Faster Image Synthesis: Improving Performance for GANs

Humza Qavi
UT Austin
humzaqavi@utexas.edu

Huy Tran
huydtran@utexas.edu

Abstract

Generative Adversarial Networks(GANs) take a lot of time and data to train, making them expensive to train when resources are limited. FastGAN is a few-shot GAN that takes advantage of optimizations such as Skip Layer Excitations(SLE) and a self-supervised discriminator. We feel that SLE can be further improved with modifications and combinations of different attention mechanisms. This paper will cover the results of the various attention mechanisms we've experimented with. We were able to achieve minor but noticeable improvements over FastGAN, using a combination of Efficient Channel Attention(ECA) and SLE. Although the sample size is relatively small, we believe this direction is promising and has untapped potential.

1. Introduction

Generative Adversarial Networks (GANs)[2] are currently a hot and rapidly developing subject in modern computer vision. GANs are neural networks that aim to generate images that are indistinguishable from a provided image distribution. The creation of synthetic images has various artistic, recreational, and photography applications. Aside from image generation, GANs can also perform tasks like style transfer and photo blending.

However, one major problem with using GANs is that they tend to be expensive computationally. StyleGAN2[6] is widely considered to be the leading GAN. It was trained on the FFHQ[5] dataset for the task of human face generation. Although the comprehensive dataset leads to high quality image generation, it is 70,000 images and it is highly unfeasible to collect this amount of quality data for most other domains. Not only is it hard to collect data, but GANs requires a large amount of VRAM and time to train. These characteristics make modern GANs difficult to train on the average computer. This effectively limits the amount of devices that StyleGAN2 can run on, which notably doesn't include mobile and IOT devices. Having a performant model with a smaller memory footprint is much more practical.

One approach to this problem is the Few-Shot GAN[9], a GAN that tries to achieve similar results with significantly smaller datasets(< 1000 samples). This type of GAN is more practical for most applications such as medical images, images of a specific person, or artwork from a specific artist. These are all use cases where there may not be a sufficient amount of data to train a regular image synthesis GAN like StyleGAN2[6]. However, using a limited dataset comes with the inherent risks of the model being more susceptible to overfitting and mode collapse than regular GANs.

Regardless, some existing works try to make Few-Shot GANs more practical. For example, FastGAN[7] improves training time and image synthesis results by implementing Skip Layer Excitations(SLE) and a self-supervised discriminator. There is also Small-GAN[11] which tries to speed up GAN training in general using methods like core set selection.

This paper will be focusing on modifying the FastGAN implementation, specifically the SLE block. The SLE block is a type of Squeeze-and-Excitation(SE) block[4] with the addition of a multi-resolution skip connection. A basic SE Block aims to model inter-dependencies between channels, which can be used to learn global information that the network can use for feature re-calibration. We tried various modifications to the SLE block incorporating ideas from other attention mechanisms like Efficient Channel Attention(ECA)[12] or Convolutional Block Attention Module(CBAM)[14].

Ultimately, our best iteration of the SLE block was a combination of ECA and SLE. By leveraging the lightweight parameters of ECA and the multi-resolution skip connection of SLE, we efficiently learn channel weights of smaller feature maps while applying them to the corresponding large feature map. We believe our results show that there is still room for improvements for few-shot GANs and warrants further investigation.

2. Related Works

2.1. FastGAN

FastGAN[7] aims to improve the training times and performance of GANs by implementing a Few-Shot GAN with Skip Layer Excitations(SLE) and a self supervised discriminator. SLE is a type of skip-connection that applies channel-wise multiplications between activations. The FastGAN paper compares results to StyleGAN2 using a 100 image dataset. We want to compare results to other other GANs because StyleGAN2[6] was not designed to be trained with a few-shot dataset. FastGAN's performance still has room for improvement with varied results such as the shell dataset appearing warped, while panda dataset having little to no artifacts. The official implementation is public on GitHub. We plan to modify the architecture of the existing model's skip connections.

2.2. Few-Shot GAN

Gathering large datasets is expensive and impractical in many cases. One solution to few-shot learning is transferring knowledge from a pretrained model. The Few-Shot GAN[9] implements this by adapting the pretrained weights to the new parameter space. Although somewhat effective, this approach is still unfeasible when training time or memory footprint is a concern so our model won't be using pretrained weights and will be trained from scratch.

2.3. SENet

Squeeze and Excitation blocks[4] are a way for a network to perform feature recalibration. To achieve this, SE blocks first take in the input image (H x W x C) and perform a squeeze operation, using average pooling to create a channel-wise vector of (1 x 1 x C). This vector represents an embedding of the global distribution of channel-wise feature responses. This vector is then used to produce per channel modulation weights, this process is known as the excitation step. These weights are then applied to the original feature maps to produce the output of the SE blocks. Overall, these SE blocks are versatile and can perform different roles in a network at different depths.

The SLE blocks used in FastGAN are themselves a variation of SE blocks which interact with more distant feature maps through multi-resolution skip connections, which allow for the channel-wise vector to be generated from feature maps from different parts of the network with different resolutions. We intend to compare the results of using SLE with the original SE blocks.

2.4. Efficient Channel Attention

ECA modules[12] are an alternative to SE blocks. Compared to SE/SLE blocks, ECA modules don't use Fully Connected layers to capture cross-channel interaction which has the advantage of not involving dimensionality reduction. Instead, ECA modules implement local cross-channel interaction by doing a convolution along the channels after using global average pooling. This greatly improves performance because it leads to substantially lower model complexity compared to SE which has to use fully connected layers instead of just a 1-d convolution.

2.5. Convolutional Block Attention Module

CBAM[14] is also an alternative to SE blocks. The main idea behind CBAM is to learns spatial attention as well as channel attention. Channel attention is learned by creating a feature vector from average-pooling and max-pooling and passing it through a MLP with a single hidden layer, which then produces a vector of channel weights. Similarly, spatial attention is learned by doing average-pooling and max-pooling along the channel axis and passing the results into a convolutional layer. These two types of attention can be applied in sequence or in parallel, although it is suggested that applying channel attention followed by spatial attention leads to the best results.

2.6. Small GAN

Another way to improve GAN training is through coreset selection[11]. GANs generally benefit from larger mini-batch sizes, but this normally comes with a substantial computational cost. To prevent this, coreset selection can be used to sub-sample a smaller "coreset" of samples from a larger batch that tries to cover the same modes as the larger selection. This can reduce both training time and memory usage for GANs although it's likely that it won't be directly applicable to few-shot GANs since it requires sampling large batches from the data set.

2.7. MobileStyleGAN

MobileStyleGAN[1] is another approach to reduce the computational complexity of generative models, but instead of focusing on the attention mechanism it uses a wavelet based CNN architecture. In more detail, instead of predicting a pixel based output like StyleGAN or FastGAN, MobileStyleGAN tries to predict the discrete wavelet transform of the output. This leads to less parameters overall, since wavelet based representations contain more structural information than pixel based representations so it becomes possible to generate high resolution output using only low resolution feature maps.

However although MobileStyleGAN does produce a lightweight model for evaluation, MobileStyleGAN is trained using knowledge distillation from a StyleGAN2 network which means that unlike FastGAN the training speed for MobileStyleGAN is still relatively slow. In addition, MobileStyleGAN is not designed for few-shot learning and is unlikely to perform well on small datasets.

3. Methodology

3.1. Alternative Blocks

We chose to modify the SLE block because it had impressive results in the FastGAN [7] paper in terms of performance and efficiency. After analyzing the structure of the SLE block , we decided to combine it with various modifications of the ECA[12] and CBAM[14] blocks while keeping its multi-resolution skip connections. In our blocks, we worked with 2 inputs, a small and larger feature map. In the original SLE block, the spatial dimension of the small feature map was aggregated into a single 1d vector representing channel dimensions. This vector shared the same amount of channels as the larger feature map, allowing the learned weights from the small feature map to be applied despite different spatial dimensions. We tried attaching different attention mechanisms in 3 main ways: to process the smaller feature map before passing it into the SLE Block, to process the results of the SLE block before multiplying by the larger feature map, and to process the larger feature map before multiplying with the results of the SLE block.

3.2. Considering ECA

Unlike other blocks that scale with image size, ECA keeps a minimal number of parameters, only learning a limited number of 1D kernels for cross-channel weights[12]. As a control, we first tried a pure ECA approach, where we ignored features of the SLE block and only applied ECA to the larger feature map. This is in line with the standard ECA-Net approach. This showed us some effects from from SLE such as strengthening gradient signals between layers are vital to the performance of the model. Our best results came from applying ECA to the smaller feature map, before the SLE block. We also tried applying ECA to the smaller feature map generated by SLE, as well applying ECA to the larger feature map. Both of these approaches gave us worse results. We speculate this is because the larger feature maps have less channels.

3.3. Considering CBAM

CBAM takes advantage of both channel and spatial attention[14]. We used CBAM in hopes of improving results due to the addition of the spatial module. However, we found that the frequent 2D convolutions in both channel and spatial modules led to a relatively large increase in parameters since the skip connection is used repeatedly in the model between each up-sample. Our best implementation of CBAM only used the channel block without any spatial attention. We tried implementing spatial attention as part of a standard CBAM as well as a mixture of ECA and CBAM using the ECA for channel attention and CBAM for spatial attention. However, all results of blocks using spatial attention modules suffered from extreme mode collapse.

Figure 1. Noticeable mode collapse occurs when using CBAM or spatial attention

3.4. Efficient Skip Layer Excitation Block

The block that gave us the best results, was a combination of ECA and SLE. The Efficient Skip Layer Excitation (ESLE) block is implemented as an ECA block that processes the smaller feature map before passing it into an SLE block. The result of which is then multiplied by the larger feature map. The use of the ECA block allows the model to better learn channel weights for the smaller feature map with a minimal amount of added parameters.

This approach used in tandem with the SLE block allows us to make use of ECA's unique advantages such as capturing attention without dimensionality reduction and adaptive cross-channel interactions. Adding the ECA block at the beginning seemed to yield better results, likely because of increased channel size for the smaller feature map. Another benefit of increased channel size is an increased amount of cross-channel interaction. Although many ECA implementations use a fixed kernel size of 3, the kernel size in our ECA block increases with channel size, causing the amount of cross-channel interaction to scale with growing channel size.

$$k = \psi(C) = |\frac{\log_2(C)}{\gamma} + \frac{b}{\gamma}|_{odd} \quad (1)$$

Figure 2. Formula to adaptively calculate the kernel size based on kernel size from ECA[12]. γ and b are arbitrary tuning parameters we set to 2 and 1 respectively.

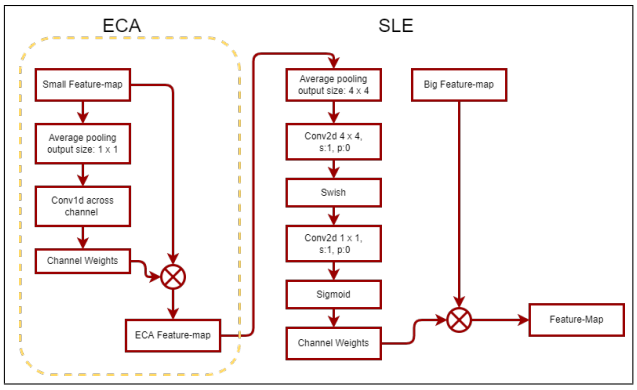

Figure 3. Data Flow of Efficient Skip Layer Excitation Block (ESLE), used several times throughout both the generator and discriminator models

4. Datasets

From the collection of datasets used in the FastGAN paper, we considered several of the datasets to train on including dogs, cats, and anime. After training on these datasets, some of the images were noticeably low quality. Because of the small size of the datasets, we were able to look through each picture in each dataset. A substantial amount of the animal images were low resolution, had occluded faces, or were nearly visually identical. Because the model is a few-shot GAN, bad data in a small dataset has more prominent effects on the generated images. Since this was heavily present in the dog dataset, we decided not to use it. The majority of the images in the anime dataset seemed to be of high quality, but a few images were completely distorted. After further investigation, the images in this dataset were found to be GAN generated and had noticeable artifacts. We also decided against using this dataset. We ended up using more standard few-shot image datasets such as Oxford 102 Flowers and Caltech-UCSD Birds 200.

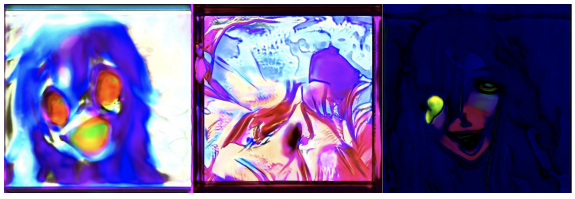

Figure 4. Examples of poor quality images found in the anime dataset. Images suggest that dataset was GAN generated.

4.1. Cats

The Cats dataset originates from the LHI Hybrid Image Template Animal Faces Dataset [10]. 160 256x256 images of cats were sampled from this dataset with a variety of breeds, poses, and lighting conditions. Something to note is that several images in the dataset are of either the same cat, or very similar cats which has a noticeable impact on the model's output distribution.

Figure 5. Multiple similar images found in the dataset, potentially of the same or similar cats

4.2. Flowers

The other main dataset we use, is the Oxford-102[8] dataset which contains 102 categories of flowers commonly found across the United Kingdom. The few-shot version of this dataset consists of 102 1024x1024 images randomly sampled across each category. By design, these images have substantial scale, pose, and light variations as can be seen in the isomap below.

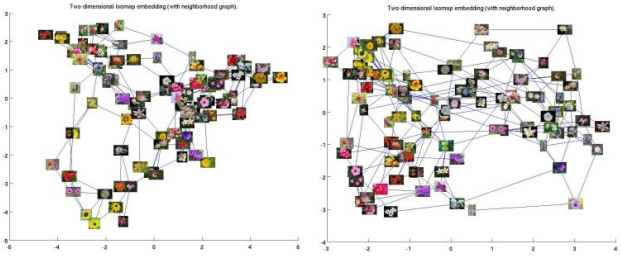

Figure 6. A shape isomap (left) and a color isomap (right) showing the distribution of images in the Oxford-102 dataset.[8]

4.3. Birds

The Caltech-UCSD Birds 200 (CUB-200)[13] dataset consists of 200 categories of North American bird species, with each image having a resolution of 1024x1024. To create the few-shot version of this dataset, we sampled an image from each of these 200 categories to create a 200 image dataset.

5. Experiment

5.1. Evaluation Metrics

The two primary metrics we use to evaluate our model's performance are Frechet Inception Distance (FID)[3].

5.1.1 FID

Frechet Inception Distance (FID)[3] is a distance metric that evaluates the quality of generated images by finding the distance between inception feature vectors for real and generated images. A lower FID tends to correlate with better performance. To measure FID we use the standard practice of generating 50,000 images to use as the test distribution, which using the entire training set as the reference distribution.

5.1.2 Parameter Count

Another useful metric to compare our model variations with is parameter count. Since the number of parameters correlates with model complexity and puts limitations on the hardware that is able to run the model, parameter count is important to keep track of.

5.2. Experiment Setup

To evaluate these various attention mechanisms, we start with an unmodified instance of FastGAN[7]. We then replace the default SLE block with a modified version that incorporates the attention mechanism we intend to test. Using an Adam optimizer with a learning rate of 0.002 and a batch size of 5, we then train the model for 50,000 iterations on the chosen dataset. We picked the batch size of 5 because it was the largest batch size we could run on the default FastGAN model. The models were trained on a single RTX 2070 Super. Promising models were then further trained for 100,000 iterations on more datasets. We then compare these model variations based on FID.

6. Results

6.1. Parameter Count

Due to hardware limitations, since we were running on a RTX 2070 Super with 8 GB of VRAM, we tried to keep parameter counts minimal. Increasing parameter counts too much, would make the model unable to train without decreasing the batch size, which would greatly decrease the performance of the model.

As can be seen in Table 1, ECA has a minimal effect on parameter counts. Although the CBAM implementations appear to have less parameter counts, this was because we had no implementations combined with SLE. The SLE block has always given good performance. However, combining CBAM and SLE led to too many parameters, causing the model to exceed the 8 GB of available memory. The increase of parameters from purely CBAM-channel to CBAM-channel-spatial was minimal, but there was a significant drop in performance and visible mode collapse. This was also true in ECBAM. For this reason, we decided against using a spatial attention module.

Block	Attention Mechanism	Parameter Count
SLE (default)	$SLE(f_s) * f_l$	29.18M
ECA	$ECA(f_l)$	**26.37M**
ESLE	$SLE(ECA(f_s)) * f_l$	29.18M
SLE-ECA	$ECA(SLE(f_s) * f_l)$	29.18M
SLECA	$ECA(SLE(f_s)) * f_l$	29.18M
CBAM-channel	$CBAM_c(f_l)$	26.47M
CBAM-channel-spatial	$CBAM_s(CBAM_c(f_l))$	26.47M
ECBAM	$CBAM_s(ECA(f_l))$	26.37M

Table 1. # of Trainable Parameters with Different attention mechanisms. The attention mechanism column shows specific data flow for the block input and output. f_s represents the small feature-map which is from 4 layers earlier in the model,while f_l represents the current large feature-map.

6.2. Training Time

One of the original goals in this project was to reduce training time, but even the plain ECA had a negligible effect on training time. The bulk of the training time came from the repeated convolutions and depth of the model, so changing attention mechanisms would not improve training times. All models ran at approximately 7 hours per 50k iterations on a RTX 2070 Super.

6.3. FID

Block	Attention Mechanism	FID (Cats)
SLE (default)	$SLE(f_s) * f_l$	**46.31**
ECA	$ECA(f_l)$	56.03
ESLE	$SLE(ECA(f_s)) * f_l$	48.98
SLE-ECA	$ECA(SLE(f_s) * f_l)$	53.47
SLECA	$ECA(SLE(f_s)) * f_l$	49.74
CBAM-channel	$CBAM_c(f_l)$	65.16
CBAM-channel-spatial	$CBAM_s(CBAM_c(f_l))$	164.30
ECBAM	$CBAM_s(ECA(f_l))$	125.75

Table 2. FIDs on Cat Dataset (Best of 50k Iterations). The attention mechanism column shows specific data flow for the block input and output. f_s represents the small feature-map which is from 4 layers earlier in the model,while f_l represents the current large feature-map.

For efficient use of our time and hardware, we trained for 50k iterations on the Cat dataset for each block. Because a batch size of 5 was less than the default batch size of the FastGAN, we used this data to find the most promising block.

	Cats		Flowers		Birds	
Iter.	SLE	ESLE	SLE	ESLE	SLE	ESLE
10	86.2	67.5	174.0	179.1	209.2	150.4
20	59.2	52.2	111.0	134.0	115.0	132.7
30	62.2	57.5	100.7	83.9	138.6	133.9
40	52.8	53.9	100.7	78.1	137.1	122.9
50	**46.3**	49.0	81.6	78.7	118.8	128.4
60	50.0	43.9	84.7	81.9	130.5	119.0
70	50.4	48.1	75.5	86.4	**104.1**	105.9
80	64.9	**43.7**	83.6	71.2	106.3	112.1
90	77.7	49.5	75.4	**68.9**	121.9	106.4
100	118.1	62.4	**71.8**	75.4	132.0	**94.1**

Table 3. FID on Cats, Flowers, & Birds Datasets (100k Iterations). SLE is the attention mechanism used by the default FastGAN implementation. ESLE is the $SLE(ECA(f_s)) * f_l$ attention mechanism

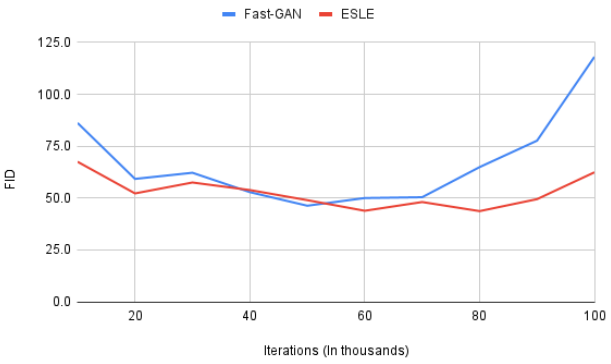

Figure 7. Graph comparing FID on the Cats dataset over 100k iterations between the SLE/FastGAN and the ESLE variations of the model

After training for more iterations, ESLE achieved lower FID scores than the SLE block. SLE has lower scores on some iterations, but ESLE trends lower for the majority of iterations.

6.4. Generated Images

These images were randomly selected from the respective models with the lowest FID.

Figure 8. 8 uncurated images sampled from the best default FastGAN/SLE variation of the model after 100k iterations on the cats dataset

Figure 9. 8 uncurated images sampled from the best ESLE variation of the model after 100k iterations on the cats dataset

Figure 10. 8 uncurated images sampled from the best default FastGAN/SLE variation of the model after 100k iterations on the flowers dataset

Figure 11. 8 uncurated images sampled from the best ESLE variation of the model after 100k iterations on the flowers dataset

Figure 12. 8 uncurated images sampled from the best default FastGAN/SLE variation of the model after 100k iterations on the birds dataset

Figure 13. 8 uncurated images sampled from the best ESLE variation of the model after 100k iterations on the birds dataset

6.5. Qualitative Observations

The images generated by the models using the ESLE block were satisfactory. Although it is hard to definitively say which models were better based on these images, generating large amounts of image often exposed the weaknesses of the SLE block. The SLE block tended to suffer from mode collapse quickly after reaching its best FID score.

There were noticeable appearances of similar cats in both SLE and ESLE generated images. When looking in the dataset, we noticed there were cats that resembled these recurring cats. For both models, cat ears seemed to be the most obvious flaw in the image. Ears were often misaligned with the direction that the cat was facing. There is a noticeable, but minor improvement when looking through both generated datasets. In general, cats were the easiest to train on due to their similar features and shapes regardless of breed.

The consistent high resolution of the images in this dataset produced more quality results. Because of the organic shape of flowers, some images appeared to be amorphous colorful blobs. This shows up in the image on the 3rd column, 1st row in figure 10. On average, these seemed to be more prevalent in SLE image synthesis. Additionally, some images had green noise in the background, likely an unsuccessful attempt to learn leaves and stems.

Despite having a lower FID, ESLE seems to generate worse birds than SLE. Interestingly, both models seemed to strongly prefer certain features in some iterations. For example, one iteration predominantly produced birds with red heads, yet the birds still maintained unique shapes. This likely indicates our evaluation metrics for the GANs were insufficient.

7. Drawbacks

One of the metrics we considered using, but didn't actually manage to compute for all our models was using LPIPS[15] with backtracking to compute a reconstruction score. This would have been able to loosely indicate when mode collapse occurs.[7]

Due to the lengthy training time of the GANs, we were unable to train every model for a full 100k iterations on every dataset. Ideally, each block would be fully evaluated for a sufficient amount of iterations as well as wider variety of dataset. With our limited testing, it is entirely possible that one of the other models would have performed better given more iterations to train. As mentioned before, the cat dataset is flawed with images that could encourage mode collapse. There are many domains our models were not tested on. Although our ESLE block achieved better FID scores than the SLE block for our datasets, 3 is a small sample size. It is not enough to confirm that ESLE outperforms SLE.

It also would have been preferable to train and evaluate StyleGAN2 on these datasets as well. Being unable to compare our models to the premier model in the field makes it difficult to gauge the viability of our model, even if StyleGAN2[6] is not a few-shot GAN. Unfortunately, due to time and computational restraints, we were unable to score StyleGAN2 on these examples.

Similarly, because of limited time, we were also unable to evaluate the model as many times as would have preferred. Ideally, each model would be trained 5-10 times per

dataset and scored as the average performance of all models to guarantee our results would be consistent.

Hardware limitations also restricted our training setup, since we were unable to train with a batch size of 8 as was used in the original FastGAN[7]. We opted to use the largest possible batch size of 5. Using a smaller batch size makes it so that our FIDs don't match up with the original paper.

8. Conclusion

Overall, the ESLE block makes minor but noticeable improvements such as decreasing FID by 3-10 on average and a qualitative decrease in mode collapse when observing the generated data directly from various datasets. Our attention mechanisms have produced promising results that could be revisited for further improvement of the few-shot GAN. More specifically, we would like reevaluate our models with better hardware and more rigorous testing. With the limitations of our current experiments, it is possible that certain block variations could have performed well, but were overlooked. Further things that could be studied include modifying the self-supervised discriminator, other attention mechanisms, and modifying other aspects of the model architecture.

9. Task Assignment

The majority of the work for this project was done together over a VS code live share. The base FastGAN code was taken from the following repository https://github.com/odegeasslbc/FastGAN-pytorch[7]. All our attention mechanisms are in models.py. The code can be found at https://github.com/HuyDTran121/CS395T-Project

References

[1] Sergei Belousov. Mobilestylegan: A lightweight convolutional neural network for high-fidelity image synthesis. *arXiv preprint arXiv: Arxiv-2104.04767*, 2021. 2

[2] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv: Arxiv-1406.2661*, 2014. 1

[3] 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. *arXiv preprint arXiv: Arxiv-1706.08500*, 2017. 4, 5

[4] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks. *arXiv preprint arXiv: Arxiv-1709.01507*, 2017. 1, 2

[5] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv: Arxiv-1812.04948*, 2018. 1

[6] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *arXiv preprint arXiv: Arxiv-1912.04958*, 2019. 1, 2, 7

[7] Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. Towards faster and stabilized gan training for high-fidelity few-shot image synthesis. *arXiv preprint arXiv: Arxiv-2101.04775*, 2021. 1, 2, 3, 5, 7, 8

[8] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008. 4

[9] Esther Robb, Wen-Sheng Chu, Abhishek Kumar, and Jia-Bin Huang. Few-shot adaptation of generative adversarial networks. *arXiv preprint arXiv: Arxiv-2010.11943*, 2020. 1, 2

[10] Zhangzhang Si and Song-Chun Zhu. Learning hybrid image templates (hit) by information projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1354–1367, 2012. 4

[11] Samarth Sinha, Han Zhang, Anirudh Goyal, Yoshua Bengio, Hugo Larochelle, and Augustus Odena. Small-gan: Speeding up gan training using core-sets. *arXiv preprint arXiv: Arxiv-1910.13540*, 2019. 1, 2

[12] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. *arXiv preprint arXiv: Arxiv-1910.03151*, 2019. 1, 2, 3

[13] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 4

[14] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. *arXiv preprint arXiv: Arxiv-1807.06521*, 2018. 1, 2, 3

[15] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *arXiv preprint arXiv: Arxiv-1801.03924*, 2018. 7