ONE CLASS AT A TIME: SINGLE-CLASS BATCH SAMPLING FOR TRAINING CLASS-CONDITIONAL GANS ON LIMITED DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Training conditional generative adversarial networks on limited data remains a challenging open problem, with severe mode collapse resulting from discriminator overfitting and insufficient per-class supervision. Existing approaches address this through data augmentation, regularisation, and knowledge sharing between classes, but the role of batch composition in shaping the discriminator's learned representations has received little attention. We investigate the effect of single-class batch sampling, wherein each training batch is composed exclusively of images from a single class, on conditional GAN training under limited data regimes. We argue that standard mixed-class batches allow the discriminator to exploit coarse inter-class differences as a shortcut for real/fake discrimination, reducing pressure to capture intra-class features and consequently weakening the signal provided to the generator. Single-class batches eliminate this shortcut, forcing the discriminator to attend to intra-class structure. Experiments across multiple limited data benchmarks demonstrate consistent improvements in quality and diversity. Our results suggest that batch composition is an overlooked but practically significant axis of design in conditional GAN training in limited data.

1 INTRODUCTION

Generative adversarial networks (GANs) are at the forefront of image generation capabilities (Chakraborty et al., 2024; Islam et al., 2024). Despite encouraging advances in diffusion models (Croitoru et al., 2023; Kazerouni et al., 2023), GANs still tend to achieve higher fidelity to the training distribution at a smaller computational cost and are more data-efficient (Croitoru et al., 2023; Kong et al., 2024; Zhang et al., 2025). These advantages are important in real world applications, where it may be difficult for practitioners of various fields to gather datasets with more than a few thousand or a few hundred samples due to, e.g., data collection costs and data privacy requirements.

Indeed, as modern GAN backbones (Brock et al., 2019; Karras et al., 2020b; Huang et al., 2024) become more robust and easier to train, there has been recent interest in applying them in scenarios of increasingly challenging data availability (Li et al., 2022; Yang & Wang, 2024). This limited data problem is exacerbated in the common case where we wish to train models on data conditioned with class labels y. Training class-conditional generative models requires learning $p(x|y)$ (as opposed to simply $p(x)$) and is an interesting task since it enables the user to generate samples belonging to a particular class, as opposed to sampling randomly from the learned manifold. As such, there has been significant effort in recent years to improve training of class-conditional GANs in diverse data availability settings (Shahbazi et al., 2022; Rangwani et al., 2022; 2023; Khorram et al., 2024).

However, training conditional GANs comes with its own set of challenges: Shahbazi et al. (2022) first identified that while unconditional GANs can achieve satisfactory performance in smaller datasets, CGANs suffer from severe mode collapse, whereby generated samples for each class exhibit very low diversity. The authors found that starting CGAN training in a completely unconditional manner and gradually incorporating conditioning significantly improved performance in a set of small datasets.

We hypothesize that at least part of the benefit of unconditional training comes from alleviating the CGAN of the burden of the conditioning task, allowing it to focus on more fine-grained details

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

within each class to distinguish between real and fake samples. Motivated by this reasoning, we propose training CGANs on limited data using single-class batches so that the CGAN may focus on each class separately. The idea is that rather than weakening the conditioning signal, single-class batches will force the discriminator to learn more precise intra-class real/fake boundaries.

With this simple change to the training regime, we obtain substantial and robust performance improvements, as measured using several state-of-the-art metrics, across a diverse set of limited image datasets. We also find that, in long-tailed datasets, single-class batches introduce a tradeoff between adequately learning classes with sufficient representation and allowing minority classes to benefit from feature sharing from having other classes in the same batch. As such, we propose using single-class batches only for classes with sufficient representation as measured by number of samples. We find that performance is not sensitive to this threshold, and are able to achieve state-of-the-art performance in several imbalanced datasets.

2 Background

2.1 Conditional Generative Adversarial Networks

Generative Adversarial Networks (GANs) define a framework for training generative models using a minimax game between two neural networks: the generator G and discriminator D. The generator is a mapping $G : \mathcal{Z} \to \mathcal{X}$, where $\mathcal{Z}$ is sampled from a tractable, known distribution p_Z (e.g., the unit Gaussian) and $\mathcal{X}$ is the data space. Given real data $x \sim p_X$ and fake data sampled from the generator $x \sim p_G$, a general formulation of GAN training is given by:

$$\min_G \max_D V(G, D), \text{ where } V(G, D) = \mathbb{E}_{x \sim p_X}[f(-D(x))] + \mathbb{E}_{z \sim p_Z}[f(D(G(z)))], \tag{1}$$

where f is flexible (Huang et al., 2024) and determines the divergence being minimized (Li & Farnia, 2023). Conditional GANs (CGANs) extend the usual GAN value function in Equation 1 by including an additional conditioning signal y:

$$V_c(G, D) = \mathbb{E}_{(x,y) \sim p_X}[f(-D(x, y))] + \mathbb{E}_{z \sim p_Z}[f(D(G(z, y)))] \tag{2}$$

This changes the task from learning the data density $p(x)$ to learning $p(x|y)$, which allows users to guide sample generation during inference by choosing desirable y. Throughout this paper, we assume y to be a class label.

2.2 Training CGANs on Limited Data

As GAN architectures and training methods become more mature, recent efforts have focused on developing methods to train them in limited data regimes. These approaches include data augmentation (Karras et al., 2020a; Zhao et al., 2020), regularization (Tseng et al., 2021; Jiang et al., 2021; Zhang et al., 2025), and architectural changes (Yang et al., 2022; Saxena et al., 2024; 2025).

However, it was Shahbazi et al. (2022) that first studied the issues of training CGANs on limited data, finding that conditional models would collapse whereas unconditional models would achieve satisfactory performance. Indeed, the types of collapse that we observe in unconditional and conditional GANs are fundamentally different: while unconditional GANs tend to ignore certain modes of the data, they usually generate high-quality samples for the modes they do learn.

In CGANs, since the class label y is such a strong prior, the model rarely ignores classes. Instead, when training with limited data, typically several classes will have few examples, and the discriminator quickly learns to reject samples that do not look very similar to those in the training data. This results in unstable training and in a form of collapse where the generator struggles to capture the diversity in each class, effectively reproducing one or a few samples that look identical to those in the training data. As such, there has been increased interest in developing methods to train CGANs on limited data (Shahbazi et al., 2022; Rangwani et al., 2022; 2023; Khorram et al., 2024).

In this paper, we explore the possibility of using single-class batches for training CGANs in limited datasets, which is orthogonal to these efforts. In Section 3 we establish the experimental setup used

throughout this paper, and in Section 4 we present our motivation for experimenting with single-class batches and discuss the resulting impact of this design choice.

3 Experimental Setup

Baseline Throughout our experiments we use the well-established StyleGAN2 (Karras et al., 2020b), with adaptive data augmentation (Karras et al., 2020a), to obtain a strong baseline. We choose this model as a baseline due to its wide adoption and variety of use in downstream tasks (Katsumata et al., 2024; Azari & Lim, 2024), its superior performance against diffusion models in limited data settings (Zhang et al., 2025), and due to its accessible code base. [1]

Datasets We test single-class batches on a diverse set of datasets, ranging from perfectly balanced to extremely imbalanced, across different domains and sample counts.

- Carnivores: A subset of the ImageNet dataset (Deng et al., 2009). Following (Shahbazi et al., 2022), we use a subset of the dataset with 20 classes and 100 images per class.
- CUB-200-2011: The original dataset (Wah et al., 2011) contains 200 categories of bird species with ≈ 60 samples per class. We use a subset of 20 classes and use all available samples for each class. The total amount of samples is 600.
- DTD: The Describable Textures Dataset (Cimpoi et al., 2014) is a benchmark for texture recognition focusing on 47 human-interpretable visual categories of textures. We limit the amount of samples per class to 40, having a total amount of samples of $1,880$.
- AnimalFaces-LT: The AnimalFaces dataset (Si & Zhu, 2012) contains images of 20 species of animals. We modify its degree of imbalance ρ, which is the ratio of the numbers of samples between the top and bottom classes, as done by (Rangwani et al., 2023; Khorram et al., 2024). This results in a total of $1,756$ samples. We suffix the dataset name with *-LT* to denote that it is long-tailed.
- Flowers-LT: The Flowers dataset (Nilsback & Zisserman, 2008) (GPL-2.0) is a fine-grained image classification benchmark with 102 flower species. We modify it to have $\rho = 100$, resulting in a total sample count of $5,285$.

Metrics We report several state-of-the-art metrics for evaluating generative image models:

- FID_{CLIP}: The Fréchet Inception Distance (FID) (Heusel et al., 2017) is a standard metric for assessing generative image models. It computes the Wasserstein-2 distance between a sample of the real and learned data distributions in the feature space of a pretrained InceptionV3 network (Szegedy et al., 2016) (if we assume both distributions to be normal, the Wasserstein-2 is the same as the Fréchet distance). FID is a global measure of the quality and diversity of generated samples. However, Kynkäänniemi et al. (2022) criticize the choice of InceptionV3 as a backbone, pointing out that it does not always align with human judgment on non-ImageNet data. Instead, they propose computing the empirical Fréchet distance using CLIP, which benefits from a massively richer feature space (Radford et al., 2021). We refer to the FID computed in CLIP feature space as FID_{CLIP}.
- $\text{iFID}_{\text{CLIP}}$: The so-called *intra-FID* (Miyato & Koyama, 2018) computes the Fréchet distance separately for each class and takes the (unweighed) mean.
- Precision and Recall: While FID is a good metric for overall model quality and diversity, it fails to separate these quantities. Precision and recall (Sajjadi et al., 2018; Kynkäänniemi et al., 2019) are explicit approximations for distribution coverage. Precision measures how much of the learned distribution overlaps with the true distribution, i.e., what fraction of generated samples could have plausibly been taken from the real distribution. Recall measures how much of the target distribution is covered by the learned distribution, i.e., it is a measure of diversity.
- CMMD: (Jayasumana et al., 2024) argue against the normality assumption of FID and compute the maximum mean discrepancy (using a Gaussian RBF kernel) in the CLIP feature space. They call this metric CMMD.

[1] https://github.com/NVlabs/stylegan2-ada-pytorch/tree/main



Figure 1: Generated images from models trained on the AnimalFaces-LT dataset. Using single-class batches (SCB) prevented collapse on class 5 and attained greater sample diversity.

Table 1: Results for the Carnivores dataset

Methods	$\text{FID}_{\text{CLIP}}(\downarrow)$	$\text{iFID}_{\text{CLIP}}(\downarrow)$	Precision(↑)	Recall(↑)	CMMD(↓)
SG2+ADA	$9.02_{\pm 0.29}$	$16.59_{\pm 0.32}$	$\mathbf{0.91}_{\pm 0.00}$	$0.14_{\pm 0.01}$	$34.88_{\pm 1.55}$
+SCB	$\mathbf{7.42}_{\pm 0.10}$	$\mathbf{15.25}_{\pm 0.19}$	$0.88_{\pm 0.00}$	$\mathbf{0.20}_{\pm 0.01}$	$\mathbf{26.81}_{\pm 0.66}$

4 Single-Class Batches in Conditional GANs

Knowledge sharing methods for training CGANs (Shahbazi et al., 2022; Khorram et al., 2024) on limited and long-tailed data either reduce the strength of the conditional objective or add unconditional objectives in order to improve feature sharing between classes and alleviate the burden of the conditional objective during training. These methods are not architecture-agnostic and require hyperparameter tuning. We take a different view: rather than weakening the influence of the conditional objective during training, we propose using single-class batches while allowing the generator to receive better gradients.

Our reasoning is as follows: over many updates of traditional mixed batches, the discriminator is trained to represent directions in its feature space that are discriminative for real and fake samples of all classes at once. If some of these directions are more salient or easier to learn (e.g., from classes with a greater number of samples), they tend to dominate earlier gradient signal to the generator. This is undesirable, since then the discriminator can become very powerful using only coarse inter-class features rather than learning intra-class features and passing them on to the generator.

By using single-class batches, in each forward pass the discriminator learns to distinguish between real and fake samples which come from a single class distribution. As such, it cannot learn to exploit coarse inter-class features that allow it to easily distinguish between real/fake samples. Instead, it must focus on which features, for each class, the generator is not yet reproducing adequately. We present empirical evidence for this phenomenon in Section 5.

4.1 Results on Balanced Datasets

Tables 1 to 3 show the results of adding single-class batches (SCB) to an SG2+ADA baseline (mean and standard deviation across 5 runs). In all cases, measures of overall quality and fidelity (Fréchet-based metrics and CMMD) improved with single-class batches. We also show in Fig. 1 images generated from the SG2+ADA baseline and when single-class batches are used.

Table 2: Results for the CUB-200-2011 dataset

Methods	$\text{FID}_{\text{CLIP}}(\downarrow)$	$\text{iFID}_{\text{CLIP}}(\downarrow)$	Precision(↑)	Recall(↑)	CMMD(↓)
SG2+ADA	$20.04_{\pm 1.13}$	$33.32_{\pm 1.22}$	$0.74_{\pm 0.04}$	$0.00_{\pm 0.00}$	$79.28_{\pm 5.35}$
+SCB	$\mathbf{10.55}_{\pm 1.27}$	$\mathbf{22.91}_{\pm 1.34}$	$\mathbf{0.83}_{\pm 0.03}$	$\mathbf{0.03}_{\pm 0.01}$	$\mathbf{33.21}_{\pm 5.95}$

Table 3: Results for the DTD256 dataset

Methods	$\text{FID}_{\text{CLIP}}(\downarrow)$	$\text{iFID}_{\text{CLIP}}(\downarrow)$	Precision(↑)	Recall(↑)	CMMD(↓)
SG2+ADA	$10.07_{\pm 0.24}$	$30.14_{\pm 0.58}$	$\mathbf{0.82}_{\pm 0.01}$	$0.10_{\pm 0.00}$	$25.47_{\pm 1.12}$
+SCB	$\mathbf{9.31}_{\pm 0.32}$	$\mathbf{29.87}_{\pm 0.37}$	$0.77_{\pm 0.01}$	$\mathbf{0.21}_{\pm 0.02}$	$\mathbf{25.01}_{\pm 1.09}$

Table 4: Results for the AnimalFaces-LT dataset

Methods	$\text{FID}_{\text{CLIP}}(\downarrow)$	$\text{iFID}_{\text{CLIP}}(\downarrow)$	Precision(↑)	Recall(↑)	CMMD(↓)
SG2+ADA	$9.00_{\pm 1.88}$	$44.45_{\pm 0.62}$	$0.86_{\pm 0.02}$	$0.07_{\pm 0.03}$	$22.01_{\pm 5.53}$
+SCB	$7.17_{\pm 0.29}$	$42.81_{\pm 0.19}$	$0.87_{\pm 0.01}$	$\mathbf{0.15}_{\pm 0.02}$	$17.36_{\pm 1.15}$
+SCB ($T = 100$)	$\mathbf{6.74}_{\pm 0.15}$	$\mathbf{42.59}_{\pm 0.26}$	$\mathbf{0.87}_{\pm 0.00}$	$0.14_{\pm 0.02}$	$\mathbf{16.62}_{\pm 0.89}$

Table 5: Results for the Flowers-LT dataset

Methods	$\text{FID}_{\text{CLIP}}(\downarrow)$	$\text{iFID}_{\text{CLIP}}(\downarrow)$	Precision(↑)	Recall(↑)	CMMD(↓)
SG2+ADA	$2.07_{\pm 0.12}$	$13.51_{\pm 0.64}$	$\mathbf{0.87}_{\pm 0.00}$	$0.05_{\pm 0.01}$	$\mathbf{5.27}_{\pm 0.39}$
+SCB	$1.84_{\pm 0.09}$	$17.01_{\pm 0.52}$	$0.78_{\pm 0.02}$	$\mathbf{0.28}_{\pm 0.02}$	$5.33_{\pm 0.45}$
+SCB ($T = 100$)	$\mathbf{1.83}_{\pm 0.20}$	$\mathbf{13.45}_{\pm 0.50}$	$0.80_{\pm 0.01}$	$0.23_{\pm 0.02}$	$5.39_{\pm 0.94}$

We also note significant increases in recall, which indicate that the CGAN was able to model more of the target distribution. These support our hypothesis that single-class batches force the discriminator to learn intra-class features, which in turn forces the generator to create more diverse samples.

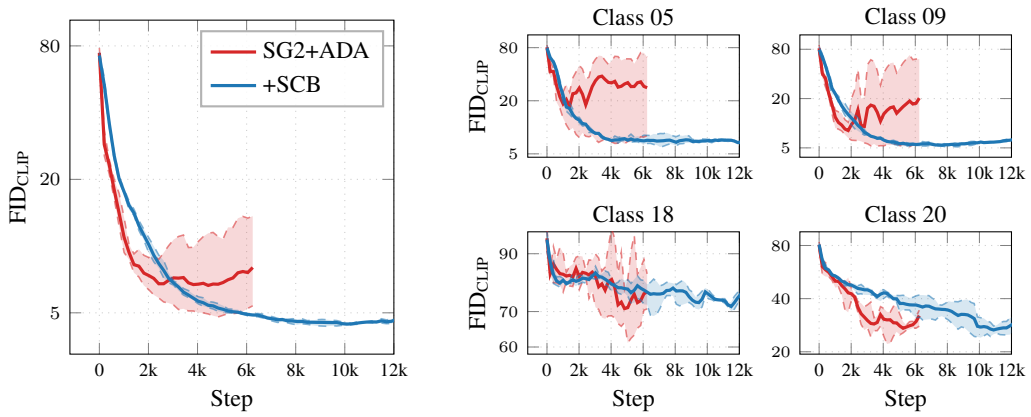
Finally, we highlight that while the baseline retained higher precision in two of the three datasets, this is somewhat expected. Since precision quantifies how much of the learned distribution overlaps with the target distribution, it can be made arbitrarily close to 1 by reducing diversity. A consequence of this is that we can trade off recall for precision by using the so-called *truncation trick* (Brock et al., 2019), whereby the user clips latent vectors to remain in a higher-probability region of the latent space while sacrificing diversity.

4.2 Results on Long-Tailed Datasets

Tables 4 and 5 show the results of adding single-class batches to an SG2+ADA baseline on datasets with imbalance ratios of 25 and 100, respectively. Recall increased significantly for both datasets. We highlight that, for both real and fake samples, class labels are sampled according to their frequency in the dataset. This is a standard procedure (Karras et al., 2020b; Khorram et al., 2024), and gives much better results than sampling class labels uniformly (see the Appendix for more details).

These results help understand the effect of single-class batches. In AnimalFaces-LT (Table 4), though FID_{CLIP} improved significantly, the same cannot be said of $\text{iFID}_{\text{CLIP}}$, where gains were more modest. This means that reductions in FID_{CLIP} were mostly due to improvements in head classes (the classes with the greatest amounts of samples), which account for the majority of the distribution. Figure 2a shows that overall convergence is greatly improved with single-class batches, while Figure 2b illustrates that tail classes may not receive the same benefit, and may even incur performance degradation. In the Appendix, we show per-class FID scores for AnimalFaces-LT.

This effect was more pronounced in Flowers-LT, with single-class batches causing $\text{iFID}_{\text{CLIP}}$ to degrade even though FID_{CLIP} improved, reinforcing that the benefits of single-class batches are in large part due to improved learning of head classes. With these observations in mind, we consider whether we can retain the benefits of single-class batches without degrading performance on tail classes.



(a) FID_{CLIP} convergence on the full dataset. (b) Per-class FID_{CLIP} for a representative subset of classes.

Figure 2: FID_{CLIP} training curves for StyleGAN2+ADA on AnimalFaces-LT using mixed (red) and single-class batches (SCB, blue). Solid lines show the median over 5 runs; dashed lines and shaded regions indicate the min/max range. Single-class batches cause training to be much more stable, though full convergence takes longer. **(Left)** FID_{CLIP} computed using the entire dataset. Using single-class batches greatly reduces variance in the performance between training runs and significantly improves results. **(Right)** Per-class FID_{CLIP} for four classes. Using single-class batches seems to deteriorate performance on tail classes, though it still decreases variance between runs.

4.3 Single-Class Batches on Sufficiently Well-Represented Classes

We identify a direct tradeoff between global performance, mostly due to improvements in head classes, and performance degradation on tail classes. Indeed, it is known that mixed batches encourage knowledge sharing (Hou et al., 2022). As such, an immediate concern is whether we can find an adequate balance between these objectives. To that end, we consider two options:

1. Divide each batch into equal-size buckets and sample a class label separately for each bucket. For example, if we have a batch size of 64 and divide it into 4 buckets, then each bucket would contain 16 samples belonging to a given class. We find that this allows us to neatly trade off convergence stability and global performance for performance in tail classes. We present results for this option in the Appendix;

2. An option which achieves better results is to only use single-class batches for classes with a sufficient amount of samples, which we expect would benefit less from feature sharing. We adopt this procedure throughout the rest of the paper, and explain it in more detail below.

For a dataset with a total of N samples and classes $\mathcal{C} = \{1, \dots, C\}$, we define a threshold T such that classes with sample count greater or equal to T are considered head classes and those with sample count smaller than T are considered tail classes:

$$\mathcal{C}_{\text{head}} = \{c \in \mathcal{C} \mid \text{count}(c) \geq T\}, \quad \mathcal{C}_{\text{tail}} = \{c \in \mathcal{C} \mid \text{count}(c) < T\}. \tag{3}$$

The hyperparameter T determines which classes appear in single-class batches and which classes appear in traditional mixed-class batches. With $N_{\text{head}} = \sum_{c \in \mathcal{C}_{\text{head}}} \text{count}(c)$ the total amount of samples in head classes, then, for each batch:

- With probability $\frac{N_{\text{head}}}{N}$, we sample *a single label* from $\mathcal{C}_{\text{head}}$. Each class $c \in \mathcal{C}_{\text{head}}$ is chosen with probability proportional to its frequency in $\mathcal{C}_{\text{head}}$:

$$P(c | c \in \mathcal{C}_{\text{head}}) = \frac{\text{count}(c)}{N_{\text{head}}} \tag{4}$$

- With probability $1 - \frac{N_{\text{head}}}{N}$, *for each sample in the batch*, we sample a class label from $\mathcal{C}_{\text{tail}}$ with probability proportional to the class' frequency within $\mathcal{C}_{\text{tail}}$:

$$P(c|c \in \mathcal{C}_{\text{tail}}) = \frac{\text{count}(c)}{\sum_{c' \in \mathcal{C}_{\text{tail}}} \text{count}(c')} \tag{5}$$

This procedure ensures that we only use single-class batches whose sample counts are greater than T and that all classes are still sampled according to their relative frequency in the dataset, as done in the baseline.

Tables 4 and 5 show the results of using single-class batches for class with a minimum of 100 samples (SCB ($T = 100$)). Performance generally improves significantly. We achieve good results for several values of T and present these ablations in Section 5.4.

5 Analysis

5.1 On Discriminator Feature Discriminability

We provide direct empirical evidence that using single-class batches results in the discriminator learning more precise intra-class real/fake boundaries.

Table 6: Classification and real/fake discrimination linear probe accuracies, on AnimalFaces-LT discriminator features, for SG2ADA with and without SCB.

Method	Class. Acc. ↑	Disc. Acc. ↑
SG2ADA	98.69 ± 0.17	91.42 ± 2.35
+SCB	98.67 ± 0.67	98.67 ± 0.41

Table 6 shows classification and real/fake discrimination accuracies for a linear probe trained on AnimalFaces-LT discriminator features. While they achieve identical accuracy in distinguishing between classes, the discriminator trained with single-class batches is much better at distinguishing between real and fake samples, supporting our hypothesis that it learns more fine-grained intra-class features.

5.2 Analysis of Overfitting

We find that single-class batches are very protective against overfitting. Rangwani et al. (2022) and Khorram et al. (2024) analysed the spectra of the layers of the generator as a means to identify overfitting. They report that when the generator focuses on a few predominant coarse features, this is reflected on an increase of the generator's spectral norms (in the case of the StyleGAN2 generator, we look at the spectral norm of the intermediate latent vectors w). Indeed, recent methods for stabilizing GAN training and reducing mode collapse explicitly regularize the spectral norms of either or both networks (Lin et al., 2021; Rangwani et al., 2022).

In Figure 3 we show that by simply using single-class batches, the spectral norm of the intermediate latent vectors is significantly decreased even though no spectral regularization is employed, indicating that the generator is unable – or at least significantly less able – to pick out coarse inter-class features and use them to distinguish between real and fake samples.

Further evidence that single-class batches reduce generator overfitting is presented in Figure 4. ADA's p denotes the probability with which augmentations are applied to real and fake images (Karras et al., 2020a). Since p is increased adaptively throughout training based on the average confidence of the discriminator on real samples, the fact that single-class batches make p increase slower implies that it is harder for the discriminator to distinguish between real and fake samples.

Similarly, R_1 regularization (Mescheder et al., 2018) is incorporated in StyleGAN2 (Karras et al., 2020b), and measures the squared gradient of the discriminator output with respect to real images. A lower R_1 penalty means that the discriminator is less sensitive around real samples, i.e., that its decision boundary is smoother. This indicates a lower degree of overfitting and, since the discriminator is less sensitive to diversity in observed samples, allows the generator to become more diverse. These findings are in line with the results presented in Section 4, which show significant increases in diversity through the recall metric.

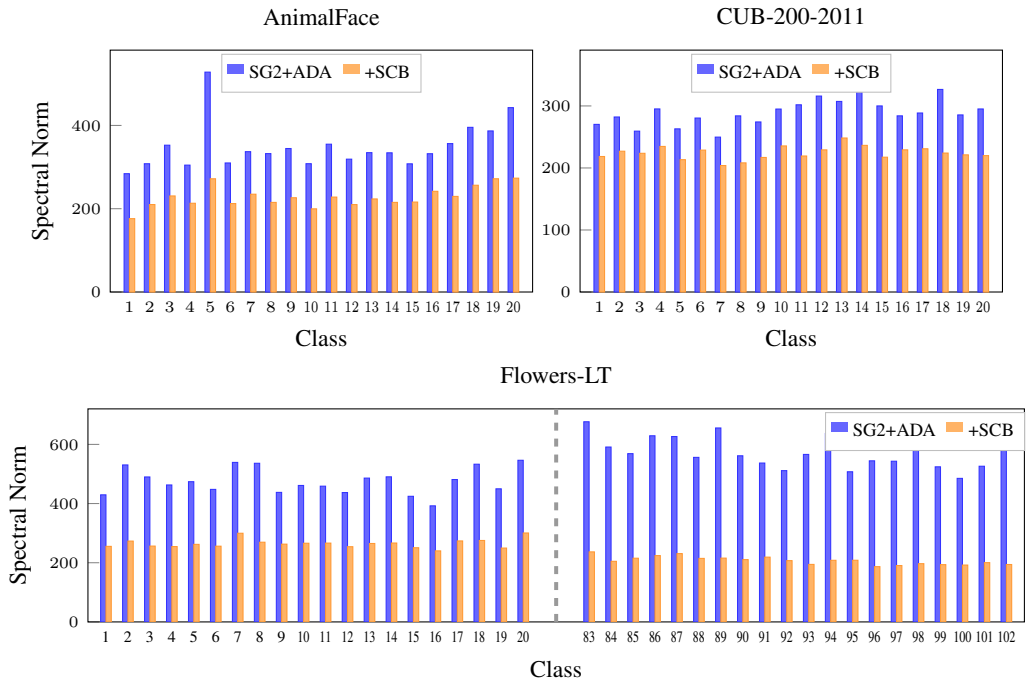


Figure 3: Per-class spectral norm of intermediate latent vectors w for StyleGAN2+ADA with and without single-class batches (SCB) across three datasets. Spectral norms are significantly reduced without any explicit spectral regularization.

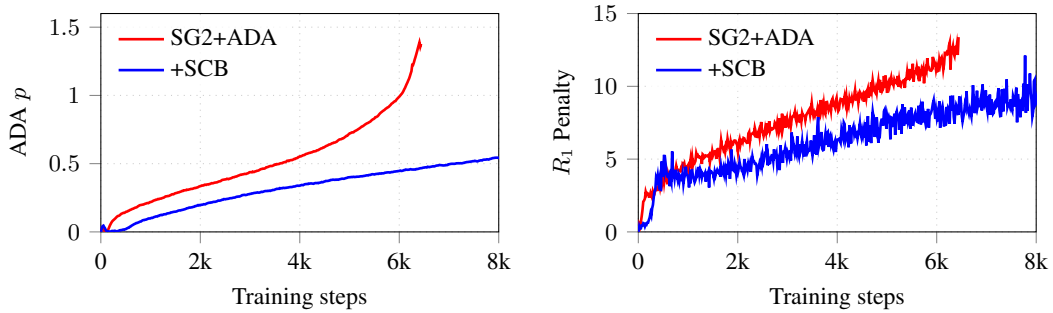


Figure 4: Evolution of ADA's p and the R_1 penalty throughout training in the AnimalFaces-LT dataset. In both cases, higher values indicate greater levels of overfitting. Using single-class batches reduces both, indicating that it is preventing overfitting.

5.3 Limitations and Applicability

We now discuss the applicability and some limitations of single-class batches:

- There is a tradeoff between the global improvements given by single-class batches and possible performance degradation in tail classes. It is known that mixed batches encourage knowledge sharing (Hou et al., 2022), and single-class batches eliminate this entirely. However, that seems to only be an issue in extremely imbalanced datasets with a larger amount of samples: we observed $\text{iFID}_{\text{CLIP}}$ degradation only on Flowers-LT, which has significantly more samples than the other datasets. Performing single-class batches only for classes with sufficient representation seems to solve this issue.
- Relating to the previous point, performance degrades as the total sample count increases. From the CUB-200-2011 dataset, where we saw dramatic increases in performance (Table 2) to Flowers-LT (Table 5), improvements from using single-class batches seem to diminish

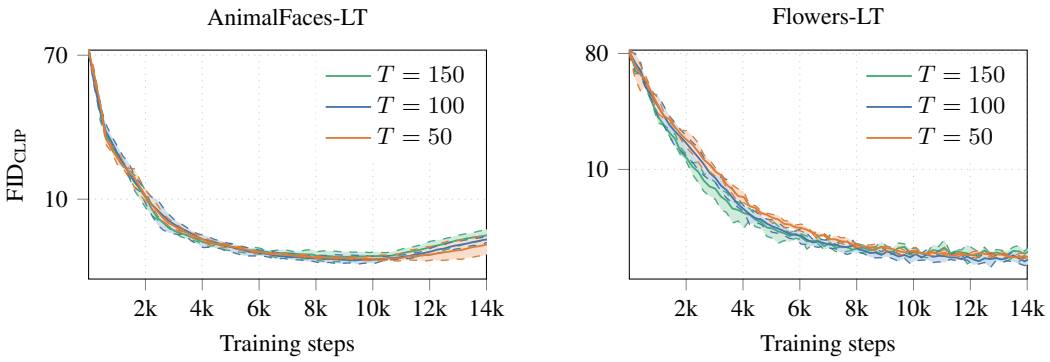


Figure 5: Ablation over the minimum threshold T for applying single-class batches to StyleGAN2+ADA on the AnimalFaces-LT (left) and Flowers-LT (right) datasets. Solid lines show the mean over runs; dashed lines and shaded regions indicate min/max range. Note the log scale.

as sample count increases. This is somewhat expected, as with sufficient samples it is more difficult for the discriminator to overfit to the training set.

- As shown in Figure 2, converging takes longer when using single-class batches. Since by problem definition we are dealing with limited datasets, and given the demonstrated performance increases, we expect that the increase in training time is worthwhile.

5.4 Ablations

To mitigate potential performance degradation on long-tailed datasets, we use single-class batches only for classes with at least T samples. Figure 5 shows results for different values of T, demonstrating that performance is not sensitive to this choice. In the Appendix, we further evaluate batch sizes of 32, 64, and 128 and find that single-class batches consistently improve performance across all three settings.

6 Conclusion

We introduced single-class batch sampling as a simple and architecture-agnostic modification to the training of conditional GANs in limited data regimes. By restricting each batch to a single class, we prevent the discriminator from exploiting coarse inter-class differences and instead encourage it to focus on intra-class structure. This results in better gradients for the generator, leading to consistent improvements in sample quality and diversity across a range of balanced and imbalanced datasets.

Our experiments demonstrate that this effect is particularly beneficial in low-data settings, where discriminator overfitting is a primary challenge. We provide empirical evidence that single-class batches increase learning of intra-class features, as indicated by discriminator features being more discriminable for real/fake samples, and that they mitigate overfitting, as indicated by slower increases in ADA augmentation strength and lower R_1 penalties, as well as reduced spectral norms in the generator. These improvements translate into higher recall and better coverage of the target distribution, while achieving improvements in quality.

We also identify an important practical tradeoff in long-tailed datasets: while single-class batches significantly improve performance on well-represented (head) classes, they can degrade performance on underrepresented (tail) classes due to reduced knowledge sharing. To address this, we propose a hybrid strategy that applies single-class batches only to sufficiently well-represented classes, achieving strong overall performance while preserving benefits for tail classes. Notably, performance is robust to the choice of threshold, making this approach practical in real-world settings.

Overall, our results highlight batch composition as an overlooked but impactful design dimension in conditional GAN training. We believe that further exploration of sampling strategies may yield additional gains and improve our understanding of optimization dynamics in conditional generative models, and hope that our findings make GANs more accessible to practitioners in various fields.

AI USE STATEMENT

In this work, we used generative AI tools for creating and modifying scientific figures or images, creating or editing software code, creating artifacts, sourcing/searching for information, and identifying relevant literature. We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results, formulate questions for surveys or interviews, suggest experimental parameters, draft parts of a research paper, transcribe recordings of research material, summarize or analyse existing literature, discover research topics or identify gaps, brainstorming, edit a research paper to improve readability, format references, suggest a structure for a research paper, propose a title or keywords for a research paper.

We have reviewed all AI-assisted work. We checked the correctness of software artifacts generated with AI assistance and manually inspected generated figures. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

As stated in the main paper, we base our experiments on the official StyleGAN2+ADA implementation, which is open and accessible on GitHub. We include a link to this code in the paper; the modification we propose is straightforward. The datasets are adequately cited, and their characteristics and modifications (e.g., degree of long-tailedness) are made clear in the main text.

REFERENCES

Bita Azari and Angelica Lim. Emostyle: One-shot facial expression editing using continuous emotion parameters. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6385–6394, 2024.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis, 2019. URL https://arxiv.org/abs/1809.11096.

Tanujit Chakraborty, Ujjwal Reddy KS, Shraddha M Naik, Madhurima Panja, and Bayapureddy Manvitha. Ten years of generative adversarial nets (gans): a survey of the state-of-the-art. *Machine Learning: Science and Technology*, 5(1):011001, 2024.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(9): 10850–10869, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

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. *Advances in neural information processing systems*, 30, 2017.

Zhi Hou, Baosheng Yu, and Dacheng Tao. Batchformer: Learning to explore sample relationships for robust representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7256–7266, 2022.

Nick Huang, Aaron Gokaslan, Volodymyr Kuleshov, and James Tompkin. The gan is dead; long live the gan! a modern gan baseline. *Advances in Neural Information Processing Systems*, 37: 44177–44215, 2024.

Showrov Islam, Md Tarek Aziz, Hadiur Rahman Nabil, Jamin Rahman Jim, Muhammad Firoz Mridha, Md Mohsin Kabir, Nobuyoshi Asai, and Jungpil Shin. Generative adversarial networks (gans) in medical imaging: Advancements, applications, and challenges. *IEEE Access*, 12:35728–35753, 2024.

Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9307–9315, 2024.

Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Deceive d: Adaptive pseudo augmentation for gan training with limited data. *Advances in Neural Information Processing Systems*, 34: 21655–21667, 2021.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data, 2020a. URL https://arxiv.org/abs/2006.06676.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan, 2020b. URL https://arxiv.org/abs/1912.04958.

Kai Katsumata, Duc Minh Vo, Bei Liu, and Hideki Nakayama. Revisiting latent space of gan inversion for robust real image editing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5313–5322, 2024.

Amirhossein Kazerouni, Ehsan Khodapanah Aghdam, Moein Heidari, Reza Azad, Mohsen Fayyaz, Ilker Hacihaliloglu, and Dorit Merhof. Diffusion models in medical imaging: A comprehensive survey. *Medical image analysis*, 88:102846, 2023.

Saeed Khorram, Mingqi Jiang, Mohamad Shahbazi, Mohamad H. Danesh, and Li Fuxin. Taming the tail in class-conditional gans: Knowledge sharing via unconditional training at lower resolutions, 2024. URL https://arxiv.org/abs/2402.17065.

Fei Kong, Jinhao Duan, Lichao Sun, Hao Cheng, Renjing Xu, Hengtao Shen, Xiaofeng Zhu, Xiaoshuang Shi, and Kaidi Xu. Act-diffusion: Efficient adversarial consistency training for one-step diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8890–8899, 2024.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.

Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The role of imagenet classes in fréchet inception distance. *arXiv preprint arXiv:2203.06026*, 2022.

Cheuk Ting Li and Farzan Farnia. Mode-seeking divergences: theory and applications to gans. In *International Conference on Artificial Intelligence and Statistics*, pp. 8321–8350. PMLR, 2023.

Ziqiang Li, Beihao Xia, Jing Zhang, Chaoyue Wang, and Bin Li. A comprehensive survey on data-efficient gans in image generation, 2022. URL https://arxiv.org/abs/2204.08329.

Zinan Lin, Vyas Sekar, and Giulia Fanti. Why spectral normalization stabilizes gans: Analysis and improvements. *Advances in neural information processing systems*, 34:9625–9638, 2021.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pp. 3481–3490. PMLR, 2018.

Takeru Miyato and Masanori Koyama. cgans with projection discriminator, 2018. URL https://arxiv.org/abs/1802.05637.

M-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pp. 8748–8763. PMLR, 2021.

Harsh Rangwani, Naman Jaswani, Tejan Karmali, Varun Jampani, and R Venkatesh Babu. Improving gans for long-tailed data through group spectral regularization. In *European Conference on Computer Vision*, pp. 426–442. Springer, 2022.

Harsh Rangwani, Lavish Bansal, Kartik Sharma, Tejan Karmali, Varun Jampani, and R. Venkatesh Babu. Noisytwins: Class-consistent and diverse image generation through stylegans, 2023. URL https://arxiv.org/abs/2304.05866.

Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. *Advances in neural information processing systems*, 31, 2018.

Divya Saxena, Jiannong Cao, Jiahao Xu, and Tarun Kulshrestha. Rg-gan: Dynamic regenerative pruning for data-efficient generative adversarial networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 4704–4712, 2024.

Divya Saxena, Jiannong Cao, Jiahao Xu, and Tarun Kulshrestha. Re-gan: Data-efficient gans training via architectural reconfiguration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(11):9580–9596, 2025. doi: 10.1109/TPAMI.2025.3590650.

Mohamad Shahbazi, Martin Danelljan, Danda Pani Paudel, and Luc Van Gool. Collapse by conditioning: Training class-conditional gans with limited data. *arXiv preprint arXiv:2201.06578*, 2022.

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. doi: 10.1109/TPAMI.2011.227.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Hung-Yu Tseng, Lu Jiang, Ce Liu, Ming-Hsuan Yang, and Weilong Yang. Regularizing generative adversarial networks under limited data. In *CVPR*, 2021.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. Caltech-ucsd birds-200-2011. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Ceyuan Yang, Yujun Shen, Yinghao Xu, Deli Zhao, Bo Dai, and Bolei Zhou. Improving gans with a dynamic discriminator. *Advances in Neural Information Processing Systems*, 35:15093–15104, 2022.

Mengping Yang and Zhe Wang. Image synthesis under limited data: A survey and taxonomy, 2024. URL https://arxiv.org/abs/2307.16879.

Zhaoyu Zhang, Yang Hua, Guanxiong Sun, Hui Wang, and Seán McLoone. Improving the training of data-efficient gans via quality aware dynamic discriminator rejection sampling. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 30682–30691, 2025.

Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. *Advances in neural information processing systems*, 33:7559–7570, 2020.

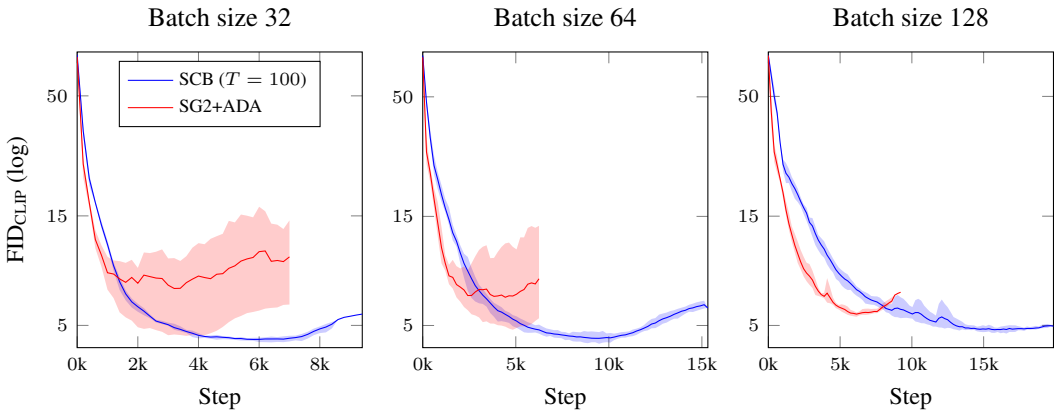


Figure 6: FID_{CLIP} throughout training, for StyleGAN2+ADA with and without single-class batches (with $T = 100$), across three batch sizes. Shaded regions show the min–max range across three seeds, and the thick line is the median.

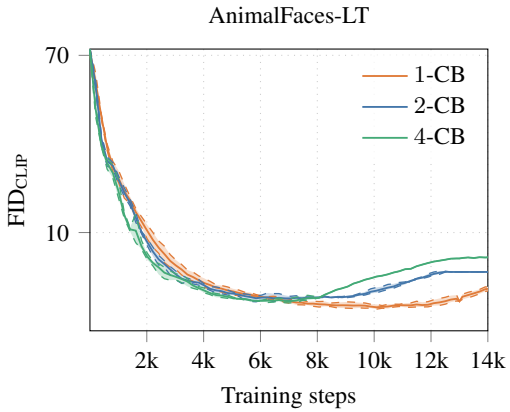


Figure 7: Ablation over the number of buckets in which to split a batch on AnimalFaces-LT. Solid lines show the mean over runs; dashed lines and shaded regions indicate min/max range. Note the log scale.

A Ablation on Batch Size

Figure 6 shows ablations on batch size. Not only does our method improve performance across all batch sizes, but variance across different seeds is much lower.

B Ablation on Split-Class Batches

As mentioned in the main paper, rather than only using single-class batches for classes with a sufficient amount of samples, we considered dividing each batch into equal-size buckets and sampling a class label separately for each bucket. In Figure 7 we present ablations on the number of buckets B in which to split each batch. We use B-CB to denote that each batch was split into B buckets. Note that 1-CB corresponds to sampling a single class for each batch as done in the main paper. In all cases, as done throughout the paper, we sample each class label according to its empirical frequency. We find that having a single class per batch gives the best results.

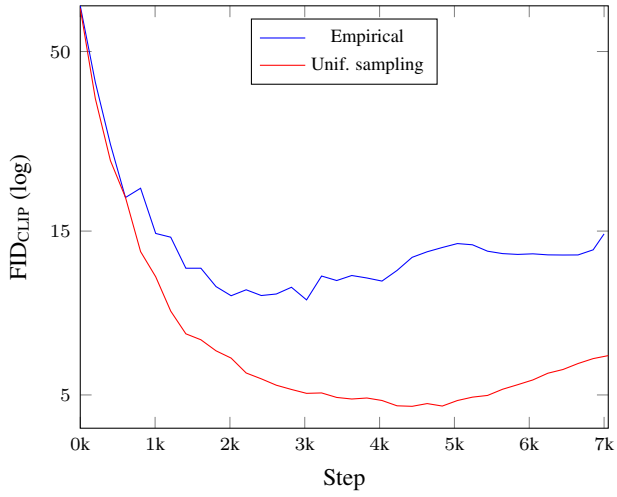


Figure 8: FID$_{CLIP}$ throughout training for a StyleGAN2+ADA, on AnimalFaces-LT, performing class-wise sampling either uniformly or according to the empirical class distribution.

C Uniform vs. Empirical Class Sampling

Figure 8 shows that sampling class labels according to their empirical frequency, rather than uniformly, gives much better results.

D Per-Class FID Breakdown

Figure 9 shows per-class FID$_{CLIP}$ scores for the AnimalFaces-LT dataset. The classes shown in Figure 1 show marked noticeable improvements in FID$_{CLIP}$.

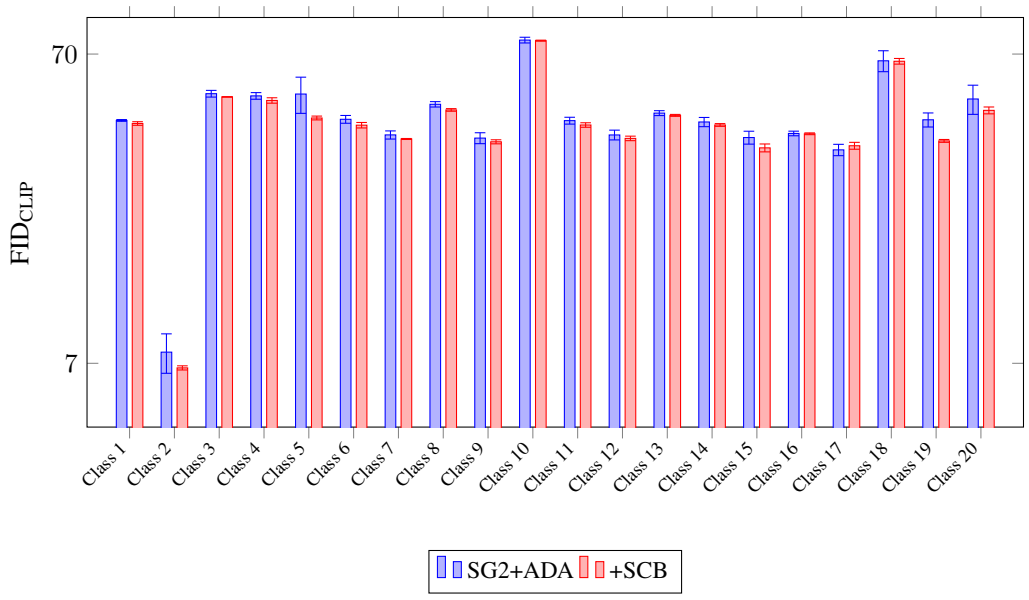


Figure 9: Per-class FID$_{CLIP}$ scores (log scale) across the three variants. The whiskers on each bar correspond to a standard deviation.