Google Research

JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

# Sequential training of GANs against GAN-classifiers reveals correlated "knowledge gaps" present among independently trained GAN instances

Arkanath Pathak, Nicholas Dufour

THU-PM-369

# Overview

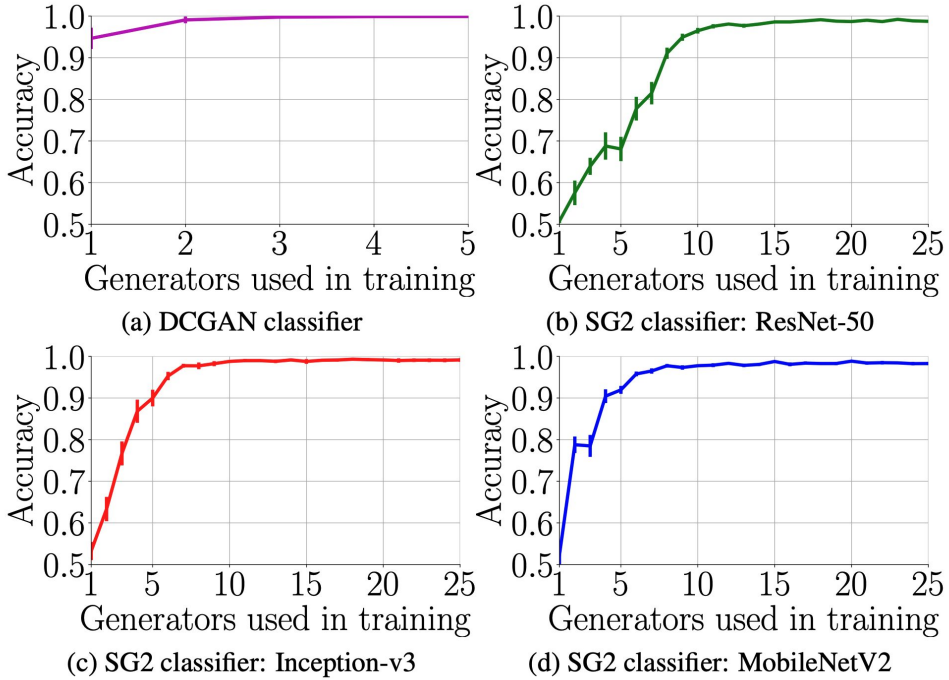
# Setup

$$\mathcal{L}^{\mathbf{\Sigma}}_{\mathbf{G^{(i)}}} = -[\log(D^{(i)}(G^{(i)}(w))) + \phi \sum_{j=0}^{i-1} \log(C_0^{(j)}(G^{(i)}(w)))]$$

- We modify the GAN training loss function to encourage fooling pre-trained GAN-classifier(s), weighted by parameter $\phi$
- Low-parameter setting
  - A small **DCGAN** is trained on handwritten digits (**MNIST**)
  - Vanilla CNN GAN-classifier
- High-parameter setting
  - **StyleGAN2** (SG2) trained on human faces (**FFHQ**)
  - Experiments with three different classifier architectures: **ResNet-50**, **Inception-v3** and **MobileNetV2**

# Classifier generalization

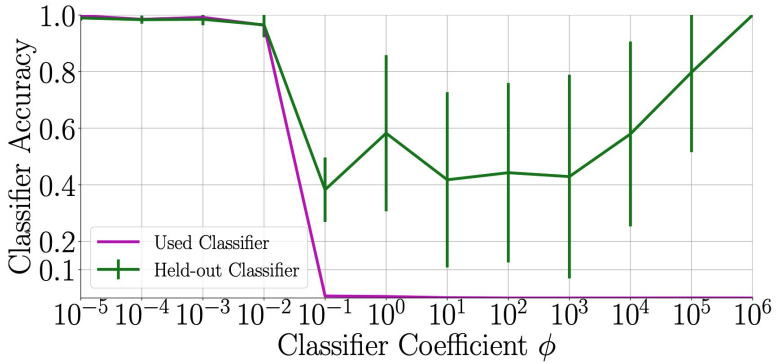
- Classifiers generalize to unseen GANs. Generalization strength depends on the number of generators sampled from during training.
- This effect is weaker in the low parameter setting and stronger in the high parameter setting.



# DCGAN generators unable to fool held-out classifiers

- In the low-parameter setting, the DCGAN generators are unable to fool a held-out GAN-classifier instance, even at high-values of $\phi$.
- As we increase $\phi$ to higher values, the output quality collapses and the variance in the detection accuracy of held-out classifiers increases.

$$\mathcal{L}^*_{\mathbf{G}} = -[\frac{1}{1+\phi}\log(D) + \frac{\phi}{1+\phi}\log(C)]$$



# SG2 generators can fool classifiers, with caveats

- In the high parameter setting, learning to fool one ResNet-50 instance conferred the ability to fool any held-out ResNet-50 classifier instance.
- This effect is slightly diminished in Inception-v3 and highly diminished with MobileNetV2 classifiers.

| | GAN trained to fool... | | |
|---|---|---|---|
| Classifier | ResNet-50 | Inception-v3 | MobileNetV2 |
| ResNet-50 | **0.05±0.02** | 0.74±0.12 | 0.5±0.39 |
| Inception-v3 | 0.51±0.25 | **0.16±0.26** | 0.53±0.34 |
| MobileNetV2 | 0.31±0.17 | 0.36±0.16 | 0.41±0.38 |

# Artifacts are shared within subsequent iterations

- GAN generators trained to fool a classifier do not transition to distinct artifact spaces.
- GAN instances reliably align on the same space within an iteration.
- Generators of subsequent iterations share a new artifact space, not captured by classifiers of previous iterations.
- Output quality is not affected as measured with *FID* or visually.

| | GAN Instances | | | | |
|---|---|---|---|---|---|
| Classifier | Iteration 0 | Iteration 1 | Iteration 2 | Iteration 3 | Iteration 4 |
| Iteration 0 | **0.99±0.00** | 0.03±0.00 | 0.04±0.01 | 0.05±0.01 | 0.05±0.01 |
| Iteration 1 | 0.00±0.00 | **1.00±0.01** | 0.01±0.00 | 0.06±0.00 | 0.01±0.00 |
| Iteration 2 | 0.00±0.00 | 0.73±0.14 | **0.96±0.05** | 0.01±0.00 | 0.03±0.01 |
| Iteration 3 | 0.00±0.00 | 0.87±0.11 | 0.18±0.13 | **0.93±0.07** | 0.02±0.00 |
| Iteration 4 | 0.01±0.00 | 0.71±0.14 | 0.51±0.14 | 0.07±0.03 | **0.86±0.11** |
| Mean FID | 36.98 | 36.62 | 36.67 | 36.39 | 36.83 |

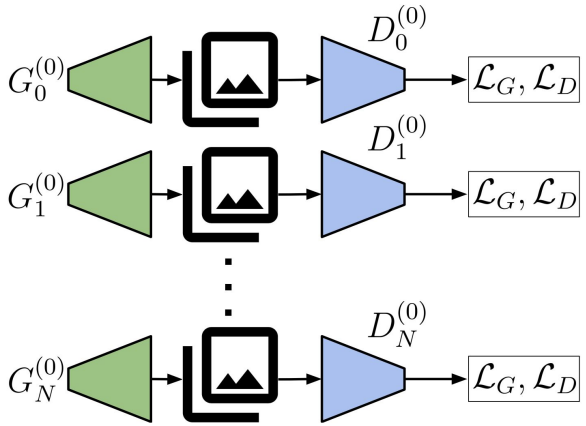
# Deep Dive

# Background

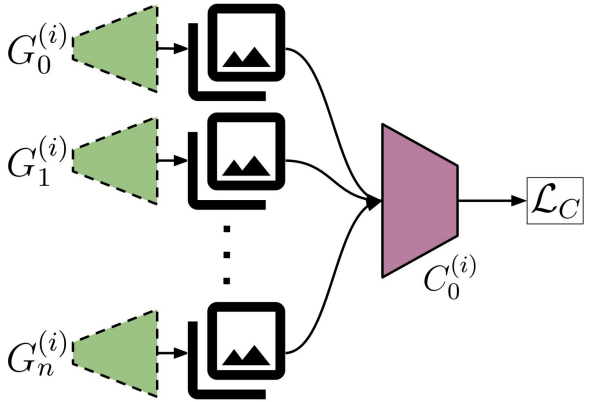
- SOTA GANs generate diverse and realistic images, at scale and of high resolution.
- GAN-generated human faces are widely disseminated and have been used for creating fake identities on the internet.
- CNN **classifiers** are known to be effective at detecting GAN-generated images, therefore there are "knowledge gaps": properties of natural images that are consistently ignored during generator training (guided by co-trained **discriminators**).
- We train GANs against GAN-classifiers to eliminate the knowledge gaps learned by the classifiers, and study the effect on GAN outputs and the space of artifacts.

# Setup: Classifier Training

- A number of GAN generators are initialized randomly but trained on samples from the same dataset.
- GAN-classifiers are trained using the outputs sampled from the most recent cohort generators.
- The training dataset is balanced: we have an equal number of real images and generated samples. The dataset of real images used in training the classifier is the same as was used in training the GANs.
- The training data is also balanced when sampling across the different generators.



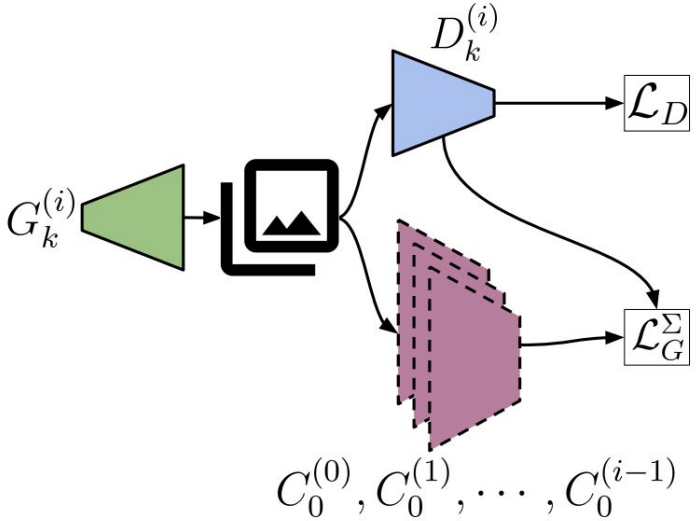

(a) Stage 1 at iteration 0: GAN training with standard loss function




(b) Stage 2 at iteration $i$: Classifier training

# Setup: Sequential GAN training

- The GAN generators in subsequent iterations are trained with a modified loss function with an additional adversarial loss term for the previously-trained classifiers trained in the previous iterations.
- Classifiers are frozen during the GAN training.
- Generators learn to fool both the co-trained discriminators and the already-trained classifiers.
- A coefficient $\phi$ is used to weight the relative influence of the classifiers.




$$\mathcal{L}_{\mathbf{G}^{(\mathbf{i})}}^{\mathbf{\Sigma}} = -[\log(D^{(i)}(G^{(i)}(w)))$$

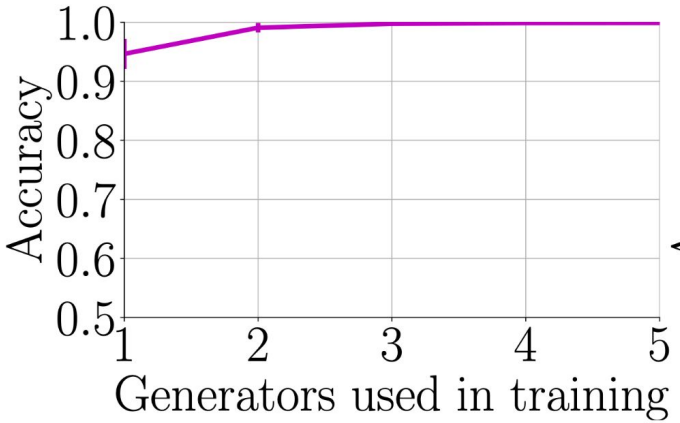
$$+ \phi \sum_{j=0}^{i-1} \log(C_0^{(j)}(G^{(i)}(w)))]$$

# Classifier generalization

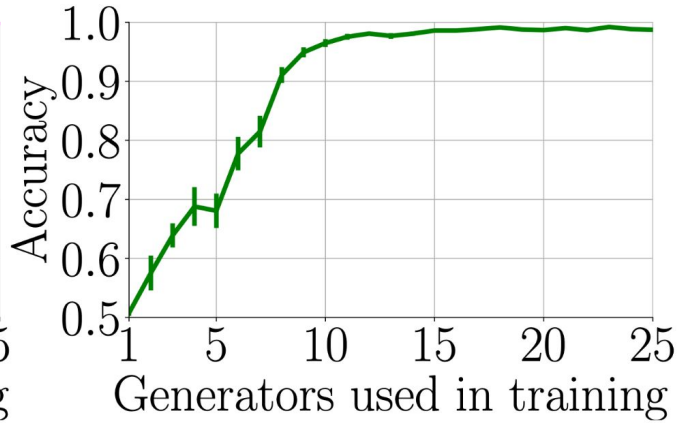
To profile the number of independent generators necessary to train a classifier with generalization capability, we incrementally increase $n$, the number of generators in the pool of generators used in classifier training: $S_0 = \{G_0^{(0)}, ..., G_n^{(0)}\}$

The classifier is evaluated on samples from held-out generators: $S_1 = \{G_{n+1}^{(0)}, ..., G_N^{(0)}\}$

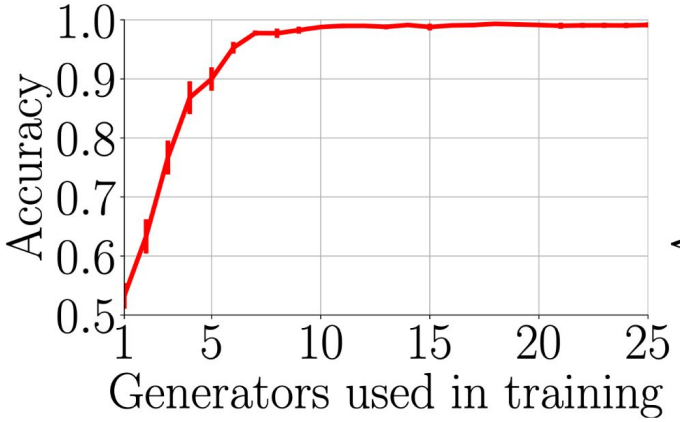
- DCGAN classifier generalizes well using a single generator, and almost perfectly when trained on more than one generator.
- In contrast, SG2 generators produce sufficient diversity between instances that a classifier requires samples from several generators, irrespective of the classifier architecture.



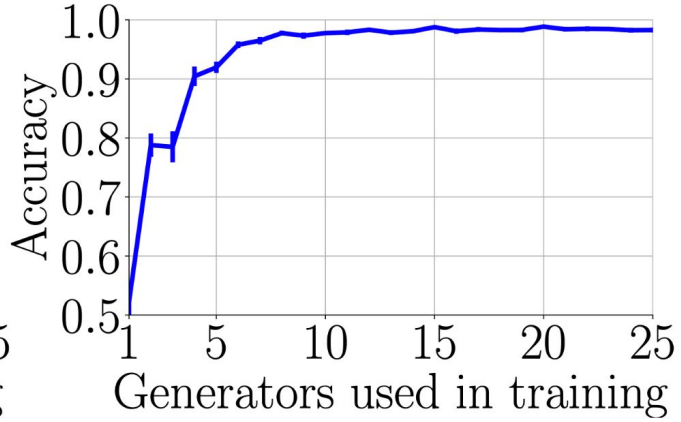

(a) DCGAN classifier




(b) SG2 classifier: ResNet-50




(c) SG2 classifier: Inception-v3



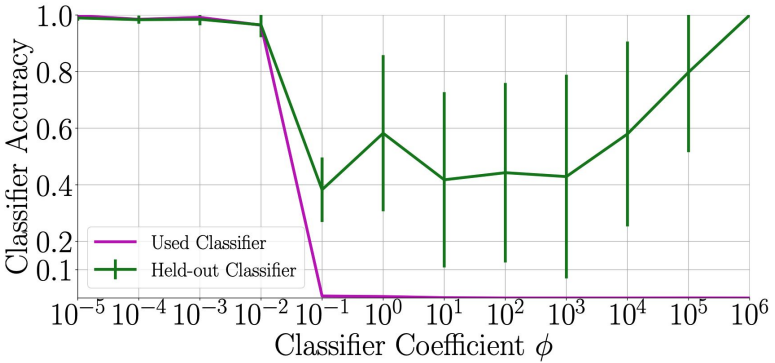

(d) SG2 classifier: MobileNetV2

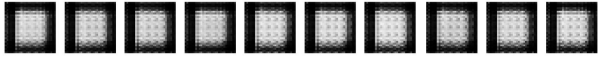
# Measuring the “fooling” performance

- A number of classifier instances are trained in each iteration.
- When testing a GAN trained to fool the previous iteration, we measure the accuracy of held-out GAN classifiers (not included in the modified loss).
- We find contrasting behavior in the two different settings:
  - In the low parameter setting, we find that the DCGAN is unable to fool held-out classifiers.
  - In the high parameter setting, an SG2 generator can successfully fool held-out ResNet-50 classifiers, but fails to fool held-out MobileNetV2 classifiers.

# DCGAN generators unable to fool held-out classifiers

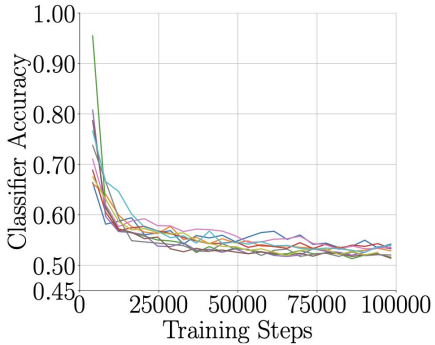
$$\mathcal{L}^*_{\mathbf{G}} = -[\frac{1}{1+\phi}\log(D) + \frac{\phi}{1+\phi}\log(C)]$$




| $\phi$ | DCGAN Image Samples |
|---|---|
| $10^{-2}$ |  |
| $10^{-1}$ |  |
| $10^{0}$ |  |
| $10^{1}$ |  |
| $10^{2}$ |  |
| $10^{3}$ |  |
| $10^{4}$ |  |
| $10^{5}$ |  |
| $10^{6}$ |  |

# SG2 generators can reliably fool ResNet-50 classifiers

- Using a low value of $\phi = 0.001$, we find that the SG2 generators learn to fool the classifier, early in their training, where the rest of the training is guided by the co-trained discriminator. **The output quality of the samples is not affected.**
- Held-out ResNet-50 classifiers can not detect generated images, obtaining a low accuracy of 0.05 on generated images.
- The reliability of this finding suggests that all ResNet-50 classifier instances learn strongly overlapping subsets of artifacts exhibited by the generators.



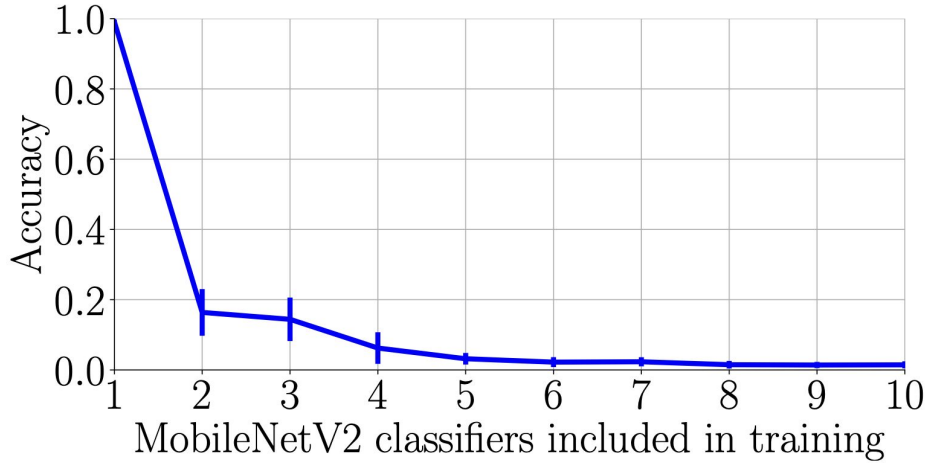

| | GAN trained to fool... | | |
|---|---|---|---|
| Classifier | ResNet-50 | Inception-v3 | MobileNetV2 |
| ResNet-50 | **0.05±0.02** | 0.74±0.12 | 0.5±0.39 |
| Inception-v3 | 0.51±0.25 | **0.16±0.26** | 0.53±0.34 |
| MobileNetV2 | 0.31±0.17 | 0.36±0.16 | 0.41±0.38 |

# MobileNetV2 instances don’t always share artifacts

- With ResNet-50, each classifier instance learns a bulk of the artifacts available to them, and thus have high overlap. With MobileNetV2, however, we found that fooling one classifier instance only fools unseen classifiers instances half the time.
- A MobileNetV2 classifier instance learns a smaller subset of ***available*** artifacts, reducing the probability of overlap between instances, and accordingly, reduces the probability that a generator capable of fooling one will be able to fool others.
- To quantify this diversity, we modify the loss function to include multiple MobileNetV2 classifier instances from the previous iteration, and find that around 10 classifier instances was sufficient to be able to generalize.

# Can fool held-out MobileNetV2 classifier instance if multiple classifier instances are included in training

$$\mathcal{L}^{\Sigma\Sigma}_{\mathbf{G}^{(1)}} = -[\log(D^{(1)}(G^{(1)}(w))) + \phi \sum_{k=0}^{t-1} \log(C^{(0)}_k(G^{(1)}(w)))]$$



# Classifier instances form “mutually fooling” clusters

| MobileNetV2 Classifier | GAN trained to fool MobileNetV2 classifier #... | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | **0.04** | **0.06** | **0.05** | **0.08** | **0.07** | **0.06** | 0.98 | 0.98 | 0.47 | 0.95 |
| 1 | **0.16** | **0.05** | **0.11** | **0.10** | **0.08** | **0.10** | 0.99 | 0.99 | 0.64 | 0.97 |
| 2 | **0.09** | **0.08** | **0.03** | **0.10** | **0.08** | **0.08** | 0.99 | 0.99 | 0.44 | 0.94 |
| 3 | **0.09** | **0.05** | **0.05** | **0.03** | **0.05** | **0.05** | 0.99 | 0.99 | 0.53 | 0.96 |
| 4 | **0.09** | **0.05** | **0.09** | **0.06** | **0.04** | **0.10** | 0.98 | 0.98 | 0.60 | 0.94 |
| 5 | 0.22 | **0.15** | **0.14** | **0.20** | **0.20** | **0.05** | 1.00 | 1.00 | 0.70 | 0.98 |
| 6 | 0.53 | 0.45 | 0.59 | 0.37 | 0.52 | 0.35 | **0.01** | 0.21 | 0.37 | **0.03** |
| 7 | 0.28 | 0.29 | 0.45 | **0.17** | **0.17** | 0.21 | **0.02** | **0.00** | **0.12** | **0.03** |
| 8 | **0.13** | **0.14** | **0.06** | **0.11** | **0.17** | **0.09** | 0.92 | 0.91 | **0.03** | 0.74 |
| 9 | 0.44 | 0.42 | 0.37 | 0.34 | 0.40 | 0.33 | 0.60 | 0.70 | 0.28 | **0.02** |

# Subsequent iterations of GAN/Classifier training

- SG2 generators are able to reliably fool held-out ResNet-50 classifiers in the first iteration; we now investigate the dynamics in subsequent iterations.
- Generators must fool classifiers all each previous iterations. Therefore, classifiers should not detect generated samples of subsequent iterations.
- We train 5 iterations, where a pool of SG2 generators and ResNet-50 classifier instances are trained for each iteration.
- Held-out classifiers are evaluated against samples across all 5 iterations conducted.
- We find that the generators continue to be able to fool classifiers of previous iterations and the visual quality is not affected in the 5 iterations.

# Classifier performance in sequential iterations

| Classifier | GAN Instances | | | | |
|---|---|---|---|---|---|
| | Iteration 0 | Iteration 1 | Iteration 2 | Iteration 3 | Iteration 4 |
| Iteration 0 | **0.99±0.00** | 0.03±0.00 | 0.04±0.01 | 0.05±0.01 | 0.05±0.01 |
| Iteration 1 | 0.00±0.00 | **1.00±0.01** | 0.01±0.00 | 0.06±0.00 | 0.01±0.00 |
| Iteration 2 | 0.00±0.00 | 0.73±0.14 | **0.96±0.05** | 0.01±0.00 | 0.03±0.01 |
| Iteration 3 | 0.00±0.00 | 0.87±0.11 | 0.18±0.13 | **0.93±0.07** | 0.02±0.00 |
| Iteration 4 | 0.01±0.00 | 0.71±0.14 | 0.51±0.14 | 0.07±0.03 | **0.86±0.11** |
| Mean FID | 36.98 | 36.62 | 36.67 | 36.39 | 36.83 |

Expectedly, classifiers from each iteration are unable to detect generated images from subsequent iterations.

# Classifier performance in sequential iterations

| Classifier | GAN Instances | | | | |
|---|---|---|---|---|---|
| | Iteration 0 | Iteration 1 | Iteration 2 | Iteration 3 | Iteration 4 |
| Iteration 0 | **0.99±0.00** | 0.03±0.00 | 0.04±0.01 | 0.05±0.01 | 0.05±0.01 |
| Iteration 1 | 0.00±0.00 | **1.00±0.01** | 0.01±0.00 | 0.06±0.00 | 0.01±0.00 |
| Iteration 2 | 0.00±0.00 | 0.73±0.14 | **0.96±0.05** | 0.01±0.00 | 0.03±0.01 |
| Iteration 3 | 0.00±0.00 | 0.87±0.11 | 0.18±0.13 | **0.93±0.07** | 0.02±0.00 |
| Iteration 4 | 0.01±0.00 | 0.71±0.14 | 0.51±0.14 | 0.07±0.03 | **0.86±0.11** |
| Mean FID | 36.98 | 36.62 | 36.67 | 36.39 | 36.83 |

The converse is not always true: higher-iteration classifiers can sometimes, but not always, detect lower-iteration generators.

# Classifier performance in sequential iterations

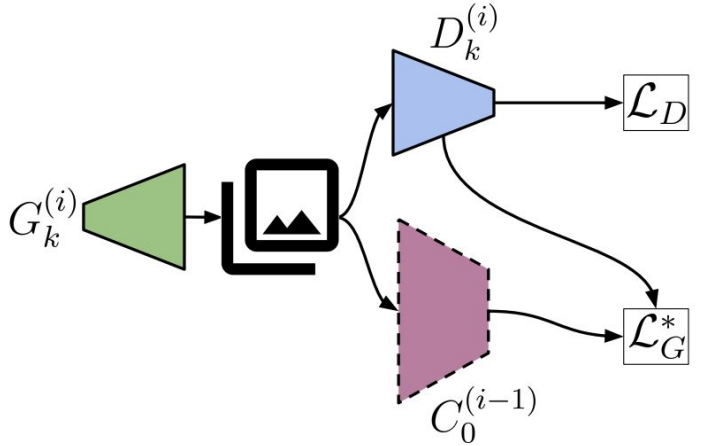
| Classifier | GAN Instances | | | | |
|---|---|---|---|---|---|
| | Iteration 0 | Iteration 1 | Iteration 2 | Iteration 3 | Iteration 4 |
| Iteration 0 | **0.99±0.00** | 0.03±0.00 | 0.04±0.01 | 0.05±0.01 | 0.05±0.01 |
| Iteration 1 | 0.00±0.00 | **1.00±0.01** | 0.01±0.00 | 0.06±0.00 | 0.01±0.00 |
| Iteration 2 | 0.00±0.00 | 0.73±0.14 | **0.96±0.05** | 0.01±0.00 | 0.03±0.01 |
| Iteration 3 | 0.00±0.00 | 0.87±0.11 | 0.18±0.13 | **0.93±0.07** | 0.02±0.00 |
| Iteration 4 | 0.01±0.00 | 0.71±0.14 | 0.51±0.14 | 0.07±0.03 | **0.86±0.11** |
| Mean FID | 36.98 | 36.62 | 36.67 | 36.39 | 36.83 |

Generalization ability of the classifiers remains high in the initial iterations, suggesting the existence of an “artifact preference”: Precluding one set of artifacts leads predictably to the generation of a new set of artifacts.

# Order preference confirmed in “memoryless” setting

- In this setting, generators must only fool a classifier from the previous iteration, rather than all preceding iterations.
- When fooling the previous classifier of the previous iteration $C^{(i-1)}$, we find that the samples are detectable by the classifier $C^{(i-2)}$ from two iterations ago.
- The notion that SG2 generators produce artifacts according to an order preference is reinforced with this finding.

$$\mathcal{L}^{*}_{\mathbf{G}^{(i)}} = -[\log(D^{(i)}(G^{(i)}(w))) + \phi \log(C_0^{(i-1)}(G^{(i)}(w)))]$$



# Summary

- Artifacts present in generator outputs are consistent across independently trained generators of the same GAN architecture.
- When trained against a GAN-classifier, generators can not eliminate output artifacts learned by a held-out classifier in the low-parameter setting.
- However, generators can fool GAN-classifiers and move to a new space of artifacts in the high-parameter setting.
- This new space is shared by independently trained generator instances of the new iteration, suggesting the existing of an orderly preference in artifacts exhibited by generators.
- This hypothesis is reinforced when the GAN is trained in the “memoryless” setting, to only fool the classifier from the immediately preceding iteration. These GANs are only able to fool classifiers of their iteration parity.

# Summary

- Generators are unable to fool GAN-classifiers of an unseen architecture, therefore, the set of artifacts learned by a GAN-classifier is strongly dependent on the classifier architecture.
- Unlike ResNet-50, a MobileNetV2 instance only learns a subset of artifacts available to learn using the MobileNetV2 architecture: fooling a held-out classifier instance requires including multiple classifier instances in training.
- Upon further investigation, we find that MobileNetV2 classifier instances appear to form “mutually-fooling” clusters where classifier instances within a cluster learn a shared set of artifacts, supporting our hypothesis that MobileNetV2 classifiers learn distinct subsets of the artifact space.