_2, then calculate the base score value $\tau_0 = s(S_1, S_2)$ using Equation 7. To ensure that models which produce results on a similar scale of variation to the test set are not unfairly penalized, we penalize only models whose scores are at least one standard deviation lower than the mean similarity scale (i.e. $\tau = \tau_0 - \sigma$, where σ is the standard deviation of the summand in Eq. 7).

5 Results

5.1 Quantitative Results

Results for all baselines are shown in Table 1. Results are shown for reference sizes of 1, 3 and 10, across each of the 3 datasets. As shown in the table, SetGAN outperforms all existing baselines across nearly all datasets and reference sizes - in some cases by significant margins. In addition to SetGAN's results being of high fidelity and quality, they are also highly diverse - as shown by its high LPIPS scores across all datasets. The only model to achieve a greater diversity score is FSDM on the Flowers dataset, which is likely due to its poor fidelity with the reference class (see section 5.2).

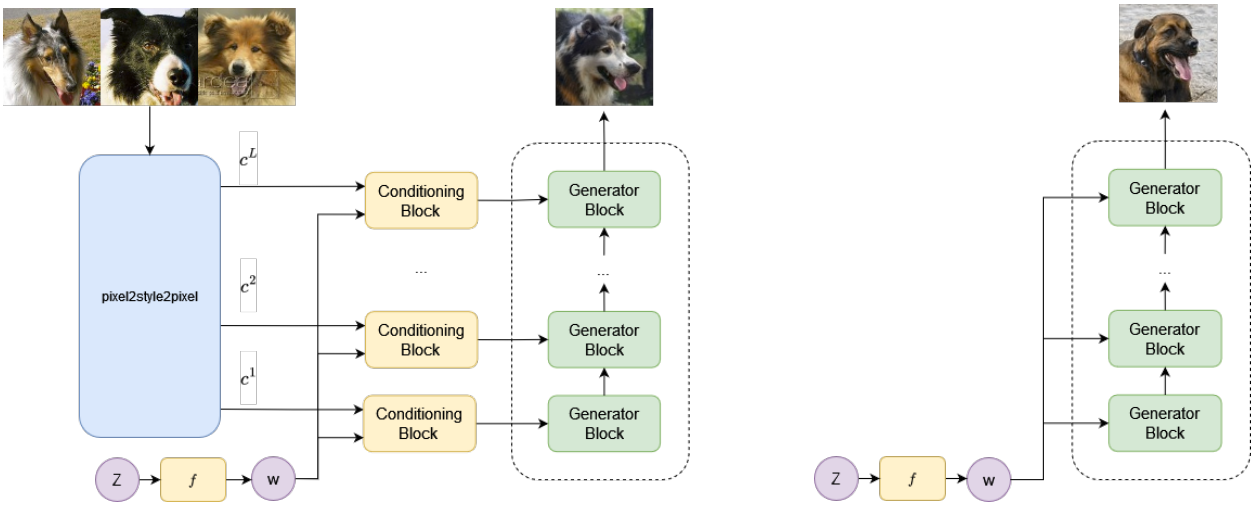
Notably, WaveGAN performs markedly worse than the other models under the MiFID score - largely due to it being heavily penalized for its tendency to produce nearly-indistinguishable copies of the reference images (see 5.2). While other models such as AGE often produce images very similar to the reference images, they were not copies, and as such did not fall under the threshold to be penalized.

While Inception-based scores and CLIP-based scores generally ranked models similarly, there were some cases where they demonstrated interesting differences - particularly on the VGGFace dataset. One possible explanation for this might be that the Inception network trained on ImageNet-1k data in which images of human faces were infrequent - unlike CLIP, which used many diverse image datasets from across the internet.

5.2 Qualitative Results

5.2.1 Test images with similar factors of variation to the training classes

Figure 4 shows images generated by the four models conditioned on reference images from the test classes of each of the three datasets considered. For these experiments, all reference images were drawn from the same unseen test class - measuring the models' effectiveness at generalization along similar factors of variation to the training classes. As shown in the figure, SetGAN generates diverse, high quality images, and avoids many of the struggles that other models demonstrate. Models such as AGE and WaveGAN often simply copy one of the input images, or generate small, subtle variations on it. This causes their generations to be limited in diversity, particularly when conditioned on only a small number of images. WaveGAN in particular very frequently copies the reference image almost exactly, differing from it only in imperceptible high-frequency perturbations. FSDM does succeed at generating diverse images, but often struggles to closely match the input class. This is particularly notable in the results from the flowers dataset, where its generations were often starkly different from the reference class.



(a) The full generation process for a given style w and set of reference images R using SetGAN.

(b) The "template image" generated by the same w with no conditioning layers.

Figure 6: Diagrams of an example generation process from SetGAN.

5.2.2 Test images with different factors of variation from the training classes

In addition to evaluating the models' generations given images from the same unseen class, we can also examine how the models perform given images from *different* classes. Rather than grouping reference images by their original class in the dataset (i.e. the type of animal, type of flower, or individual person), we selected three groups of images wherein each image was taken from a different class, but all images shared common traits. In the first example, the images consisted of animals of many different types or breeds, with the shared trait being long upward-pointing ears. The second example contained images of flowers from many different types, all of which contained clusters of multiple pink or purple flowers. The third contained images of many different women who were all wearing bold, dark eyeshadow. The resulting generations are shown in Figure 3.

In all cases, SetGAN accurately reproduced the target features while generating a diverse range of output images. Other baselines which conditioned on a single image (i.e. AGE and WaveGAN) each struggled with this - again generating output images either identical to the inputs or very similar with subtle variations. These subtle variations would sometimes lead to deviations from the target features, as the models did not have multiple images to compare to in order to identify which features were shared. For example, the generations from AGE led to some images with short ears, single flowers, or less distinctive makeup. The FSDM baseline *was* also capable of incorporating features from multiple images, but the results were often of lower quality and were less faithful to the target features than those of SetGAN.

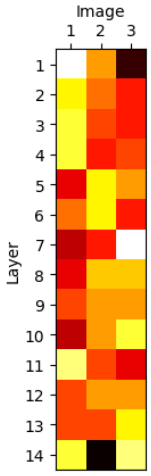


Figure 7: Heatmap of attention weights by layer for Fig. 6a

6 Analysis

In order to visualize how SetGAN constructs an output image from a given set of inputs, consider the example generation shown in Fig. 6a. As explained in section 3.3.1, the generation process begins by encoding each of these reference images into a latent representation C_i using the pSp encoder. The model will then generate a series of m Gaussian noise vectors (one per output image) and pass these through the pretrained StyleGAN2 mapping network to obtain base latent codes $W \in \mathcal{W}$. If these latent codes were fed directly to the generator, they would result in samples from the pretrained StyleGAN2 model, without

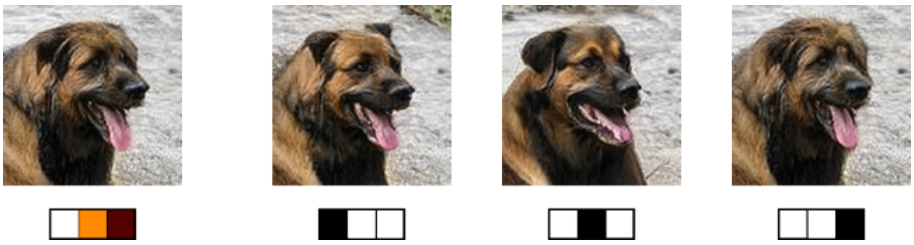


Figure 8: Sample generations using the reference images in Figure 6 with only the first conditioning layer active. Heatmaps underneath each image indicate the attention weights given to each reference image.

any conditioning. We can consider these unconditional generations to be "template" images, which will then be transformed and modified to become the final output (see Fig. 6b).

In order to incorporate the information from the reference images, a series of attention-based conditioning layers will then combine the base latent codes W with the reference encodings c_i^ℓ at each layer of the network to produce a series of *conditional* style vectors ω. This will have the effect of progressively shifting the template image towards the reference images as it progresses through the network.

6.1 Effect of the conditioning network by reference image

Figure 7 shows the relative weight given by the attention blocks in the conditioning network to each of the reference images from the generation process in Figure 6a. In order to visualize how the different weight for each image affects the generation outputs, let us consider the effects of the conditioning network on just a single style. With the conditioning network active on only the first style vector, Figure 8 shows examples of the output with varying degrees of weight given to each reference image - including examples with the true weights taken from the heatmap in Figure 7, as well as 100% weight given to each reference image in turn.

The effect of the varying attention weights at this layer on the final image can be clearly seen from these examples. Features such as the ear shape, ear orientation, fur texture and tongue/mouth position change significantly in accordance with the reference image being most closely attended to at this layer. The effect can be clearly seen on those same features in the final output image. The ears take on a slightly rounded shape, the fur texture becomes shaggy and long, and the open mouth takes on a slight upward lilt that looks almost like a smile - all features strongly similar to the third reference image. This matches the values shown in Figure 7, where the weights are indeed highly concentrated around that same image.

Figure 9: Generations from Fig. 6 with some attention layers inactive. Red boxes indicate active layers.

6.2 Effect of the conditioning network by layer

These previous examples highlighted the effects of the attention layers in attending to and incorporating features from the reference images - but only using a single layer. To see the cumulative effects of these conditioning layers throughout the generation process, we apply the generation process with a variable number of conditioning layers active. As before, inactive layers use only the base style vector as input. The results of this experiment are shown in Figure 9. Initially, no conditioning layers are active, and the generator produces the template image

mentioned previously. As more layers are introduced, additional features from the reference image are used to adjust this template image further and further towards the images in the reference set.

Interestingly, the features affected by the introduction of the conditioning layers vary strongly by the position of the layer in the network. Enabling the conditioning layers in the early layers affects coarse features such as fur texture, stripes/patches, head facing and ear position. In contrast, the middle conditioning layers affect the background, fur color, and finer adjustments to face structure/expression. Finally, the last layers in the network affect subtler qualities like color saturation and fine textural details.

This matches closely with the common observation that the layers of the StyleGAN2 network affect the properties of the output image based on their location in the network, with earlier layers affecting coarser features of the image and later layers affecting the finer details. As our decoder is directly based on the StyleGAN2 decoder, it is unsurprising to observe the same property here.

6.3 Effect of the base style vector

One interesting consequence of the many residual or skip connections through SetGAN's architecture is the predominant role played by the base style vector in the generation. As discussed previously, this base style vector represents a sort of "template image", that will then be modified by each of the conditioning layers in turn to attend to the features of the reference images. Despite the significant effects of these layers shown in the previous sections, the initial template image retains a strong effect on the final generation. Figure 10 shows a series of generations using the same base style vector as in Fig. 6, but different reference images. Notice how all of these images retain similar features in terms of their orientation, head position and overall expression.

Figure 10: Generations from different reference batches using the same base style.

To understand the reason for this, consider Equation 6.3, which shows how the conditional encodings are incorporated into the styles:

$$\omega^{\ell} = g^{\ell}(W, T^{\ell}(W, C))$$

In this equation, g^{ℓ} represents a learned transform applied to the concatenation of the base style vector with the conditional style computed by the appropriate attention block. At the beginning of training, g^{ℓ} is initialized to act as an identity map on the base style, making this essentially a residual connection. As such, the computed conditional encoding will act as an offset *relative* to the base style - anchoring the output generation strongly to the template image.

7 Conclusion and Future Work

The task-specific experiments shown in this paper demonstrate that SetGAN can effectively replicate and even surpass the ability of other GAN-based approaches to learn the factors of variation within different classes in a dataset and generalize them to new classes at inference time. In addition, SetGAN shows potential to generalize beyond the structure of the training classes and flexibly perform generation conditioned on reference images sharing features across a wide array of different axes of similarity. We hope that in the future, this may be extended to more truly general, zero-shot forms of image generation on larger and more diverse datasets. Other approaches such as Giannone et al. (2022) have shown results on datasets such as CIFAR-100 and Mini-ImageNet, but these datasets are low-resolution and contain a very limited number of classes, which limits the model's ability to generalize to truly diverse and varied unseen classes at inference time. While Giannone et al. (2022) do report some succesful results at few-shot generation with these datasets, they often struggle to adapt to unseen classes at inference time as a result of this, and end up producing samples from unrelated training classes. Instead, our focus is on scaling our approach to truly

diverse and large-scale high resolution datasets such as ImageNet. This may provide a path to achieving truly zero shot set-based image-to-image generation, and will be the focus of future work.

Broader Impact Statement

Image generation models can often be ethically fraught. Conditional text-to-image generative models have been the focus of significant uproar, both from those in the AI community and outside of it. The use of large-scale online image datasets has incited controversy due to intellectual property concerns and the alleged role these models play in disenfranchising artists. There are also ongoing concerns about the risk of generative models enabling the spread of disinformation, fake news and propaganda due to the difficulty in distinguishing AI-generated content from that which is human-generated. Image generation models can also be used to create so-called "deepfakes", and may be used to generate misleading or obscene content featuring the likenesses of real individuals.

All experiments performed in this paper are of limited scope and are unlikely to lead to major ethical challenges in the manner of their use. These experiments also do not leverage the large-scale online image data that have elicited accusations of intellectual property theft. That being said, given that SetGAN has the potential to be scaled to a model with much broader and more general scope, it will become very important to be mindful of these concerns as we move forward.

References

Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement, 2021.

Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. 11 2017.

Ching-Yuan Bai, Hsuan-Tien Lin, Colin Raffel, and Wendy Chi wen Kan. On training sample memorization: Lessons from benchmarking generative modeling with a large-scale competition. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. ACM, aug 2021. doi: 10.1145/3447548.3467198.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiao, and Aditya Ramesh. Improving image generation with better captions. 2023.

Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age, 2018.

Louis Clouâtre and Marc Demers. Figr: Few-shot image generation with reptile, 2019.

Guanqi Ding, Xinzhe Han, Shuhui Wang, Shuzhe Wu, Xin Jin, Dandan Tu, and Qingming Huang. Attribute group editing for reliable few-shot image generation, 2022.

Alessandro Ferrero, Shireen Elhabian, and Ross Whitaker. Setgan: Improving the stability and diversity of generative models through a permutation invariant architecture, 2022.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks, 2017.

Giorgio Giannone, Didrik Nielsen, and Ole Winther. Few-shot diffusion models, 2022.

Zheng Gu, Wenbin Li, Jing Huo, Lei Wang, and Yang Gao. Lofgan: Fusing local representations for few-shot image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8463–8471, October 2021.

Ishaan Gulrajani, Colin Raffel, and Luke Metz. Towards gan benchmarks which require generalization, 2020.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

Yan Hong, Li Niu, Jianfu Zhang, Jing Liang, and Liqing Zhang. Deltagan: Towards diverse few-shot image generation with sample-specific delta. *CoRR*, abs/2009.08753, 2020a. URL `https://arxiv.org/abs/2009.08753`.

Yan Hong, Li Niu, Jianfu Zhang, Weijie Zhao, Chen Fu, and Liqing Zhang. F2GAN: fusing-and-filling GAN for few-shot image generation. *CoRR*, abs/2008.01999, 2020b. URL `https://arxiv.org/abs/2008.01999`.

Marco Jiralerspong, Avishek Joey Bose, Ian Gemp, Chongli Qin, Yoram Bachrach, and Gauthier Gidel. Feature likelihood score: Evaluating generalization of generative models using samples, 2023.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan, 2020.

Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The role of imagenet classes in fréchet inception distance, 2023.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pp. 3744–3753. PMLR, 2019.

Yijun Li, Richard Zhang, Jingwan Lu, and Eli Shechtman. Few-shot image generation with elastic weight consolidation. *CoRR*, abs/2012.02780, 2020. URL https://arxiv.org/abs/2012.02780.

Weixin Liang, Zixuan Liu, and Can Liu. Dawson: A domain adaptive few shot generation framework, 2020.

Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation, 2019a.

Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsueprvised image-to-image translation. In *arxiv*, 2019b.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729, 2008. doi: 10.1109/ICVGIP.2008.47.

Utkarsh Ojha, Yijun Li, Jingwan Lu, Alexei A. Efros, Yong Jae Lee, Eli Shechtman, and Richard Zhang. Few-shot image generation via cross-domain correspondence. *CoRR*, abs/2104.06820, 2021. URL https://arxiv.org/abs/2104.06820.

OpenAI. Gpt-4 technical report, 2023.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *CVPR*, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

Harsh Rangwani, Lavish Bansal, Kartik Sharma, Tejan Karmali, Varun Jampani, and R. Venkatesh Babu. Noisytwins: Class-consistent and diverse image generation through stylegans, 2023.

Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.

Chitwan Saharia, William Chan, Huiwen Chang, Chris A. Lee, Jonathan Ho, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models, 2022.

Hiroshi Sasaki, Chris G. Willcocks, and Toby P. Breckon. Unit-ddpm: Unpaired image translation with denoising diffusion probabilistic models, 2021.

Kira A. Selby, Ahmad Rashid, Ivan Kobyzev, Mehdi Rezagholizadeh, and Pascal Poupart. Learning functions on multiple sets using multi-set transformers. In *The 38th Conference on Uncertainty in Artificial Intelligence*, 2022. URL https://openreview.net/forum?id=HzMEEOUs5x5.

Michael Soloveitchik, Tzvi Diskin, Efrat Morin, and Ami Wiesel. Conditional frechet inception distance, 2022.

Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Mengping Yang, Zhe Wang, Ziqiu Chi, and Wenyi Feng. Wavegan: Frequency-aware gan for high-fidelity few-shot image generation, 2022.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

A Appendix

A.1 Architecture and training details

For our experiments, we used pretrained StyleGAN2 (Karras et al., 2020) and pSp (Richardson et al., 2021) models provided by Ding et al. (2022) [3]. We take the layers of these models corresponding to resolutions of 256x256 and lower (i.e., the first 14 layers) to use for our initialization. We use a standard StyleGAN2 generator as our decoder, with 14 layers and a latent dimension of 512. The mapping network uses 8 feed-forward layers, with a learning rate multiplier of 0.01. We use a standard pixel2style2pixel (pSp) encoder architecture, truncated to 14 style vectors. Conditioning networks in the generator used stacks of 2 transformer decoder blocks (see Fig. 1 in the main paper), with 16 attention heads and latent size 512. The discriminator used the standard StyleGAN2 discriminator architecture as its encoder, with the last layer removed. The multi-set transformer model in the discriminator used a stack of 4 multi-set attention blocks, with mean pooling, skip connections surrounding the multi-set transformer network, and a linear output projection. Training was performed using the StyleGAN2 training schema, with a nonsaturating loss, R1 gradient penalty ($\lambda = 10$), path length regularization ($\lambda = 2$) and style mixing ($p = 0.9$). We use lazy regularization, with the R1 penalty applied every 16 steps, and path length regularization applied every 4 steps. We use an exponential moving average of the generator weights, with $\gamma = 0.999$. Training, validation and test splits for each dataset followed the standard splits in prior work, except as discussed previously (see Ding et al. (2022); Hong et al. (2020a); Gu et al. (2021); Hong et al. (2020b)).

Experiments were performed using NVIDIA A40 GPUs. Each model was trained for approximately 1 week using 2 GPUs.

A.2 Latent space truncation

SetGAN uses latent space truncation for inference, in a similar manner to StyleGAN2. In order to improve the quality of the generated results, style vectors are shifted towards the mapping network's mean style vector $\bar{w}$ by a given factor λ. Unlike StyleGAN2, however, this truncation may be applied to SetGAN in two ways: either pre-conditioning or post-conditioning.

Given a base style vector w, pre-conditioning truncation is applied in the same manner as it is for StyleGAN: the latent vector is transformed by the procedure:

$$w \rightarrow \bar{w} + \lambda_1(w - \bar{w}) \tag{9}$$

This ensures that the base style vector used to generate the output images remains in the well-explored region near the mean, and leads to generations of higher quality but slightly lower diversity.

In addition to this, however, truncation may also be applied post-conditioning, to shift the final conditional styles w' towards the mean style vector as follows:

$$w'_j \rightarrow \bar{w} + \lambda_2(w'_j - \bar{w}) \tag{10}$$

[3] https://github.com/unibester/AGE

This has a large effect on output quality, but at a much greater cost to output diversity.

Both λ_1 and λ_2 truncation provided significant benefits on the Flowers dataset, improving sample quality and MiFID score by considerable amounts. λ_1 truncation improved sample quality and MiFID score for the Animal Faces dataset, but this was not used due to the tradeoff in sample diversity. Truncation provided little benefit on the VGGFace dataset. Results in this paper were obtained with $\lambda_1 = \lambda_2 = 0.8$ for the Flowers dataset and $\lambda_1 = \lambda_2 = 1$ for the other two datasets.

A.3 Inconsistencies in Prior Results

Many existing few-shot image generation models contain significant inconsistencies in the methodologies used for evaluation. For example, the LPIPS metric can be evaluated using either AlexNet or VGG activations, which cannot be compared directly against each other. We found that previous works such as F2GAN and DeltaGAN used AlexNet activations to measure LPIPS score, while WaveGAN and LoFGAN used VGG activations. These previous works also generated results at a variety of different resolutions - and some were then rescaled before applying the metric, while others were not. AGE generated outputs at 256x256, while other works performed their generations at 128x128. WaveGAN, LoFGAN and AGE also rescaled images to 32x32 before computing LPIPS distances, while other works did not. Different works also used different code for compiling generated images and rescaling them to the target size of the VGG, AlexNet or Inception models used to obtain vector embeddings. As discussed in Parmar et al. (2022), the details of these steps can have a substantial impact on the final results, and inconsistent methodologies between papers can lead to significant discrepancies. In addition to these inconsistencies in methodology, we found that in many cases we were unable to reproduce the reported scores of existing works - despite using code and checkpoints provided by the authors, and consulting with the authors directly.

A.4 Discussion of additional evaluation metrics

As discussed in section 4.4.2, there were many existing candidates in the literature for evaluation metrics that were sensitive to training/reference set memorization. The four most notable candidates were Conditional FID (Soloveitchik et al., 2022), neural network divergences (Gulrajani et al., 2020), MiFID (Bai et al., 2021) and Feature Likelihood Score Jiralerspong et al. (2023). MiFID was discussed in section 4.4.2. Of the remaining metrics, conditional FID (Soloveitchik et al., 2022) requires an FID calculation to be computed over the reference set, which does not work for cases with small reference sizes due to the instability of the FID calculation with small numbers of samples. Even reference sizes of 10 would only allow for 200-500 samples (depending on dataset) - far too few to perform the FID calculation. Neural network divergences (Gulrajani et al., 2020) are architecture-specific, and must be trained repeatedly for each inference setting. This makes them different to compare across different models and publications, as well as costly to evaluate.

A.4.1 Feature Likelihood Score

Finally, the last important candidate to consider is Feature Likelihood Score (FLS). Feature Likelihood Score uses a method similar to Kernel Density Estimation to fit a Gaussian Mixture density to the generated samples. The covariances of the mixture components are chosen to maximize the likelihood of the reference set, ensuring that the density will be highly concentrated if the samples are simply copied from the reference data. The score is then calculated by evaluating the likelihood of the test data under this density. This scoring method is an interesting candidate, but fails to sufficiently penalize copying - particularly in cases where imperceptible perturbations are applied to the copied image. As shown in Table 3, the FLS scores for the "noisy" synthetic baseline nearly match those of the best trained models across multiple datasets.

A.5 Inference Time

To measure the computational efficiency of each model, we measured the time required for each model to generate a single batch of inputs, with 3 generated images per set and a batch size of 20. As shown in Table 4, all GAN-based models (including SetGAN) are relatively fast to perform inference, with WaveGAN being

	FID_{Inc}	FID_{CLIP}	$\text{MiFID}_{\text{Inc}}$	$\text{MiFID}_{\text{CLIP}}$	FLS_{Inc}	FLS_{CLIP}
Animal Faces						
Best Model	46.20	5.02	46.20	5.02	125.93	133.38
Noisy	24.05	6.94	109.66	14.44	126.72	143.02
Copy	20.44	1.59	17714.97	1335.72	229.81	168.31
True	13.55	1.05	13.56	1.05	114.93	127.70
Flowers						
Best Model	57.12	9.29	57.75	9.29	142.59	144.41
Noisy	37.06	4.21	394.61	22.37	144.18	143.83
Copy	36.98	2.56	23408.14	1737.27	164.80	169.98
True	30.18	1.69	30.18	1.69	139.21	132.39
VGGFace						
Best Model	8.87	2.95	8.87	2.95	134.90	129.20
Noisy	47.96	12.20	56.51	12.64	148.23	145.64
Copy	9.54	0.77	4849.92	438.17	170.63	176.04
True	7.15	0.58	7.15	0.58	134.58	119.17

Table 3: Scores for all metrics (including FLS) on synthetic baselines.

the fastest. FSDM, as a diffusion-based approach, is extremely slow to perform inference - even at only 128 x 128 resolution.

Model	Time
AGE	00:09.79
FSDM	15:42.49
WaveGAN	00:00.42
SetGAN	00:04.64

Table 4: Time to perform a single batch of generations, with batch size 20 and 3 generated images per input set.

A.6 Full Results

Tables 5, 6 and 7 show results on all datasets including standard deviations.

	$\text{MIFID}_{\text{Inc}}$		
	1	3	10
Animal Faces			
AGE	71.35 ± 5.57	62.23 ± 3.33	56.55 ± 0.85
WaveGAN	2327.29 ± 215.18	1057.39 ± 73.21	529.08 ± 14.99
FSDM	75.68 ± 0.93	73.93 ± 1.06	77.37 ± 2.4
SetGAN	61.51 ± 2.58	52.34 ± 0.97	47.18 ± 0.63
Flowers			
AGE	81.87 ± 5.63	70.15 ± 2.46	65.48 ± 1.53
WaveGAN	2653.56 ± 143.46	1305.31 ± 20.58	699.96 ± 28.59
FSDM	69.25 ± 2.37	62.35 ± 0.29	61.47 ± 0.78
SetGAN	62.44 ± 2.27	59.84 ± 0.34	59.31 ± 0.92
VGGFace			
AGE	22.12 ± 1.33	18.39 ± 0.18	16.76 ± 0.3
WaveGAN	852.7 ± 672.02	36.97 ± 0.8	23.12 ± 0.15
FSDM	10.51 ± 0.37	11.26 ± 0.25	12.48 ± 0.13
SetGAN	9.6 ± 0.36	7.93 ± 0.41	7.83 ± 0.03

Table 5: $\text{MIFID}_{\text{Inc}}$ results on all datasets.

	$\text{MIFID}_{\text{CLIP}}$		
	1	3	10
Animal Faces			
AGE	14.09 ± 1.08	12.77 ± 0.8	11.74 ± 0.27
WaveGAN	603.44 ± 46.6	242.81 ± 6.33	136.09 ± 2.29
FSDM	8.78 ± 0.4	8.59 ± 0.38	10.38 ± 0.84
SetGAN	6.56 ± 0.38	5.84 ± 0.21	5.28 ± 0.08
Flowers			
AGE	16.82 ± 0.84	15.03 ± 0.51	14.31 ± 0.2
WaveGAN	851.14 ± 30.14	373.62 ± 21.5	182.11 ± 5.87
FSDM	10.69 ± 0.3	10.26 ± 0.27	10.18 ± 0.03
SetGAN	10.68 ± 0.94	9.79 ± 0.41	9.88 ± 0.21
VGGFace			
AGE	8.2 ± 0.32	6.51 ± 0.22	5.94 ± 0.15
WaveGAN	17.5 ± 0.76	9.4 ± 0.28	6.65 ± 0.12
FSDM	3.28 ± 0.13	3.47 ± 0.08	3.76 ± 0.07
SetGAN	4.16 ± 0.23	3.12 ± 0.14	2.82 ± 0.06

Table 6: $\text{MIFID}_{\text{CLIP}}$ results on all datasets.

	LPIPS		
	1	3	10
Animal Faces			
AGE	0.4027 ± 0.0026	0.5095 ± 0.005	0.5504 ± 0.0023
WaveGAN	0.0 ± 0.0	0.4211 ± 0.0037	0.5556 ± 0.0022
FSDM	0.6039 ± 0.0006	0.6076 ± 0.0011	0.6086 ± 0.0006
SetGAN	0.6144 ± 0.0019	0.6154 ± 0.0007	0.6181 ± 0.0007
Flowers			
AGE	0.379 ± 0.0114	0.5528 ± 0.0044	0.6078 ± 0.0037
WaveGAN	0.0 ± 0.0	0.4844 ± 0.0044	0.6345 ± 0.0031
FSDM	0.6809 ± 0.0038	0.6985 ± 0.0026	0.7042 ± 0.001
SetGAN	0.6166 ± 0.0032	0.624 ± 0.0037	0.6281 ± 0.0019
VGGFace			
AGE	0.2604 ± 0.0005	0.3693 ± 0.0043	0.4063 ± 0.0028
WaveGAN	0.0 ± 0.0	0.3246 ± 0.0029	0.4301 ± 0.0036
FSDM	0.4509 ± 0.0034	0.4477 ± 0.0014	0.4471 ± 0.0006
SetGAN	0.4633 ± 0.0049	0.4614 ± 0.0038	0.4712 ± 0.0008

Table 7: LPIPS results on all datasets.