Multiclass non-Adversarial Image Synthesis, with Application to Classification from Very Small Sample

By

ITAMAR WINTER

Under the supervision of

PROF. DAPHNA WEINSHALL

THE HEBREW
UNIVERSITY
OF JERUSALEM

Faculty of Computer Science and Engineering
THE HEBREW UNIVERSITY OF JERUSALEM

A dissertation submitted to the Hebrew University of Jerusalem as a partial fulfillment of the requirements of the degree of MASTER OF SCIENCE in the Faculty of Computer Science and Engineering.

DECEMBER 2020

Abstract

The generation of synthetic images is currently being dominated by Generative Adversarial Networks (GANs). Despite their outstanding success in generating realistic looking images, they still suffer from major drawbacks, including an unstable and highly sensitive training procedure, mode-collapse and mode-mixture, and dependency on large training sets. In this work we present a novel non-adversarial generative method - Clustered Optimization of LAtent space (COLA), which overcomes some of the limitations of GANs, and outperforms GANs when training data is scarce. In the full data regime, our method is capable of generating diverse multi-class images with no supervision, surpassing previous non-adversarial methods in terms of image quality and diversity. In the small-data regime, where only a small sample of labeled images is available for training with no access to additional unlabeled data, our results surpass state-of-the-art GAN models trained on the same amount of data. Finally, when utilizing our model to augment small datasets, we surpass the state-of-the-art performance in small-sample classification tasks on challenging datasets, including CIFAR-10, CIFAR-100, STL-10 and Tiny-ImageNet. A theoretical analysis supporting the essence of the method is presented.

no need for keywords

Keywords. Small data, generative models, latent optimization, classification, augmentation

Dedication and acknowledgements

First and foremost, I would like to extend my gratitude to my supervisor, Prof. Daphna Weinshall, for her endless dedication and support. I am grateful for the academic freedom that she has granted me which has allowed me to follow my own interests and curiosities. Needless to say, this achievement wouldn't have been accomplished without her careful guidance.

I would also like thank my friends and colleagues for the countless brainstorming sessions and for consistently lifting my mood. Good friends during research crises cannot be overstated.

The support I have received from my family was also paramount for my success. For that, I thank my mother and father who have undoubtedly affected my decisions in life which have eventually led me to this point.

Last but not least, I must convey my deep appreciation and gratefulness to my love and life partner Rose. She has wholeheartedly supported me through all this long and challenging process, and given me the much needed strength and confidence to pursue my goals.

TABLE OF CONTENTS

List of Tables

Table **Page**

List of Figures

CHAPTER 1

INTRODUCTION

Generative image modeling is a long-standing challenge in computer vision. Unconditional generative models aim at learning the underlying distribution of the data using a finite training set, and synthesizing new samples from the learned distribution. Recently, deep generative models have shown remarkable results in synthesizing high-fidelity and diverse images. Most notably, Generative Adversarial Networks (GANs) [25] have been extensively used in classical computer vision tasks such as image generation, image restoration and domain translation, alongside traditional learning tasks such as data augmentation [22] and clustering [6, 56].

Since their inception, the unsupervised training of GANs achieved effective models able to produce natural-looking images, while relying on a simple and easily modified framework. Nevertheless, and despite numerous efforts for improvement, GANs still exhibit some critical drawbacks that arise from the adversarial nature of the optimization. These include: (i) an unstable training procedure, that is highly sensitive to the choice of initialization, architecture and hyper-parameters; (ii) often the learned distribution suffers from mode-collapse, in which only a subset of the real distribution is covered by the model, or mode-mixture, where different modes are mixed with each other. These problems are amplified when training data is scarce [78].

These drawbacks have motivated research into non-adversarial alternatives such as Variational Auto Encoders (VAE) [38] and Generative Latent Optimization (GLO) [10]. VAEs learn generative deep models that include a representation layer defining the model's latent space, where both the prior and posterior distributions over the

latent space are approximated by parametric Gaussian distributions. GLO learns a non parametric prior over the latent space in unison with the generative model. Although the VAE framework stands on solid theoretical foundations, VAEs generally do not generate sharp images, partially due to the restrictive parametric assumptions that are enforced. GLO, on the other hand, imposes hardly any limitation on the learned distribution over the latent space, which is guided only by the reconstruction performance of the model. Alas, as a result the structure of the latent space holds no semantic information, and cannot be effectively sampled from. These limitations are aggravated when dealing with multi-modal distributed data, as is typically the case with multi-class data.

Broadly speaking, most contemporary generative models rely on common and often implicit assumptions: (i) the Manifold Hypothesis, which assumes that real-world high-dimensional data lie on low-dimensional manifolds embedded within the high-dimensional space; (ii) that there exists a mapping from a low dimension latent space onto the real data manifold; (iii) that this latent space can be approximated by a single Gaussian distribution (such is the latent prior distribution in most variants of GANs, VAEs, and GLO); and (iv) that the generative model is capable of learning the assumed mapping. While these assumptions may hold true when trying to learn from data that resides on a single manifold, it is impossible for a continuous mapping (CNN generator) to effectively map a connected latent space onto a disconnected data manifold of a multi-class distribution [35].

In this work, we seek to overcome both the inherent drawbacks of the GAN framework and the deficiency of the uni-modal Gaussian prior in modeling the latent space. Thus, in Chapter 3 we propose an unsupervised non-adversarial generative model, that optimizes the latent space by fitting a multi-modal data distribution. Unlike GLO, our latent space preserves semantical information about the data, while the multi-modal distribution allows for the efficient and direct sampling of new data. As will be shown in Chapter 4, the distribution over the latent space that is learnt by our model captures semantic properties of the data. As a result, our model is capable of generating better images in terms of image quality, diversity and discriminability. In Chapter 6 we provide some theoretical justification for our method.

Expanding to domains where GANs do not excel, our model is designed to be applicable for downstream tasks where training data is scarce. The task of learning from small sample is usually tackled with the aid of external data or prior knowledge. While transfer-based techniques work well when the source and target domains share distributional similarities, it is not at all the case when the target data comes from a

considerably different domain (such as medical imaging) [51, 60]. Furthermore, gaining access to large labeled datasets may not always be possible due to legal and ethical considerations. In contrast, here we tackle the small-sample classification task where **no prior knowledge or external data is present**. In this setting, the training algorithm may get as few as 5 images per class, having access to no additional labeled or unlabeled data. This constitutes a very challenging task. In Chapter 5 we show that, when using our model to augment the real data, we are able to advance the state-of-the-art and achieve top performance in small sample classification tasks.

1.1 Contributions and Outline

Our main contributions in this study are as follows:

I) Introduce a novel unsupervised non-adversarial generative model capable of synthesizing diverse discriminable images from multi-class distributions (Chapter 3).

II) Demonstrate superior image synthesis capabilities when training data is scarce, as compared to state-of-the-art GAN models (Chapter 4).

III) Apply our model to small-sample classification tasks, surpassing all previous work in this domain (Chapter 5).

IV) Provide sufficient conditions and a simplified theoretical framework, under which our method can be beneficial in approximating under-sampled distributions (Chapter 6).

CHAPTER 2

BACKGROUND & REALTED WORK

In this chapter we offer some background on the core subjects and methods that are related to this work. Additionally, we survey prior work that faced the challenges of modeling a multi modal distribution, learning a low dimensional data embedding, and learning from a small sample.

2.1 Background - Generative Models

In this section we will briefly introduce some of the core generative models that exist to date. While not covering all the methods of the field of generative models, our aim is to provide a clear and simple overview of the approaches that are most relevant to this work.

2.1.1 Auto-Encoder

An autoencoder (AE) is a neural network that consists of an encoder $\mathcal{E}$ that maps data inputs to a low-dimensional latent space, and a decoder $\mathcal{D}$ that maps the latent code back to the data space. The parameters of the encoder and decoder are optimized simultaneously to minimize a reconstruction loss between a real image x and the reconstructed image $\mathcal{D}\big(\mathcal{E}(x)\big)$. Thus, for a standard AE, a "good" embedding is merely one that for each data point x, $\mathcal{E}(x)$ preserves the information that is essential for reconstructing the image. This usually leads to a latent space with no semantic structure, which does

not permit generating novel high-quality images. Nevertheless, while auto encoders exists for more than two decades [46], recent advances have attempted to regularize the training procedure to obtain representations with useful properties. These include sparsity of the representation [61] and robustness to noise or to missing inputs [72], which lead to models that can better learn the data distribution, and in some cases may also permit the sampling of new samples [7].

2.1.2 Generative Adversarial Networks - GAN

Generative Adversarial Networks [25] are a framework for training generative models using two sub-models: a generator model G that is trained to generate new samples from the domain distribution, and a discriminator model D that tries to classify samples as either real (from the domain) or fake (generated). This procedure can be framed, in a game-theoretical sense, as a zero-sum game between the generator and discriminator, where each component has a contradictory goal, giving rise to an optimization process which is 'adversarial' in nature.

More formally, GANs assume a latent variable model, where $P(x) = \int_z P(z)P(x|z)$ and where $P(z)$ has a Gaussain distribution. The optimization process aims at minimizing the cross-entropy classification loss of the discriminator:

$$-\frac{1}{2}\mathbb{E}_{x\sim p_{data}} \log D(x) - \frac{1}{2}\mathbb{E}_z \log(1 - D(G(z))$$

Since their inception, many formulations and improvements for the classic GAN framework have been proposed, resulting in high-capacity models, capable of generating high-fidelity natural looking images. Nevertheless, as will be shown later on, GANs still exhibit some major drawbacks including an unstable training procedure, mode-collapse and a reliance on large datasets.

Most noteworthy, mode-collapse is the term referring to the state where the generator can only cover a small subset of the data distribution, and is usually attributed to the fact that for any fixed discriminator, the dominating strategy for the generator would be to produce few samples that are most probable in the eyes of the discriminator [54]. This phenomena is exhibited in our experiments, and is amplified when GANs are trained with insufficient data.

2.1.3 Variational Auto Encoder - VAE

A Variational Auto Encoder [39] is a latent variable model whose posterior is approximated by a neural network with parameters θ. The optimization process aims at maximizing the liklihood of the data under the model:

$$P(X) = \int P(X|z;\theta)P(z)dz \tag{2.1}$$

This is done by solving the following equation:

$$\log P(X) - D_{KL}\big[Q(z|X) \,\|\, P(z|X)\big] = \mathbb{E}_{z\sim Q}[\log P(X|z)] - D_{KL}\big[Q(z|X) \,\|\, P(z)\big] \tag{2.2}$$

Where D_{KL} is the Kullback-Leibler divergence measure:

$$D_{KL}\big[P \,\|\, Q\big] = \mathbb{E}_{x\sim P} \log\big(\frac{P(x)}{Q(x)}\big)$$

The right Hand side of Eq. 2.2 is termed the Evidence Lower Bound (ELBO) of the log-likelihood, and is optimized using SGD.

In this framework, both the prior $P(z)$ and the posterior $Q(z|x)$ take the form of Gaussian distributions, and $P(x|z)$ and $Q(z|x)$ are approximated using deep neural networks.

2.1.4 Generative Latent Optimization - GLO

GLO [10] is a recently proposed method for generative modeling that optimizes the latent space directly, instead of relying on an encoder. Thus, it can be seen as an encoderless auto-encoder, where the posterior $P(z|x)$ is approximated by means of optimizing z according to some similarity measure between $G(z)$ and x.

Specifically, given a dataset $\mathcal{X} = \{x_1, x_2, ..., x_N\}$, N random noise vectors $\mathcal{Z} = \{z_1, z_2, ..., z_N\}$ are uniformly sampled from the unit sphere in $\mathbb{R}^b$, and matched randomly to the data points, yieldng a matching $\{(x_1, z_1), (x_2, z_2), ..., (x_N, z_N)\}$.

The parameters of a generator function $G_\theta : \mathcal{Z} \longrightarrow \mathcal{X}$ are optimized in conjunction with the learnable noise vectors $\mathcal{Z}$ to obtain:

$$\min_{\theta,\mathcal{Z}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\big(G_\theta(z_i), x_i\big) \quad s.t. \quad ||z_i||_2 = 1$$

where $\mathcal{L}$ is the similarity measure, originally implemented using the Laplacian Pyramid Loss [50] (see Eq. 3.5). Unlike standard Auto-Encoders, it has been shown that this training protocol yields a latent space that exhibits compelling attributes such as meaningful interpolations between samples and linear arithmetic with noise vectors.

2.2 Background - Deep Image Clustering

The ultimate objective of data clustering is to partition the data into distinct groups such that similar data points would be grouped together, while dissimilar points would be separated. This task is very much dependent on the choice of the similarity measure, since different measures may lead to profoundly different clusterings. Classical clustering methods usually use heuristics based on the structure of the data to find the optimal partitioning. These can be roughly divided into density-based methods [41], which cluster the data based on regions of high density; partition-based methods [3], which are based on iterative relocation of data points between clusters, and hierarchical methods [77], which seek to find an hierarchical structure of the clusters.

The unsupervised clustering of images poses additional challenges. Firstly, the high dimension of images preclude the usage of some classical methods due to computational considerations. Secondly, and more importantly, finding a meaningful similarity measure between images is a non-trivial task, since any norm-induced distance (such as $\mathcal{L}_2$) in the pixel-space is usually uninformative. For this reason, the task of clustering images is intertwined with the task of learning a meaningful representation of the data. Consequently, recent image clustering methods have attempted to learn both tasks simultaneously. A representative collection of methods that cluster images according to their respective features extracted from a DNN can be found in Sec. 2.4.2

2.3 Background - Learning from Insufficient Data

The issue of learning from insufficient data relates both to cases where there is insufficient data samples (e.g. learning from small-sample in cases where obtaining data points is hard) or insufficient supervision (e.g. semi-supervised learning [71] where we might have a large body of unlabeled data points and a small amount of labeled ones). Both cases pose critical difficulties to many machine learning methods in general, and deep neural networks in particular. This is generally attributed to the fact that when confronted with insufficient data, deep NNs tend to over-fit the small sample, and avoid generalizing the true data distribution. There exists many techniques to combat situations where there is shortage of data, with the main ideas presented in Section 2.4.3. One of the most prominent aspects in this context includes augmenting the dataset into a larger and richer one, which is also one of the directions taken in our proposed method.

2.3.1 Data Augmentations

Data augmentation is a widely used method for generating additional data to improve many machine learning systems, which will otherwise fail to generalize when trained with insufficient data. Many computer vision tasks rely heavily on hand-crafted image transformation policies that are based on prior knowledge of the domain distribution. This knowledge is then used to compose identity-preserving transformations such as flipping, cropping, translations, rotations and color jitters [68]. Other more advanced methods attempt to find an optimal policy of combinations of augmentations that will best fit the task at hand [8, 17, 48].

Another data augmentation strategy revolves around exploiting generative models for generating new samples that will enhance the training set. While these methods are harder to train and facilitate, they avoid the need for prior knowledge regarding the invariances that are present in the data. A representative collection of such methods is further discussed in Sec. 2.4.3. It should be noted that most of these techniques are beneficial only when external data is available, and that in general, theses models necessitate large training data.

In a broader view, combining a classification model with an image generation model may be seen as multi-task learner that is able to "imagine" new datapoints. In this context, a learner that can successfully generate datapoints from the true distribution can hold an advantage in classifying the data. This slightly resembles the "self-explanation effect" [9] evident in human knowledge acquisition. This effect, which has been consistently demonstrated in real world scenarios, maintains that learners who can explain the underlying relationships and connections of the problem at hand, perform better on the learning task. Similarly, a learning model that can generate a unique image of some object class indicates that the model has learnt to "explain" in some internal way the underlying structure that gives rise to that class.

In this work, we propose a generative method that performs relatively well even when trained on an extremely small-sample, without using external data. As such, it is highly valuable in generating new samples and augmenting small training sets, even when no prior knowledge is present on which transformations are identity-preserving in the dataset.

2.4 Related Work

2.4.1 Modeling disconnected data manifolds.

The issue with mapping a connected latent space onto a disconnected data manifold was mainly addressed in the context of overcoming mode-collapse in GANs. [14, 20, 44, 70] use an encoder to match the latent code with the data distribution. While the latent representation of these methods is optimized via a reconstruction loss of the decoder, our method learns a representation that holds semantical information.

Another line of work [2, 23, 29, 36] uses multiple generators in order to cover all the modes in the data, while An et al. [1], Hoshen et al. [30] learn a mapping from a normally distributed noise to an optimized latent structure in a non-adversarial framework. Other works use a GMM prior over the latent space in VAEs [18, 67] and GANs [6]. Finally, Chen et al. [15] combines discrete and continuous latent factors to learn a disentangled representation of the data.

2.4.2 Data Embedding and Feature Learning.

Learning a meaningful low-dimensional embedding for high-dimensional data has been significantly improved by advances in deep neural networks and self-supervised learning. Thus, [13, 26, 32] all harness the large capacity of deep neural networks to learn efficient clustered representations of natural images. In a related line of work, *self-supervised learning* involves the learning of meaningful visual features from a pretext task using labels that are produced from the data itself with no direct supervision. These include jigsaw puzzle solving [58], predicting positions of patches in an image [19], and predicting image rotations (*'RotNet'*) [24]. In this work, we learn a clustered embedding of the data and a self-supervised pretext task en masse, which greatly improves the quality of the learned representation.

2.4.3 Learning from small sample.

Classification from small sample, with no prior knowledge or access to external data, has been chiefly approached by attempting to augment the sample into a sufficiently large training set. Thus **DADA** [81] adapts a GAN model for this purpose, **TANDA** [62] uses GANs to learn generic data augmentations composed of pre-defined transformations using large unlabeled data, DHN [59] uses a hybrid network that incorporates learnable weights with a scattering network of predefined wavelets, and **CFVAE-DHN** [49]

augments the latent variables of a VAE, which in turn generates additional data that is classified using a DHN. Likewise, **(author?)** [5] promotes the use of the cosine-loss, and **(author?)** [11] promotes low-complexity networks.

Other methods usually incorporate some form of *transfer learning* [83], where parameters that are learned in a source domain are transferred and utilized in a different target domain.

The most prominent research paradigm in this context is *few-shot learning* [75], where there exists a dataset D consisting of sufficiently large labeled classes and a smaller dataset N made of a few small novel classes such that $D \cap N = \emptyset$. A classifier is then trained on $D \cup N$ and evaluated on unseen samples from N. We consider this paradigm to be an instance of transfer (or meta) learning and not strict classification from small sample, as it requires access to an external labeled dataset.

Similar small-sample challenges are usually tackled with the aid of external data or prior knowledge. These can be roughly divided into methods that attempt at adjusting the model training procedures and algorithms such as multi-task learning [12, 82], Embedding Learning [33], and methods that inject prior knowledge into the training dataset using data augmentations.

Transfer learning from large datasets to smaller ones has also been investigated in generative models [57, 73, 74, 76].

CHAPTER 3

OUR METHOD

Our method, Clustered Optimization of LAtent space (COLA), is an unsupervised method which learns a generative model for the synthesis of images. The method is designed to cope with a small training set of natural images, portraying distinct object categories. It involves three steps, the first two of which are illustrated in Fig. 3.1.

3.1 Step I: Clustering the latent space

The goal here is to deliver a mapping from the data space to a latent space, while clustering the mapped points into compact K clusters, see illustration in Fig. 3.1. To this end, we seek a clustering algorithm capable of semantically grouping the images, such that images from the same class will reside in proximity in the latent space, and dissimilar images will be locatd further apart. One such algorithm is the Multi-Modal Deep Clustering (MMDC) algorithm [65]. In this algorithm, a deep convolutional network E_θ, is trained to map each data point to a fixed low-dimensional target point in the latent space - the unit sphere in $\mathbb{R}^K$. The target is sampled from a pre-defined distribution over the latent space.

Specifically, given an unlabeled dataset $\mathcal{X} = \{x_i\}_{i=1}^N$, the model is initialized with some random assignment $\{(x_i, t_i)\}_{i=1}^N$, where each $t_i \in \mathbb{R}^K$ is sampled from a GMM distribution with K-components, and normalized to length 1. Training involves the minimization of

$$\|E_\theta(x_i) - t_{\pi(i)}\|_2^2 \tag{3.1}$$

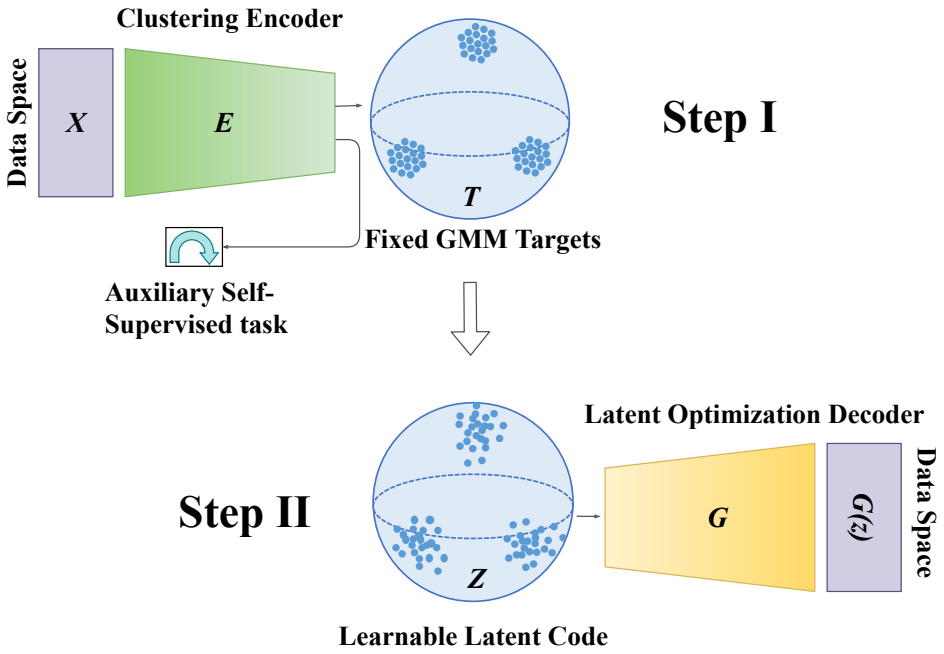


Figure 3.1: Illustration of our model: In step I images are mapped to fixed low-dimensional targets T. In step II these targets form a latent space Z that is trained in conjunction with the generator parameters to reconstruct the original image.

over the assignment $\{(x_i, t_{\pi(i)})\}_{i=1}^{N}$ and parameters θ.

This optimization problem is solved with SGD, and involves two steps per mini-batch. First, the sample $\{x_i\}_{i \in b}$ is mapped onto the latent space. The assignment problem for $\{(x_i, t_{\pi(i)}) \mid i \in b\}$ is solved using the Hungarian Algorithm [43] applied to the following problem:

$$\pi^* = \underset{\pi : b \leftrightarrow b}{\arg\min} \sum_{i \in b} \|E_\theta(x_i) - t_{\pi(i)}\|_2^2 \tag{3.2}$$

Subsequently π^* is inserted into Eq. 3.1, which is then optimized w.r.t θ. This method is enhanced with self-supervision based on the auxiliary *'RotNet'* task [24], and consistency regularization where augmented images are mapped to the same cluster.

The output of this model constitutes a latent space, where the representations of semantically similar images reside in proximity, and images from distinct classes are located further apart. This representation is used to initialize the latent space of the generative model in step II (Sec. 3.2). To simplify the presentation, henceforth we let $t_i = t_{\pi^*(i)}$ denote the final target associated with x_i.

3.2 Step II: Image generation

Given a matching between data points $\mathcal{X}$ and targets $\mathcal{T}$ - $\{(x_i, t_i)\}_{i=1}^N$, a latent code $\mathcal{Z} = \{z_i\}_{i=1}^N$ is constructed such that

$$z_i = (\frac{t_i}{\|t_i\|_2}, \frac{v_i}{\|v_i\|_2}) \in \mathbb{R}^{K+d}$$

Above $v_i \sim \mathcal{N}(\vec{0}, \sigma I_{d\times d})$ denotes an additional source of variation, and t_i denotes the class-component of the code.

The parameters of a CNN generator function $G_\theta : \mathcal{Z} \longrightarrow \mathcal{X}$ are optimized in conjunction with the learnable representation vectors $\mathcal{Z}$, as illustrated in Fig. 3.1. The optimization problem is defined as:

$$\min_{\theta, \mathcal{Z}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{rec}\big(G_\theta(z_i), x_i\big) \quad s.t. \quad \|z_i\|_2 = 1 \tag{3.3}$$

where $\mathcal{L}_{rec}$ denotes the reconstruction loss between the original image x_i and the image generated by the model $G_\theta(z_i)$.

As shown in [30], the best image quality for this kind of models may be obtained when $\mathcal{L}_{rec}$ is realized with the perceptual loss [34]:

$$\mathcal{L}_{vgg}(x, x') = |x - x'| + \sum_{layers:i}^{k} |l_i(x) - l_i(x')| \tag{3.4}$$

In (3.4) l_i denotes the perceptual layer in a pre-trained VGG network [69]. Nevertheless, since external data cannot be used in the small sample scenario adopted here, $\mathcal{L}_{rec}$ is realized in our method with the Laplacian Pyramid loss:

$$\mathcal{L}_{lap}(x, x') = |x - x'| + \gamma \sum_{i}^{k} 2^{-2i} |L_i(x) - L_i(x')| \tag{3.5}$$

In (3.5) $L_i(x)$ denotes the i-th level of the Laplacian pyramid representation of x [50]. The sum of differences is weighted to preserve the high-frequencies of the original image. The components of the representation vectors are normalized after each epoch to length 1, projecting them back to the unit spheres in $\mathbb{R}^K, \mathbb{R}^d$ respectively.

This step is summarized below in Alg. 1. Full implementation details are presented in Appendix C.2.

Algorithm 1 : Training the Generative Model

INPUT:
matched pairs $\{(x_i, t_i)\}_{i=1}^{N} \subset X \times [0,1]^K$ from step I
G_θ - CNN Generator with parameters θ
$\lambda_e^\theta, \lambda_e^z$ - learning rate at epoch e of θ, Z
σ - pre-defined latent std

for *i=1...N* **do** ▷ initialize latent space
  sample $v_i \sim \mathcal{N}(\vec{0}, \sigma I_{d\times d})$
  $z_i \leftarrow (\frac{t_i}{\|t_i\|_2}, \frac{v_i}{\|v_i\|_2}) \in \mathbb{R}^{K+d}$
end for
for *e=1...epochs* **do**
  for *i=1...iters* **do**
    sample batch $\{(x_i, z_i) | i \in B\}$

$$\mathcal{L}_B = \frac{1}{|B|} \sum_{i \in B} \mathcal{L}_{rec}(x_i, G_\theta(z_i))$$

    $\theta \leftarrow \theta - \lambda_e^\theta (\nabla_\theta \mathcal{L}_B)$
    $z \leftarrow z - \lambda_e^z (\nabla_z \mathcal{L}_B)$
  end for
    $t \leftarrow z_{[1:K]}, \quad v \leftarrow z_{[K+1:K+d]}$
$\forall_i$ $z_i \leftarrow (\frac{t_i}{\|t_i\|_2}, \frac{v_i}{\|v_i\|_2})$ ▷ Normalize inputs

end for

3.3 Step III: posterior distribution over the latent space

After training, a posterior distribution over the latent space is obtained by fitting a unique multivariate Gaussian to each cluster in the latent space. Sampling is then performed from the uniform mixture of these Gaussian distributions.

3.4 sCOLA: Supervised Algorithm

In the supervised framework, we have a labeled dataset with K classes $\mathcal{X} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$, where $y_i \in [K]$ denotes the class label of x_i, and e_y^i denotes the one-hot representation of the labels. The supervised version of our method, *sCOLA* includes steps II and III of COLA. The clustering in step I is replaced by the supervision labels from the training data, where each t_i is replaced by the corresponding

e^i_y. Fig. 3.2 shows images generated by our model with only 5 training examples per class.

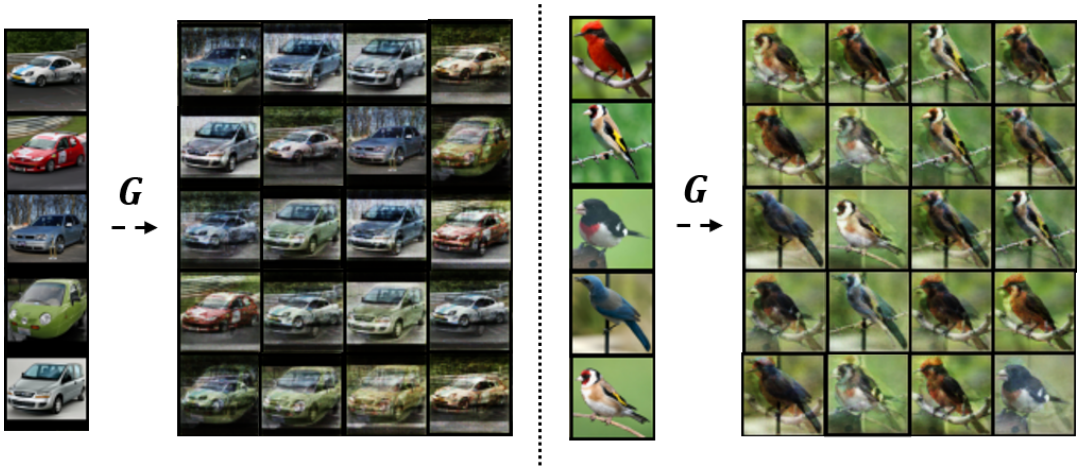


Figure 3.2: Synthetic images generated by our model when trained on STL-10 with 5 images per class and no external data. Real images are shown on the left, synthetic images are shown on the right.

CHAPTER 4

IMAGE GENERATION, LARGE AND SMALL SAMPLE

In this chapter we will demonstrate the capability of our model to produce diverse and discriminable images, employing evaluation metrics that quantify these attributes. Firstly, we compare our model with competitive conditional GAN models that use large and computationally heavy architectures. While these models maintain superiority on large datasets, this dominance diminishes as the sample size drops. Secondly, we show that our unsupervised variant surpasses other unsupervised generative adversarial models using the same architecture. Lastly, we show that our model consistently outperforms other non-adversarial methods in terms of image quality and diversity, regardless of sample size.

4.1 Methodology

Evaluation scores. Designing meaningful quantitative evaluation measures for generative models is a challenging ongoing research area. Presently, two scores seem to dominate the field: the Inception Score [64], and the Fréchet Inception Distance (FID) [28].

The Inception Score measures the average KL divergence between the conditional label distribution $p(y|x)$ (estimated by the Inception model trained on 'ImageNet') and the marginal distribution $p(y)$ obtained from all the samples.

$$IS(\mathcal{X},\mathcal{Y}) = \exp\Big(\mathbb{E}_x\big[D_{KL}\big[p(y|x)\,\|\,p(y)\big]\big]\Big) = \exp\big(H(y) - \mathbb{E}_x[H(y|x)]\big) \tag{4.1}$$

Where p(y) is the empirical marginal distribution

$$p(y) \approx \frac{1}{N} \sum_{i=1}^{N} p(y|x_i = G(z_i))$$

and H(X) is the entropy of X:

$$H(X) = -\mathbb{E}_x \log(x)$$

FID compares the statistics of activations in the penultimate layer of the Inception network (trained on 'ImageNet') between real and generated images. First, the mean and co-variance for both the generated data (μ_g, Σ_g) and the real data (μ_r, Σ_r) are estimated, and then the Fréchet distance between these two Gaussians (a.k.a Wasserstein-2 distance) is then used as a quality measure of the generated images.

$$FID(X_r, X_g) = \|\mu_r - \mu_g\|_2^2 + Tr\left(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^2\right) \tag{4.2}$$

Despite their popularity, there are some major drawbacks with using these methods:

1. Both scores try to capture image quality and diversity on a single scale, and therefore cannot distinguish between the two factors.

2. Both scores are based on the Inception network that was trained on 1,000 classes of ImageNet, and may not be suitable for significantly different datasets [4].

In addition, the IS score does not take into account the real data distribution, and merely measures the diversity of the generated images. It also cannot capture Intra-class diversity. For this reason, we will not be using this metric in this work.

The FID score solves some of the problems associated with IS, by being more consistent with human judgment, being more robust to noise, and being able to detect mode-dropping. Nevertheless, FID still suffers from the two problems outlined above. In addition, it has been shown to be biased and sensitive to the sample size, and in some cases it may not capture mode-mixture. In Appendix A we show that the FID score also fails to reveal intra-class diversity, making it less useful for multi-class datasets (see also [52]). Implementation details for the FID score used in our experiments can be found in Appendix C.4.

Given the problems discussed above, we seek an additional score that can reliably measure how well the generated images fit the true distribution of the data. More importantly, considering that generative models are commonly used in down-stream tasks, we seek a score that can measure the usefulness of the model generations in such tasks. To this end we adopt the scores proposed in [66] ('GAN-Train') and [63]

('CAS'), which are based on training a classification network on the generated images, and evaluating it on real images. The classification accuracy of this network forms an implicit measure of the recall and precision of the generated dataset, since it can only achieve a high score if the synthetic data is sufficiently diverse and discriminable. In our experiments, we follow the protocol defined in [63].

Generative methods used for comparisons. We compare our model against state-of-the-art generative models, one adversarial model based on the GAN framework, and a second non-adversarial method:

1. Adversarial **CGAN-PD** [55]: a conditional GAN with Projection-Discriminator, trained and implemented in accordance with [47].
2. Non-adversarial **GLO** [10]: the original model augmented with the superior perceptual loss from Eq. 3.4. Similarly to step III above, after training we fit a Gaussian Mixture Model to the learned latent space.

Implementation details can be found in Appendix C.2.

Datasets The datasets we use are included in Table 4.1.

Name	**Classes**	**SPC Train/Test**	**Dimension**
CIFAR-10 [42]	10	5000 / 1000	$32 \times 32 \times 3$
CIFAR-100 [42]	100	500 / 100	$32 \times 32 \times 3$
STL-10 [1] (downsampled, labeled only) [16]	10	500 / 800	$48 \times 48 \times 3$
Tiny ImageNet [45]	200	500 / 50	$64 \times 64 \times 3$

Table 4.1: Datasets used in our experiments.

[1] Unlike most generative models trained on STL-10, in this work we only use the labeled images, and discard the 100K unlabeled images.

4.2 Emperical Results

Unsupervised. In the unsupervised scenario, we compare our model to the baseline GLO model, see Fig. 4.1. Clearly our model outperforms GLO on all datasets and metrics, and produces significantly better looking images as demonstrated in Fig. 4.3. Furthermore, we recall that different GAN models can reach similar FID scores if given a high enough computational budget [53]. We therefore adopt the fair comparison protocol proposed in [53], where the architectures of all the models are fixed to the one used in 'InfoGAN' [15], and all models possess the same computational budget for hyper-parameter search. In this protocol, our method outperforms all GAN variants and is on par with the state-of-the-art non-adversarial methods, see Fig. 4.2.

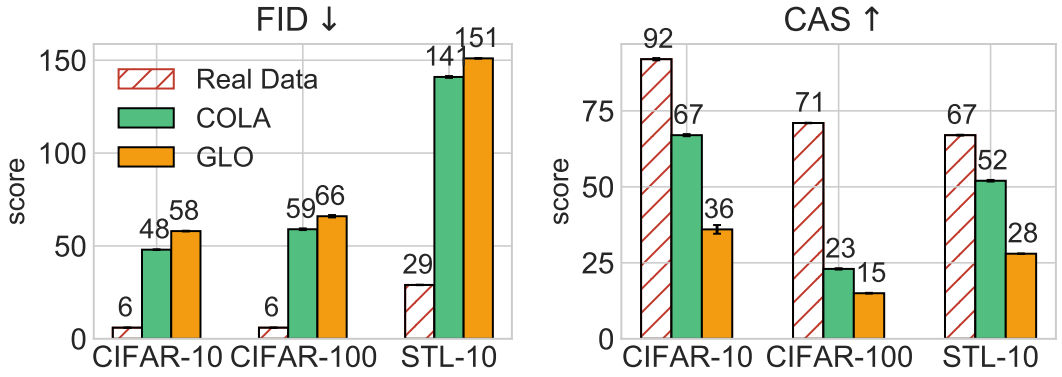


Figure 4.1: Comparison between GLO and COLA using the FID (left) and CAS (right) scores. Our model shows a clear advantage in all cases.

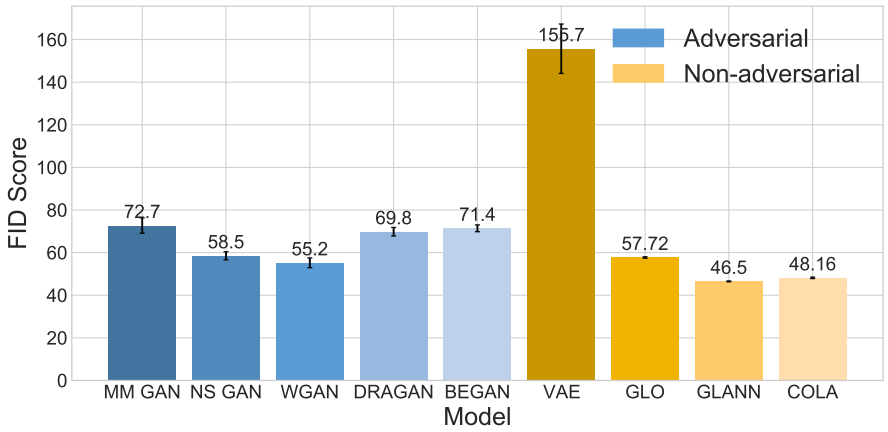


Figure 4.2: FID score computed for CIFAR-10, when all models share the same architecture of 'InfoGAN' [15]. Unlike all other models in this comparison, our method allows for the sampling of images from different individual classes.

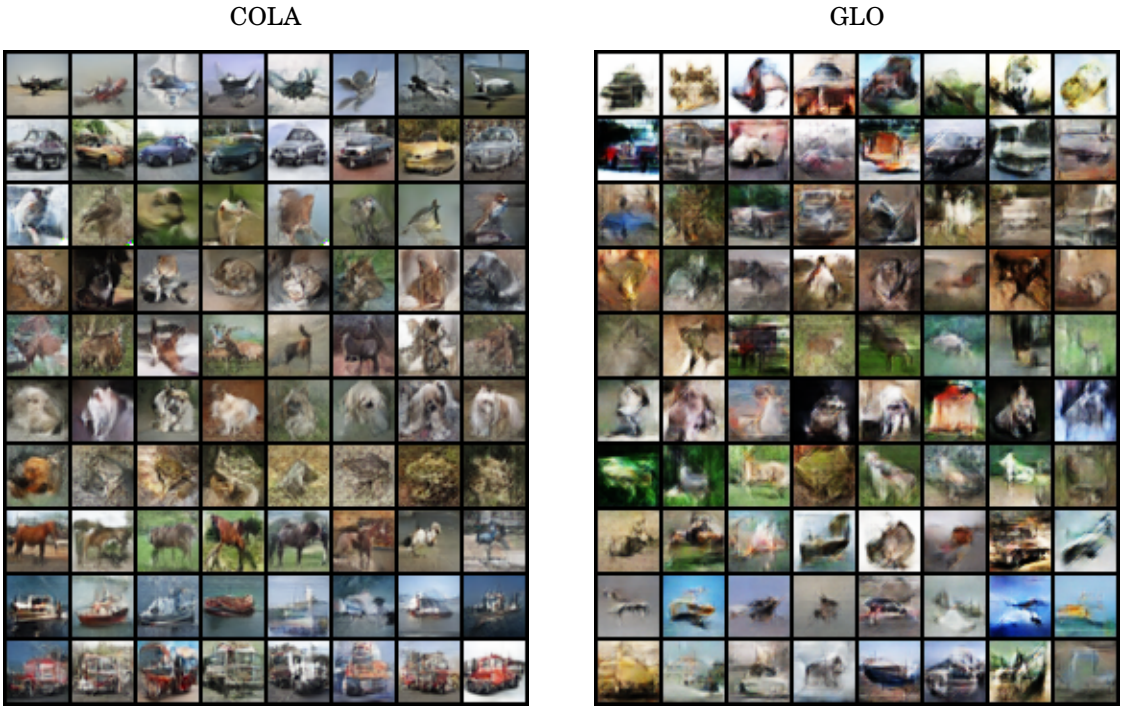


Figure 4.3: Training on CIFAR-10 with no labels: the images generated by our method (left), which imposes semantic structure on the latent space, are superior to the alternative method (right). Each row holds a random sample from a distinct object class.

Supervised. When learning from fully labeled datasets, we evaluate our model against the state-of-the-art conditional GAN variant CGAN-PD with varying sample sizes. Results are shown in Fig. 4.5.

Although conditional GANs obtain better FID scores on large datasets, their performance deteriorates rapidly when training size decreases. Furthermore, our model outperforms GANs when consulting the CAS score on almost all configurations. A qualitative comparison presented in Fig. 4.4 and in Fig. 4 in the Appendix suggests that this deterioration may be attributed to the mode-collapse manifested in CGAN when trained with insufficient data. In contrast, in the extreme small sample regime the images synthesized by our model can hardly be distinguished from real images by both scores, suggesting superior generalization ability in this regime.

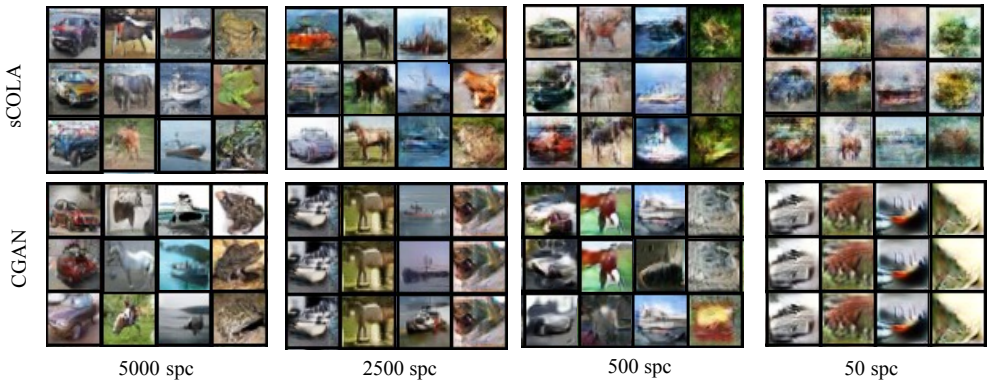


Figure 4.4: qualitative comparison between sCOLA (top) and CGAN (bottom) trained on CIFAR-10 with varying numbers of samples per class (spc). Each column corresponds to a different class in the data. CGAN evidently suffers from mode-collapse when given insufficient data for training.

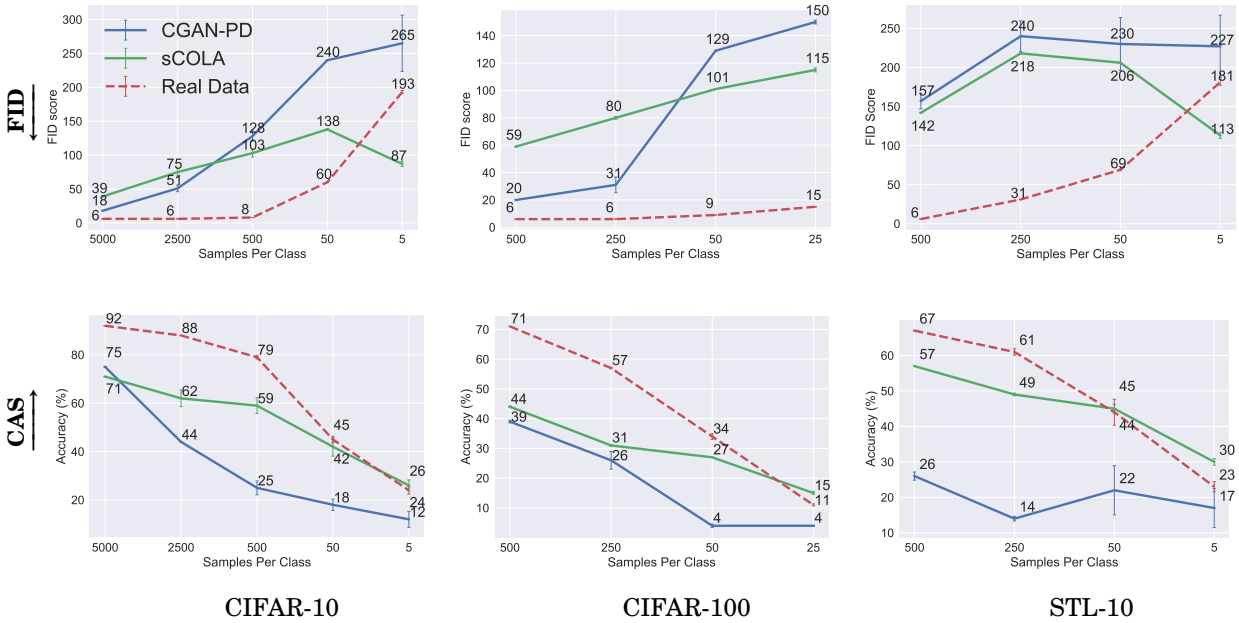


Figure 4.5: FID (top) and CAS (bottom) scores on CIFAR-10, CIFAR-100 and STL-10 with varying training sample sizes. sCOLA's generated images achieve better scores than the GAN's images in the small sample regime, and even achieve better scores than real images when data is extremely scarce (see also Chapter 5).

4.3 Ablation Study

4.3.1 Generator Architecture

We tested many commonly used generative network architectures, and assesed their impact on the model's performance. The models that were evaluated are thus:

1. ***InfoGAN*** with transposed convolutions
2. ***DCGAN*** with residual blocks and upscaling convolutions
3. ***CGAN*** with residual blocks, upscaling convolutions and conditional Batch-Norm

Note that we use the GAN models name for reference only, as our method uses these architectures in a non-adversarial approach, with no discriminator. The differences of the above are summarized in Table 4.2, and the results are given in Table 4.3. Implementation of all of the above was conducted according to [47].

Arch	# Params	Residual	Upscaling	Batch-Norm
InfoGAN	8.6M	✗	transposed convolution	none
DCGAN	4.1M	✓	bilinear upsampling	global
CGAN	4.1M	✓	bilinear upsampling	conditional

Table 4.2: Design differences between evaluated architectures

Architecture	FID ↓	CAS ↑	CAS-Test
InfoGAN	49.38 ±.32	67.65 ±.26	85.14 ±.66
DCGAN	85.94 ±2.14	61.48 ±.23	45.14 ±.74
CGAN	**39.49** ±.20	**70.66** ±.91	85.61 ±.46

Table 4.3: Results were obtained on sCOLA trained on the full train set of CIFAR-10. The conditional batch norm had a notable effect on the quality of the model's generations.

CHAPTER 5

CLASSIFICATION FROM SMALL SAMPLE

In this chpater we show the benefits of using our method in the small sample regime, where only a small sample is available to train the classifier, and **no external information can be used**. We will show that using our model to augment the small training set significantly improves the performance of a deep network classifier trained on this data.

Classification approach. sCOLA is first trained on the small training sample, and then used to generate novel samples from each class. The synthetic images are then combined with the real images, resulting in an extended training set (termed "Mix") that consists of 50% real images, and 50% synthetic images generated by our model. This extended set is then used to train a CNN classifier. For comparison, we train the same CNN classifier with the original images, making sure that both methods see the same subset of images with an identical training procedure.

5.1 Methodology

The datasets we use are described in Table 4.1. For each dataset we train our method with various sample sizes, ranging from 100 samples per class (spc) to as low as 5 spc. For each sample size we run our model on 3 random samples of the same size, and evaluate the classifier's accuracy on the original held-out test set of the data. In order to isolate the contribution of our approach from other factors, we fix the classifier's architecture to

an off-the-shelf ResNet-20 [27] for all datasets except for Tiny ImageNet, which, due to its larger size, resolution and number of classes, necessitates the use of a larger network. Consequently we use the same WRN-16-8 [79] network as CFVAE-DHN (excluding DHN initialization). Full implementation details can be found in Appendix C.3.

5.2 Empirical Results

We compare our model trained on CIFAR-10 and Tiny ImageNet with the best published results reported in [49]. A short description of these methods can be found in Section 2.4.3. The results are summarized in Table 5.1. Our method achieves the best results across all sample sizes on both datasets.

	CIFAR-10					**Tiny ImageNet**			
	100	50	20	10	5	100	50	20	10
DADA	48.32 ±.23	40.48 ±.57	30.44 ±.37	21.67 ±.58	-	17.64 ±.82	14.97 ±1.08	10.13 ±2.04	-
Tanda	45.17 ±1.84	39.16 ±1.18	29.84 ±1.23	20.18 ±.73	-	27.07 ±.94	17.95 ±.59	13.92 ±.59	-
CFVAE-DHN	55.58 ±.12	52.06 ±.36	32.65 ±.38	34.11 ±.67	-	**35.97** ±.35	28.82 ±.79	21.37 ±.29	-
sCOLA	**58.59** ±0.58	**54.51** ±0.22	**49.63** ±1.29	**42.86** ±2.04	**29.05** ±1.09	**35.24** ±.34	**29.70** ±.05	**23.99** ±.52	**17.14** ±.27

Table 5.1: Classification accuracy for **CIFAR-10** (left) and **Tiny ImageNet** (right). Each column corresponds to a different sample size per class. The architecture used by our method is smaller or similar to the ones used by the other methods (see methodology).

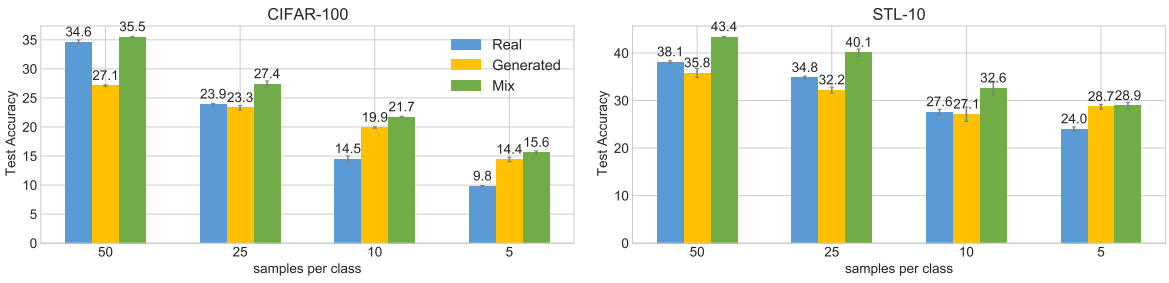


Figure 5.1: Classification accuracy for CIFAR-100 (left) and STL-10 (right) with varying sample size per class.

Additionally, we expand our experiments to datasets with no published results to date on small sample classification tasks. For these datasets, we show that when a classifier

is trained on a mixed dataset consisting of real and synthetic images, it yields better results compared to those obtained when being trained only on the real images or only on the synthetic images. This suggests that our model succeeds in learning the data distribution well enough, and can subsequently generate novel samples that do not exist in the real data. Fig. 5.1 shows results on CIFAR-100 and STL-10.

Furthermore, it is noteworthy that in the extreme small sample scenario, where there are only a handful of images in each class, a classifier trained on the generated dataset performs better than one trained on the real images. This phenomena corresponds with our findings in Section 4.2 where we demonstrated that our model's generated images achieve a better score than real images on the CAS and FID measures when dealing with extremely small data. This can be attributed to te fact that our model is capable of transferring visual features between images, resulting in a virtually richer dataset, which is beneficial for generalizing the true distribution.

5.3 Ablation Study

5.3.1 Image similarity measure

Many generative models are trained using some form of reconstruction loss that is based on a similarity measure between the original image and the one reconstructed by the model. Since using the Euclidean distance in pixel space is highly inadequate (similar objects may differ drastically in pixel space) alternative similarity measures that capture the perceptual relationship between images have been thoroughly investigated in recent years. In this context, it has become a common practice to use a perceptual loss based on a VGG network that was pre-trained on ImageNet for image synthesis tasks [21, 30, 34]. Later works [80] have evaluated numerous similarity measures that are based on deep features of neural nets and concluded that they exhibit a strong correlation with human judgment even when these features where obtained in an unsupervised or self-supervised manner. Nevertheless, establishing a similarity measure without large amounts of data remains to this date an uncharted territory. In order to make our model applicable in the small-data regime, we seek a meaningful similarity measure that uses little to no data. Furthermore, while most works in this area have evaluated the similarity measure according to human judgment on image quality, we sought a measure that will also prove useful in downstream tasks. To this end, we have investigated various methods and evaluated them on the downstream task of image classification of the generated images. Results on partitions of size 1% of CIFAR-10 can be found in Fig. 5.2. We found that the Laplacian Pyramid Loss used in [10] yields the best results compared to other unsupervised measures. The loss functions that were compared are as follows:

1. **ImageNet VGG16** - The original perceptual loss described in Eq. 3.4
2. **Laplacian Pyramid** - The Laplacian Pyramid Loss as described in Eq. 3.5
3. **k-means ResNet32** - Perceptual distance based on layers of ResNet32 initialized with stacked k-means as described in [40]
4. **Random ResNet32** - Perceptual distance based on layers of ResNet32 initialized with random weights
5. **CIFAR-10 ResNet32** - Perceptual distance based on layers of ResNet32 trained on CIFAR-10
6. **L1** - The $\mathcal{L}_1$ loss in pixel space

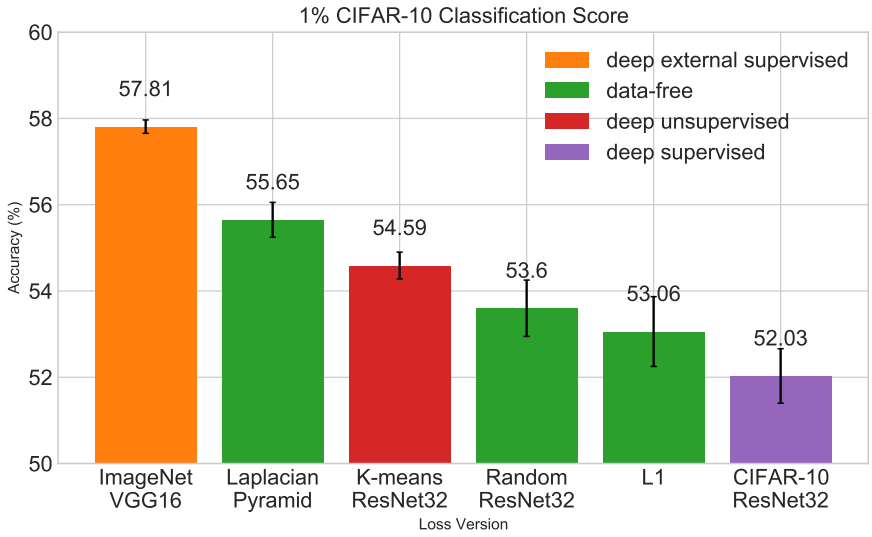


Figure 5.2: The effect of different similarity measures in the optimization of sCOLA when trained on on 1% of the images in CIFAR-10. Classification score is obtained using the same framework described in Section C.3 in the appendix.

5.3.2 Small Data Classifier Architecture.

We experimented with various architectures to check whether larger networks are preferable over smaller ones in the small-sample regime. Our results, depicted in Table 5.2, suggest that there is no notable advantage in using deeper and larger nets, and that smaller nets seem to perform just as well, with the added benefit of shorter training time.

Architecture	# Params	Accuracy
ResNet-20	0.27M	59.03 ±.58
ResNet-32	0.46M	58.4 ±.62
WRN-28-10	36.5M	59.6 ±.99

Table 5.2: Classification accuracy of different networks on a mixed dataset of images generated by our model and real images from CIFAR-10 with 100 samples per class

CHAPTER 6

THEORETICAL ANALYSIS

Stripped off its technical details, the method in Chapter 3 essentially learns a noisy surrogate distribution Z to approximate the real data distribution X and generate new data. In this work, our ultimate goal is not to generate new high quality data, but rather to estimate some function $f : X \longrightarrow \Omega$ from a sample of X. When X denotes data sampled from K discrete classes, a multi-class classifier is such a function whose codomain is either $[K]$ or $\mathbb{R}^K$. If the sample of X is too small, the surrogate distribution Z can be used to generate more data and improve the estimation of f. The analysis below identifies sufficient conditions on the respective sample sizes, such that improvement can indeed be guaranteed.

6.1 Sample Size Analysis

Notations. Assume an i.i.d. sample of random variable pairs - $\{X_i, Y_i\}_{i=1}^N$, where $X_i/_{Y_i=k} \overset{iid}{\sim} \mathcal{D}_k$ and $\mathcal{D}_k$ denotes the class conditional distributions of variable X. Let $\mathcal{X}_k$ denote the conditional sub-sample of datapoints from class k: $\mathcal{X}_k = \{X_{i_j}, Y_{i_j}/_{Y_{i_j}=k}\}_{j=1}^{m_k}$, where $\sum_{k=1}^K m_k = N$.

For simplicity, we will assume in our analysis that f depends only on the expected value of the conditional distributions $\{\mathcal{D}_k\}_{k=1}^K$, denoted μ_k (section 6.2 bridges the gap between the theoretical assumptions and the empirical findings, highlighting when this assumption is indeed reasonable) . Let $\tilde{\mu}_k(X)$ denote an estimator of μ_k from an iid sample of random variable X. Our task is to obtain a set of good estimators $\{\tilde{\mu}_k\}_{k=1}^K$. In

order to simplify the notations, we shall henceforth drop the class index k, with the understanding that the following analysis does not depend on k.

In accordance, let $\mathcal{X}^m$ denote an *iid* sample of size m from the real conditional distribution X of some class k, and $\mathcal{Z}^n$ denote an *iid* sample of size n from the class surrogate distribution Z. Let $\mu_x = \mu$ denote the expected value of X, and μ_z denote the expected value of Z, where $|\mu_x - \mu_z| = d$. Let $\bar{\mathcal{X}}^m$ and $\bar{\mathcal{Z}}^n$ denote the population means of the two samples respectively. Recall that $Var[\bar{\mathcal{X}}^m] = \frac{Var[X]}{m}$ and $Var[\bar{\mathcal{Z}}^n] = \frac{Var[Z]}{n}$ (see Lemma 1).

As customary, we use the population mean of each sample to estimate the unknown distribution's mean μ. Accordingly:

$$\begin{aligned} \tilde{\mu}(X) &= \bar{\mathcal{X}}^m \\ \tilde{\mu}(Z) &= \bar{\mathcal{Z}}^n \end{aligned} \tag{6.1}$$

The error of the two estimators is measured as follows:

$$\begin{aligned} Err(X) &= (\bar{\mathcal{X}}^m - \mu)^2 \\ Err(Z) &= (\bar{\mathcal{Z}}^n - \mu)^2 \end{aligned} \tag{6.2}$$

Proposition 1. *If* $Var[X] > md^2$, *then*

$$n \geq \frac{mVar[Z]}{Var[X] - md^2} \implies \mathbb{E}[Err(Z)] \leq \mathbb{E}[Err(X)]$$

Proof.

$$\begin{aligned} \mathbb{E}[Err(Z)] &= \mathbb{E}[(\bar{\mathcal{Z}}^n - \mu)^2] \qquad \text{(use } \mu = \mu_x\text{)} \\ &\leq \mathbb{E}[(\bar{\mathcal{Z}}^n - \mu_z)^2] + d^2 = \frac{Var[Z]}{n} + d^2 \end{aligned}$$

As long as $Var[X] > md^2$

$$n \geq \frac{mVar[Z]}{Var[X] - md^2} \implies \frac{Var[Z]}{n} + d^2 \leq \frac{Var[X]}{m}$$

and therefore

$$\mathbb{E}[Err(Z)] \leq Var[\bar{\mathcal{X}}^m] = \mathbb{E}[Err(X)]$$

■

Corollary 1.1. *For each class k, if the sample of the surrogate random variable Z is sufficiently large*

$$n \geq \frac{mVar[Z]}{Var[X] - md^2}$$

then the estimator of classifier f obtained from $\mathcal{Z}^n$ is more accurate than the estimator obtained from $\mathcal{X}^m$.

Proposition 2. *Assume that $Pr[0 \leq X, Z \leq 1] = 1$, which can be achieved by dataset normalizing. Then $\forall \epsilon > d$, if $n \geq m(\frac{\epsilon}{\epsilon - d})^2$, then the bound obtained by the Hoeffding's inequality on $Pr(|Err(Z)| \geq \epsilon)$ is tighter than the corresponding bound on $Pr(|Err(X)| \geq \epsilon)$.*

Proof. We invoke the Hoeffding's inequality:

$$Pr(|\bar{\mathcal{X}}^m - \mu| > \epsilon) \leq 2e^{-2m\epsilon^2}$$

and note that

$$|Err(Z)| \leq |\bar{\mathcal{Z}}^n - \mu_z| + |\mu_x - \mu_z| = |\bar{\mathcal{Z}}^n - \mu_z| + d$$

It follows that

$$\begin{aligned} Pr(|Err(Z)| \geq \epsilon) &\leq Pr(|\bar{\mathcal{Z}}^n - \mu_z| + d \geq \epsilon) \\ &= Pr(|\bar{\mathcal{Z}}^n - \mu_z| \geq \epsilon - d) \\ &\leq 2e^{-2n(\epsilon-d)^2} := B(Z) \\ Pr(|Err(X)| \geq \epsilon) &= Pr(|\bar{\mathcal{X}}^m - \mu_x| \geq \epsilon) \\ &\leq 2e^{-2m\epsilon^2} := B(X) \end{aligned}$$

Finally

$$n \geq m(\frac{\epsilon}{\epsilon - d})^2 \implies B(Z) \leq B(X)$$

■

Corollary 2.1. *For each class k, if the sample from the surrogate random variable Z is sufficiently large*

$$n \geq m(\frac{\epsilon}{\epsilon - d})^2,$$

then the estimator of classifier f obtained from $\mathcal{Z}^n$ is more confident than the estimator obtained from $\mathcal{X}^m$.

Lemma 1.

$$Var[\bar{\mathcal{X}}^m] = \frac{Var[X]}{m}$$

Proof.

$$\begin{aligned}
Var[\bar{\mathcal{X}}^m] &= \mathbb{E}[(\bar{\mathcal{X}}^m - \mathbb{E}[\bar{\mathcal{X}}^m])^2] \\
&= \mathbb{E}[(\bar{\mathcal{X}}^m - \mu_x)^2] = \mathbb{E}[(\bar{\mathcal{X}}^m)^2] - \mu_x^2 \\
\mathbb{E}[(\bar{\mathcal{X}}^m)^2] &= \mathbb{E}[(\frac{1}{m}\sum_i^m x_i)^2] \\
&= \frac{1}{m^2}\mathbb{E}[2\sum_i^m \sum_{j>i} x_i x_j + \sum_i^m x_i^2] \\
&= \frac{1}{m^2}\Big(m(m-1)\mathbb{E}[x_i x_j] + m\mathbb{E}[x^2]\Big) \\
&= \frac{1}{m^2}\Big(m(m-1)\mu_x^2 + m\mathbb{E}[x^2]\Big) \\
&= \mu_x^2 - \frac{\mu_x^2}{m} + \frac{\mathbb{E}[x^2]}{m} \\
=> Var[\bar{\mathcal{X}}^m] &= \mathbb{E}[(\bar{\mathcal{X}}^m)^2] - \mu_x^2 \\
&= \frac{\mathbb{E}[x^2] - \mu_x^2}{m} = \frac{Var[X]}{m}
\end{aligned}$$

■

6.2 Bridging the Gap between Theory and Practice

In this section we show that reducing f to estimating μ_k is justified, and conforms with our empirical settings. We will show that a Cross-Entropy minimizer (such are the classifiers used in our empirical evaluation) can be viewed as Maximum Likelihood Estimators (MLE). Thus, if indeed the target distributions are Gaussian, classification can be reduced to estimating the first and second moments of the distributions.

Proposition 3. ***A Cross-Entropy minimizer is a MLE learner***
Let f_θ be a classifier of a classification task with K classes over dataset $\mathcal{X}$.

$$f_\theta : \mathcal{X} \to [0,1]^K, \quad \sum_{i=1}^K f_\theta^i(x) = 1 \quad \forall x \in \mathcal{X}$$

Then if f_θ minimizes the Cross-Entropy Loss :

$$l_{ce}(p,q) = -\sum_{i=1}^K p_i \log(q_i)$$

then f_θ acts as a Maximum Likelihood Estimator.

Proof. Denote the real distribution of a sample x from class $y(x)$ as the one-hot vector e_y yielding:

$$l_{ce}(p,q) = -p^y \log(q^y) = -\log(q^y)$$

Since q is estimated via f_θ we can derive the cross entropy loss of a single observation as:

$$l_{ce}(x) = -\log(f_\theta^{y(x)}(x))$$

and the cross entropy loss of a dataset $\mathcal{X}^m \sim D$:

$$l_{ce}(\mathcal{X}^m) = -\sum_{i=1}^{m} \log(f_\theta^{y(x)}(x))$$

On the other hand, the MLE criterion aims at finding the best parameters θ that maximize the log likelihood of the dataset under the probability estimated by f_θ :

$$\theta^*_{MLE} = \arg\max_\theta \sum_{j=1}^{K} \sum_{x \in j} \log(f_\theta^j(x)) = \sum_{i=1}^{m} \log(f_\theta^{y(x)}(x))$$

$$= \arg\min_\theta -\sum_{i=1}^{m} \log(f_\theta^{y(x)}(x)) = \arg\min_\theta l_{ce}(\mathcal{X}^m)$$

■

Corollary 3.1. *A classifier that minimizes the cross entropy loss, is also optimizing the MLE criterion. The optimal solution to the MLE criterion for Gaussian distributions is obtained by by estimating the parameters of $\mathcal{N}(\mu, \Sigma)$ according to the sampled dataset. i.e.*

$$\hat{\mu} = \frac{1}{m} \sum_{i=1}^{m} x_i = \bar{x}$$

$$\hat{\Sigma} = \frac{1}{m} \sum_{i=1}^{m} (x_i - \bar{x})(x_i - \bar{x})^T$$

(This can be easily shown as these are the roots of the partial derivatives of the log likelihood) Hence, the theoretical analysis presented in Sec. 6.1 holds even if f_θ is a Cross-Entropy classifier, as is the case in our empirical experiments.

CHAPTER 7

SUMMARY AND DISCUSSION

We described a novel unsupervised non-adversarial generative model that is capable of generating diverse multi-class images. This model outperforms previous non-adversarial generative methods, and outperforms more complicated GAN models when the training sample is small.

Unlike GAN models, our model is characterized by stable and relatively fast training, it is relatively insensitive to the choice of hyper-parameters, and it has control over each class variance in the synthesized dataset. Furthermore, empirical results show that our method is robust to the risk of mode-collapse, which plagues most GAN models when trained with insufficient data.

We further demonstrated the capability of our model to augment small data for classification, advancing the state-of-the-art in this domain. Noteably, unlike many other works in this domain, our method does not rely on any external data, and hardly depends on any prior knowledge about the true distribution. This makes it a well suited candidate for real world problems where data is hard to analyze and obtain, such as in the case of highly specialized medical imaging.

7.1 Future Work

Since our method is only used to augment the small sample, it remains orthogonal to future advances in algorithms devised for small training sets. This remains an under-researched venue, where significant progress is yet to be made.

In the unsupervised domain, further improvements may be introduced by using better, more sophisticated clustering algorithms, alongside more capable generator networks. Moreover, devising better similarity measures between images (such as the perceptual distance) that do not require a large labeled dataset may also contribute to the performance of our model.

Another interesting path to investigate involves using the classification performance as a supervisory signal for the generation of images. Thus, a generative model can be trained to produce images that optimize the classification score of a classifier trained on those images. Such a model would better reflect the co-dependency between quality generations and their usefulness in down-streaming tasks.

Lastly, investigating the effect of using the generator to transfer content from one class to another (as is briefly introduced in Appendix B) on the performance of a classifier that is trained on such data, is another interesting research direction.

BIBLIOGRAPHY

[1] D. AN, Y. GUO, N. LEI, Z. LUO, S.-T. YAU, AND X. GU, *Ae-ot: A new generative model based on extended semi-discrete optimal transport*, in International Conference on Learning Representations, 2019.

[2] S. ARORA, R. GE, Y. LIANG, T. MA, AND Y. ZHANG, *Generalization and equilibrium in generative adversarial nets (gans)*, arXiv preprint arXiv:1703.00573, (2017).

[3] S. ÄYRÄMÖ AND T. KÄRKKÄINEN, *Introduction to partitioning-based clustering methods with a robust example*, Reports of the Department of Mathematical Information Technology. Series C, Software engineering and computational intelligence, (2006).

[4] S. BARRATT AND R. SHARMA, *A note on the inception score*, arXiv preprint arXiv:1801.01973, (2018).

[5] B. BARZ AND J. DENZLER, *Deep learning on small datasets without pre-training using cosine loss*, in The IEEE Winter Conference on Applications of Computer Vision, 2020, pp. 1371–1380.

[6] M. BEN-YOSEF AND D. WEINSHALL, *Gaussian mixture generative adversarial networks for diverse datasets, and the unsupervised clustering of images*, arXiv preprint arXiv:1808.10356, (2018).

[7] Y. BENGIO, L. YAO, G. ALAIN, AND P. VINCENT, *Generalized denoising auto-encoders as generative models*, in Advances in neural information processing systems, 2013, pp. 899–907.

[8] D. BERTHELOT, N. CARLINI, E. D. CUBUK, A. KURAKIN, K. SOHN, H. ZHANG, AND C. RAFFEL, *Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring*, in International Conference on Learning Representations, 2019.

[9] K. BISRA, Q. LIU, J. C. NESBIT, F. SALIMI, AND P. H. WINNE, *Inducing self-explanation: A meta-analysis*, 2018.

[10] P. BOJANOWSKI, A. JOULIN, D. LOPEZ-PAZ, AND A. SZLAM, *Optimizing the latent space of generative networks*, arXiv preprint arXiv:1707.05776, (2017).

[11] L. BRIGATO AND L. IOCCHI, *A close look at deep learning with small data*, arXiv preprint arXiv:2003.12843, (2020).

[12] R. CARUANA, *Multitask learning*, Machine learning, 28 (1997), pp. 41–75.

[13] J. CHANG, L. WANG, G. MENG, S. XIANG, AND C. PAN, *Deep adaptive image clustering*, in Proceedings of the IEEE international conference on computer vision, 2017, pp. 5879–5887.

[14] T. CHE, Y. LI, A. P. JACOB, Y. BENGIO, AND W. LI, *Mode regularized generative adversarial networks*, arXiv preprint arXiv:1612.02136, (2016).

[15] X. CHEN, Y. DUAN, R. HOUTHOOFT, J. SCHULMAN, I. SUTSKEVER, AND P. ABBEEL, *Infogan: Interpretable representation learning by information maximizing generative adversarial nets*, in Advances in neural information processing systems, 2016, pp. 2172–2180.

[16] A. COATES, A. NG, AND H. LEE, *An analysis of single-layer networks in unsupervised feature learning*, in Proceedings of the fourteenth international conference on artificial intelligence and statistics, 2011, pp. 215–223.

[17] E. D. CUBUK, B. ZOPH, J. SHLENS, AND Q. V. LE, *Randaugment: Practical automated data augmentation with a reduced search space*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 702–703.

[18] N. DILOKTHANAKUL, P. A. MEDIANO, M. GARNELO, M. C. LEE, H. SALIMBENI, K. ARULKUMARAN, AND M. SHANAHAN, *Deep unsupervised clustering with gaussian mixture variational autoencoders*, arXiv preprint arXiv:1611.02648, (2016).

[19] C. DOERSCH, A. GUPTA, AND A. A. EFROS, *Unsupervised visual representation learning by context prediction*, in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1422–1430.

[20] J. DONAHUE, P. KRÄHENBÜHL, AND T. DARRELL, *Adversarial feature learning*, arXiv preprint arXiv:1605.09782, (2016).

[21] A. DOSOVITSKIY AND T. BROX, *Generating images with perceptual similarity metrics based on deep networks*, in Advances in neural information processing systems, 2016, pp. 658–666.

[22] M. FRID-ADAR, E. KLANG, M. AMITAI, J. GOLDBERGER, AND H. GREENSPAN, *Synthetic data augmentation using gan for improved liver lesion classification*, in 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018), IEEE, 2018, pp. 289–293.

[23] A. GHOSH, V. KULHARIA, V. P. NAMBOODIRI, P. H. TORR, AND P. K. DOKANIA, *Multi-agent diverse generative adversarial networks*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8513–8521.

[24] S. GIDARIS, P. SINGH, AND N. KOMODAKIS, *Unsupervised representation learning by predicting image rotations*, arXiv preprint arXiv:1803.07728, (2018).

[25] I. GOODFELLOW, J. POUGET-ABADIE, M. MIRZA, B. XU, D. WARDE-FARLEY, S. OZAIR, A. COURVILLE, AND Y. BENGIO, *Generative adversarial nets*, in Advances in neural information processing systems, 2014, pp. 2672–2680.

[26] P. HAEUSSER, J. PLAPP, V. GOLKOV, E. ALJALBOUT, AND D. CREMERS, *Associative deep clustering: Training a classification network with no labels*, in German Conference on Pattern Recognition, Springer, 2018, pp. 18–32.

[27] K. HE, X. ZHANG, S. REN, AND J. SUN, *Deep residual learning for image recognition*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[28] M. HEUSEL, H. RAMSAUER, T. UNTERTHINER, B. NESSLER, AND S. HOCHREITER, *Gans trained by a two time-scale update rule converge to a local nash equilibrium*, in Advances in neural information processing systems, 2017, pp. 6626–6637.

[29] Q. HOANG, T. D. NGUYEN, T. LE, AND D. PHUNG, *Mgan: Training generative adversarial nets with multiple generators*, in International Conference on Learning Representations, 2018.

[30] Y. HOSHEN, K. LI, AND J. MALIK, *Non-adversarial image synthesis with generative latent nearest neighbors*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5811–5819.

[31] S. IOFFE AND C. SZEGEDY, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, arXiv preprint arXiv:1502.03167, (2015).

[32] X. JI, J. F. HENRIQUES, AND A. VEDALDI, *Invariant information clustering for unsupervised image classification and segmentation*, in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 9865–9874.

[33] Y. JIA, E. SHELHAMER, J. DONAHUE, S. KARAYEV, J. LONG, R. GIRSHICK, S. GUADARRAMA, AND T. DARRELL, *Caffe: Convolutional architecture for fast feature embedding*, in Proceedings of the 22nd ACM international conference on Multimedia, 2014, pp. 675–678.

[34] J. JOHNSON, A. ALAHI, AND L. FEI-FEI, *Perceptual losses for real-time style transfer and super-resolution*, in European conference on computer vision, Springer, 2016, pp. 694–711.

[35] J. L. KELLEY, *General topology*, Courier Dover Publications, 2017.

[36] M. KHAYATKHOEI, M. K. SINGH, AND A. ELGAMMAL, *Disconnected manifold learning for generative adversarial networks*, in Advances in Neural Information Processing Systems, 2018, pp. 7343–7353.

[37] D. P. KINGMA AND J. BA, *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980, (2014).

[38] D. P. KINGMA AND M. WELLING, *Auto-encoding variational bayes*, arXiv preprint arXiv:1312.6114, (2013).

[39] ——, *An introduction to variational autoencoders*, arXiv preprint arXiv:1906.02691, (2019).

[40] P. KRÄHENBÜHL, C. DOERSCH, J. DONAHUE, AND T. DARRELL, *Data-dependent initializations of convolutional neural networks*, arXiv preprint arXiv:1511.06856, (2015).

[41] H.-P. KRIEGEL, P. KRÖGER, J. SANDER, AND A. ZIMEK, *Density-based clustering*, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1 (2011), pp. 231–240.

[42] A. KRIZHEVSKY, G. HINTON, ET AL., *Learning multiple layers of features from tiny images*, (2009).

[43] H. W. KUHN, *The hungarian method for the assignment problem*, Naval research logistics quarterly, 2 (1955), pp. 83–97.

[44] A. B. L. LARSEN, S. K. SØNDERBY, H. LAROCHELLE, AND O. WINTHER, *Autoencoding beyond pixels using a learned similarity metric*, in International conference on machine learning, PMLR, 2016, pp. 1558–1566.

[45] Y. LE AND X. YANG, *Tiny imagenet visual recognition challenge*, CS 231N, 7 (2015).

[46] Y. LE CUN AND F. FOGELMAN-SOULIÉ, *Modèles connexionnistes de l'apprentissage*, Intellectica, 2 (1987), pp. 114–143.

[47] K. S. LEE AND C. TOWN, *Mimicry: Towards the reproducibility of gan research*, arXiv preprint arXiv:2005.02494, (2020).

[48] J. LEMLEY, S. BAZRAFKAN, AND P. CORCORAN, *Smart augmentation learning an optimal data augmentation strategy*, Ieee Access, 5 (2017), pp. 5858–5869.

[49] L. LIN, B. LIU, X. ZHENG, AND Y. XIAO, *An efficient image categorization method with insufficient training samples*, IEEE Transactions on Cybernetics, (2020).

[50] H. LING AND K. OKADA, *Diffusion distance for histogram comparison*, in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 1, IEEE, 2006, pp. 246–253.

[51] G. LITJENS, T. KOOI, B. E. BEJNORDI, A. A. A. SETIO, F. CIOMPI, M. GHAFOORIAN, J. A. VAN DER LAAK, B. VAN GINNEKEN, AND C. I. SÁNCHEZ, *A survey on deep learning in medical image analysis*, Medical image analysis, 42 (2017), pp. 60–88.

[52] S. LIU, Y. WEI, J. LU, AND J. ZHOU, *An improved evaluation framework for generative adversarial networks*, arXiv preprint arXiv:1803.07474, (2018).

[53] M. LUCIC, K. KURACH, M. MICHALSKI, S. GELLY, AND O. BOUSQUET, *Are gans created equal? a large-scale study*, in Advances in neural information processing systems, 2018, pp. 700–709.

[54] L. METZ, B. POOLE, D. PFAU, AND J. SOHL-DICKSTEIN, *Unrolled generative adversarial networks*, arXiv preprint arXiv:1611.02163, (2016).

[55] T. MIYATO AND M. KOYAMA, *cgans with projection discriminator*, arXiv preprint arXiv:1802.05637, (2018).

[56] S. MUKHERJEE, H. ASNANI, E. LIN, AND S. KANNAN, *Clustergan: Latent space clustering in generative adversarial networks*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 4610–4617.

[57] A. NOGUCHI AND T. HARADA, *Image generation from small datasets via batch statistics adaptation*, in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 2750–2758.

[58] M. NOROOZI AND P. FAVARO, *Unsupervised learning of visual representations by solving jigsaw puzzles*, in European Conference on Computer Vision, Springer, 2016, pp. 69–84.

[59] E. OYALLON, E. BELILOVSKY, AND S. ZAGORUYKO, *Scaling the scattering transform: Deep hybrid networks*, in Proceedings of the IEEE international conference on computer vision, 2017, pp. 5618–5627.

[60] M. RAGHU, C. ZHANG, J. KLEINBERG, AND S. BENGIO, *Transfusion: Understanding transfer learning for medical imaging*, in Advances in neural information processing systems, 2019, pp. 3347–3357.

[61] M. RANZATO, C. POULTNEY, S. CHOPRA, AND Y. CUN, *Efficient learning of sparse representations with an energy-based model*, Advances in neural information processing systems, 19 (2006), pp. 1137–1144.

[62] A. J. RATNER, H. EHRENBERG, Z. HUSSAIN, J. DUNNMON, AND C. RÉ, *Learning to compose domain-specific transformations for data augmentation*, in Advances in neural information processing systems, 2017, pp. 3236–3246.

[63] S. RAVURI AND O. VINYALS, *Classification accuracy score for conditional generative models*, in Advances in Neural Information Processing Systems, 2019, pp. 12268–12279.

[64] T. SALIMANS, I. GOODFELLOW, W. ZAREMBA, V. CHEUNG, A. RADFORD, AND X. CHEN, *Improved techniques for training gans*, in Advances in neural information processing systems, 2016, pp. 2234–2242.

[65] G. SHIRAN AND D. WEINSHALL, *Multi-modal deep clustering: Unsupervised partitioning of images*, in Proceedings of the International Conference on Pattern Recognition (ICPR), 2020.

[66] K. SHMELKOV, C. SCHMID, AND K. ALAHARI, *How good is my GAN?*, in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 213–229.

[67] R. SHU, J. BROFOS, F. ZHANG, H. H. BUI, M. GHAVAMZADEH, AND M. KOCHENDERFER, *Stochastic video prediction with conditional density estimation*, in ECCV Workshop on Action and Anticipation for Visual Learning, vol. 2, 2016.

[68] P. Y. SIMARD, D. STEINKRAUS, J. C. PLATT, ET AL., *Best practices for convolutional neural networks applied to visual document analysis.*, in Icdar, vol. 3, 2003.

[69] K. SIMONYAN AND A. ZISSERMAN, *Very deep convolutional networks for large-scale image recognition*, arXiv preprint arXiv:1409.1556, (2014).

[70] A. SRIVASTAVA, L. VALKOV, C. RUSSELL, M. U. GUTMANN, AND C. SUTTON, *Veegan: Reducing mode collapse in gans using implicit variational learning*, in Advances in Neural Information Processing Systems, 2017, pp. 3308–3318.

[71] J. E. VAN ENGELEN AND H. H. HOOS, *A survey on semi-supervised learning*, Machine Learning, 109 (2020), pp. 373–440.

[72] P. VINCENT, H. LAROCHELLE, Y. BENGIO, AND P.-A. MANZAGOL, *Extracting and composing robust features with denoising autoencoders*, in Proceedings of the 25th international conference on Machine learning, 2008, pp. 1096–1103.

[73] Y. WANG, A. GONZALEZ-GARCIA, D. BERGA, L. HERRANZ, F. S. KHAN, AND J. V. D. WEIJER, *Minegan: effective knowledge transfer from gans to target domains with few images*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9332–9341.

[74] Y. WANG, C. WU, L. HERRANZ, J. VAN DE WEIJER, A. GONZALEZ-GARCIA, AND B. RADUCANU, *Transferring gans: generating images from limited data*, in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 218–234.

[75] Y. WANG, Q. YAO, J. T. KWOK, AND L. M. NI, *Generalizing from a few examples: A survey on few-shot learning*, ACM Computing Surveys (CSUR), 53 (2020), pp. 1–34.

[76] Y.-X. WANG, R. GIRSHICK, M. HEBERT, AND B. HARIHARAN, *Low-shot learning from imaginary data*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7278–7286.

[77] J. H. WARD JR, *Hierarchical grouping to optimize an objective function*, Journal of the American statistical association, 58 (1963), pp. 236–244.

[78] Q. XU, G. HUANG, Y. YUAN, C. GUO, Y. SUN, F. WU, AND K. WEINBERGER, *An empirical study on evaluation metrics of generative adversarial networks*, arXiv preprint arXiv:1806.07755, (2018).

[79] S. ZAGORUYKO AND N. KOMODAKIS, *Wide residual networks*, arXiv preprint arXiv:1605.07146, (2016).

[80] R. ZHANG, P. ISOLA, A. A. EFROS, E. SHECHTMAN, AND O. WANG, *The unreasonable effectiveness of deep features as a perceptual metric*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 586–595.

[81] X. ZHANG, Z. WANG, D. LIU, AND Q. LING, *Dada: Deep adversarial data augmentation for extremely low data regime classification*, in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 2807–2811.

[82] Y. ZHANG AND Q. YANG, *A survey on multi-task learning*, arXiv preprint arXiv:1707.08114, (2017).

[83] F. ZHUANG, Z. QI, K. DUAN, D. XI, Y. ZHU, H. ZHU, H. XIONG, AND Q. HE, *A comprehensive survey on transfer learning*, 2020.

APPENDIX

A The FID score is inadequate for multi-class datasets

In this section we will show that the FID fails to reveal intra-class variance, highlighting its inadequacy to serve as a single metric for assessing generative models on multi-class data. To do so, we will use our model to construct two datasets that obtain similar FID scores, but exhibits an apparent difference in terms of the intra-class variance. We notice that generating images from latent codes that reside in proximity yields images that are visually and semantically similar. On the other hand, sampling latent codes that come from the same latent cluster but with a greater distance from each other, yields far more diverse outputs under the model. An example of this effect is illustrated in Fig. 1

Consequently, we can generate two versions of synthesized datasets- one where each class is generated from concentrated latent codes, and one where each class is generated from sparsely sampled latent code from the same cluster. The first dataset will consist of homogeneous classes, exhibiting a small intra-class variance whereas the second one will hold classes with a larger variety of objects.

In this experimental setting, we use our model to generate two synthetic versions of CIFAR-10- one which we will term 'concentrated' which is made of generations sampled from latent codes concentrated around the cluster means, and a second dataset, termed 'sparse' which is based on sparsely sampled latent codes from the same cluster. We then evaluate these synthetic datasets using the FID and CAS scores. Results are presnted in Fig. 2.

Since the 'concentrated' dataset has low intra-class variation, each class consists of similar images, which makes it an inferior dataset for training a classifier (as is evident by the low accuracy obtained by a classifier trained on this data). On the other hand, the 'sparse' dataset is characterised by a higher intra-class variance, with diverse images in each class, yielding an effective training set for classification. Nevertheless, the FID

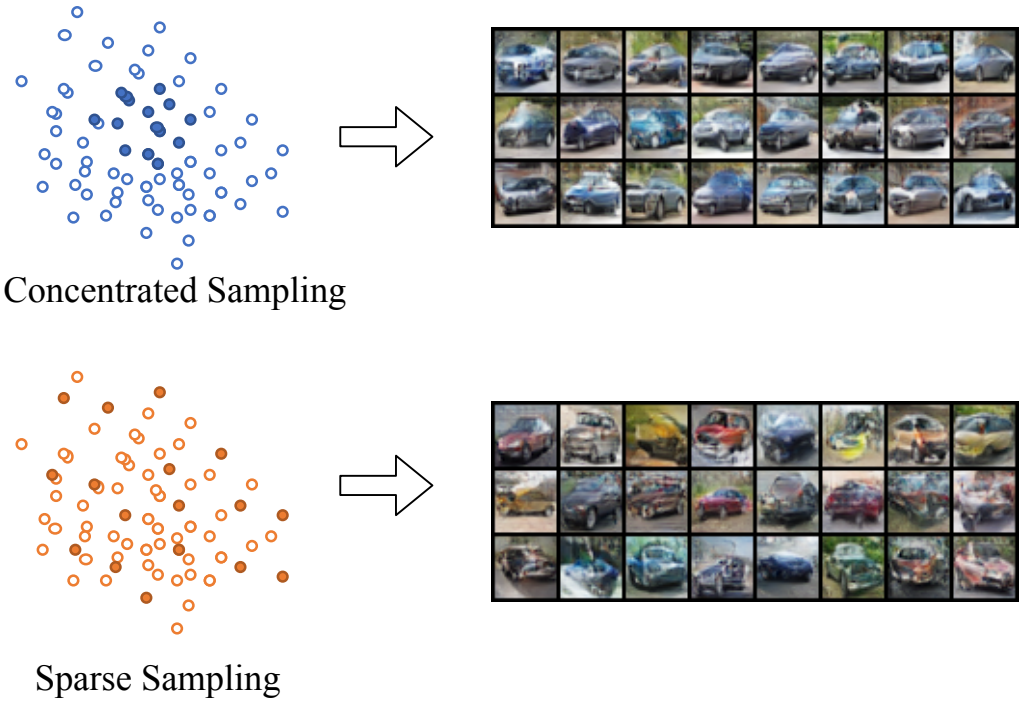


Figure 1: The effect of the scattering of latent codes on the generated images. in the top row, latent codes that are sampled around the cluster means result in similar images with small intra-class variation. In the bottom row, latent codes that are sparsely sampled result in images that exhibit a greater intra-class variance.

scores of the two datasets are barely affected by these differences, since it is based on a single multivariate Gaussian approximation of their activations in the penultimate layer of the Inception network, which cannot capture intra-class variance.

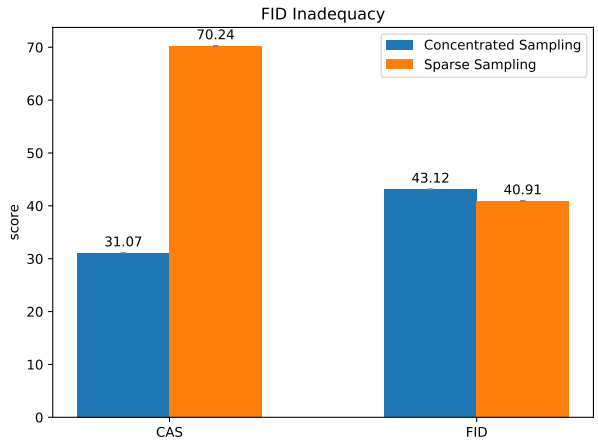


Figure 2: While the CAS score is an informative measure of the intra-class variance, the FID fails to discriminate between the two datasets.

B Class - Content Transfer

As previously stated, our model - COLA, is trained to reconstruct images from their latent representations. In the supervised version, the generator can gain access to both the class of each image, and to the image's latent code. The class component of an image can be directly inputted to the generator using conditional batch norm layers [31]. These layers normalize the intermediate inputs of the network layers according to running statistics of previous inputs. The conditional variant of these layers, normalizes each input according to the statistics of its own class only. Thus, different classes are normalized independently, and are shifted differently in the intermediate layers, leading essentially to multi-modal distributions throughout the network activations. Since the network is exposed to two independent forms of supervision - the latent code, and the class label, we may attempt to fix one supervisory signal while changing the other. Hence, if we fix the latent code of some image x from class A and input it to the model with an erroneous label of class B the model will attempt to reconstruct x given it is an image from class B (which it is not). An illustration of the outputs of this procedure is given in Fig. 3. Apparently, the model learns the salient features of each class, and when confronted with a latent code representing an object from class A with a class code of class B the model is able to generate a new image with a similar visual appearance to the original image, but with features that turns it into a member of class B. This phenomena raises several of interesting questions. In the context of this present work on generative augmentations for classification, we may ask whether such "transferred" augmentations, where each class is augmented with additional images that were reconstructed using our

model with a false class code, can aid in classification tasks. These investigations are left for future research.

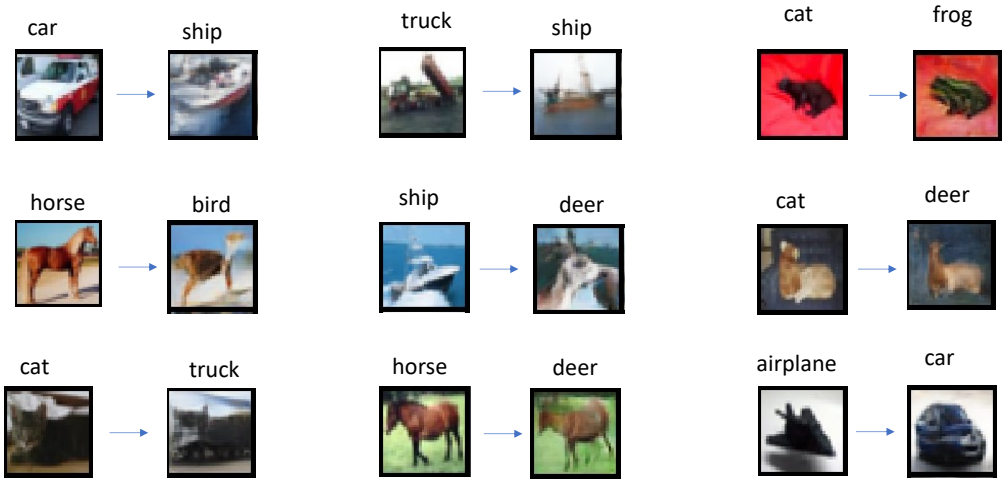


Figure 3: Examples of transferring class codes in images from CIFAR-10. Real images are on the left, images generated by our model using a different class code are on the right.

C Implementation details

C.1 Step I - Clustering the latent space.

For all experiments we use a ResNet-18 [27] network for the encoder. The network is trained with SGD with an initial learning rate of 0.05 and momentum of 0.9 for 200 epochs. Learning rate is decayd by a factor of 0.5 every 50 epochs. Training is done sequentially where an epoch optimizing the target assignment problem is followed by an epoch optimizing the rotation prediction problem. In both cases we use a batch size of 128, where in the target assignment problem images are augmented by cropping, flipping and color jitters, and in the rotation prediction task each image is rotated in all orientations, yielding a batch size of 512. Weight decay regularization of 0.0005 is used on all datasets.

C.2 Step II - Image generation.

In all our experiments we used the ADAM optimizer [37], with an initial learning rate of 0.01 for the latent code, and 0.001 for G_θ. The generative model was trained for 500 epochs, learning rate was decayed by 0.5 every 50 epochs. The only parameters that change throughout our experiments are the choice of architecture for the generator, the choice of reconstruction loss and the dimensionality of the latent space as follows:

1. **Small-Sample**: In this section, the generative function G_θ shares the same CNN generator architecture used in InfoGAN [15] (which is also the architecture used in GLO [10]). The dimension of the latent space was set to $\mathcal{Z} \subset \mathbb{R}^{K+d}$ where K is the number of classes in the dataset, and $d = 64$ for all datasets. $\mathcal{L}_{rec}$ is implemented using the unsupervised Laplacian-Pyramid loss Eq. 3.5. An ablation study of different similarity measures is presneted in Section 5.3.1.
2. **Full Data**: In the supervised version (sCOLA), the generative function G_θ shares the same CNN generator architecture used in CGAN [55], while in the unsupervised framework (COLA) we use the generator of InfoGAN [15]. In both versions $d = 128$, and $\mathcal{L}_{rec}$ is implemented using the perceptual loss Eq. 3.4.

C.3 Small-sample classification

For a fair comparison, we use the same training procedure on all data sizes. i.e. same batch size, number of epochs, iterations per epoch and learning schedule.

ResNet-20 was trained for 180 epochs, with an initial learning rate of 0.1, decayed by 0.5 every 30 epochs. Whereas WRN-16-8 was trained for 200 epochs with an inital learning rate of 0.1, decayed by 0.2 every 60 epochs. Both networks were trained using SGD optimization with a batch size of 128. An ablation study showing the effect of different architectural designs, is presented in Section 5.3.2. The classifier was trained using standard data augmentation with such image transformations as random flip and crop.

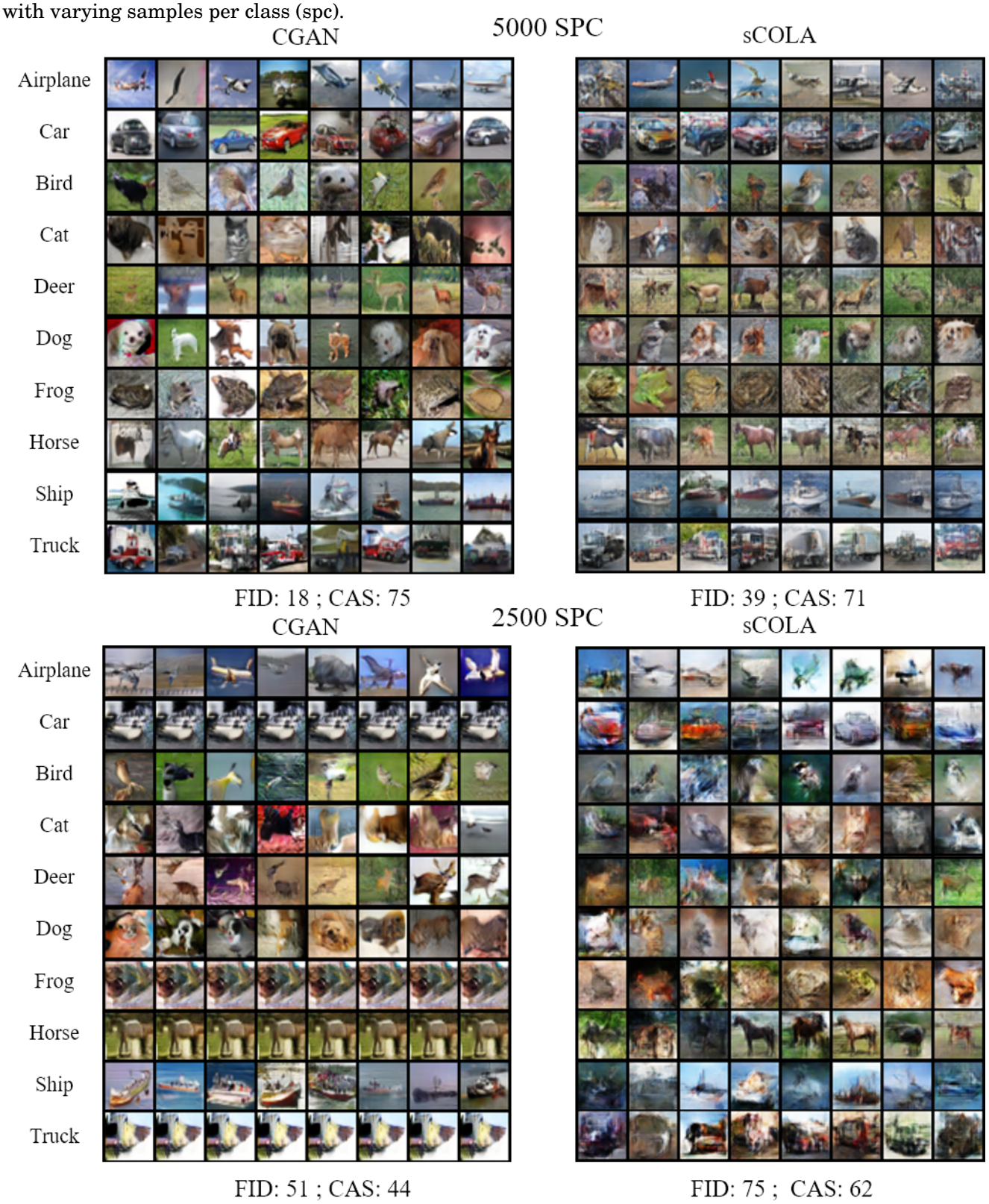
C.4 FID score implementation

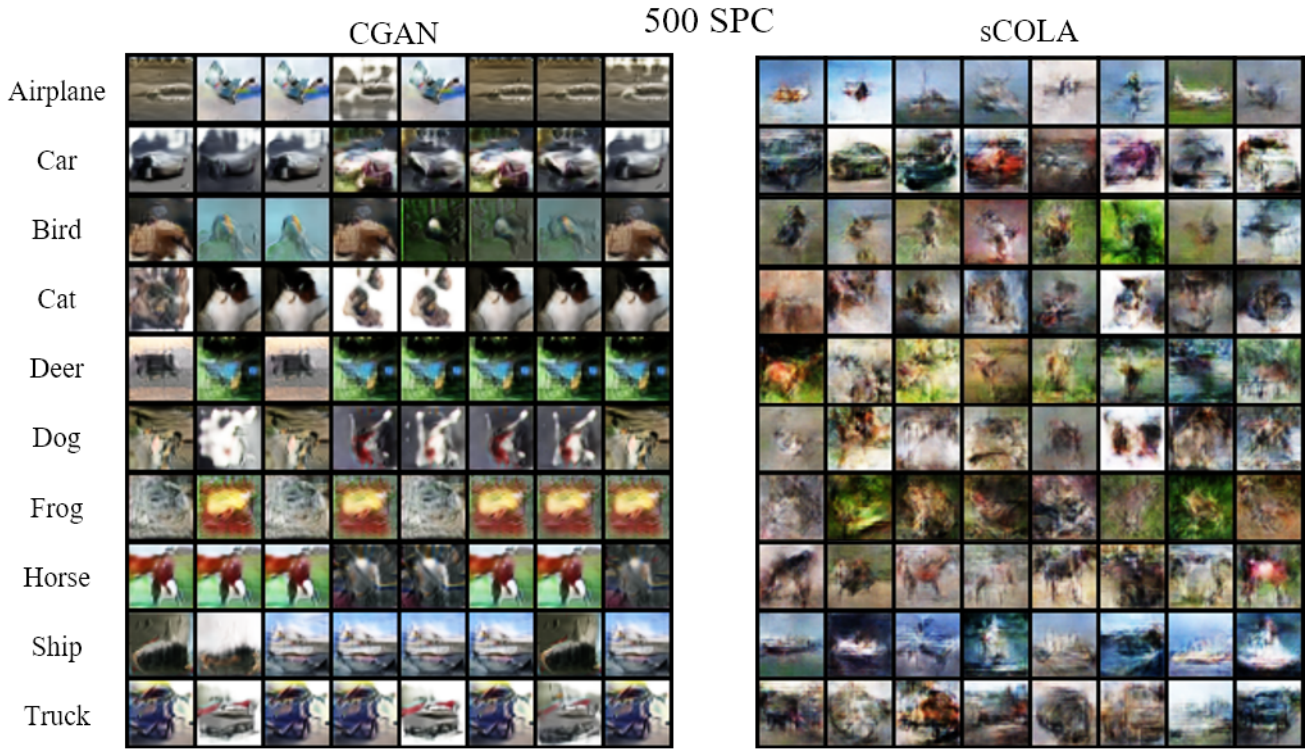
For CIFAR-10 and CIFAR-100, FID scores were computed on a sample of 10K generated images against the default Test-set of size 10K. Each model was trained 3 times, and the final score was taken as the average over 10 random samples from each model.

For STL-10, FID scores were computed on a sample of 8K generated images against the default Test-set of size 8K. Note that most generative models that had been evaluated on this dataset used the whole dataset, including the 100K unlabeled images. Therefore these previous results are not comparable to the experimental results reported here.

D Qualitative comparison

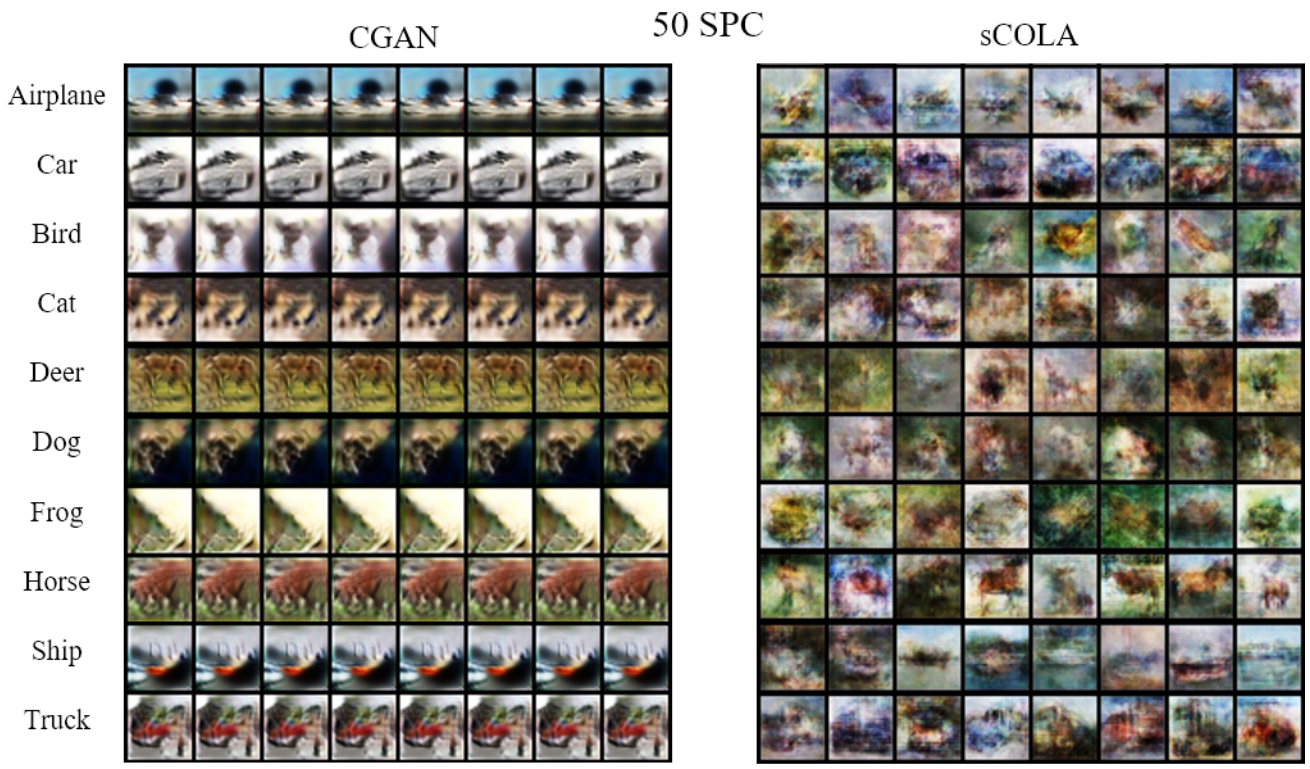
Figure 4: visualization of generations of CGAN (left) and sCOLA (right) trained on CIFAR-10 with varying samples per class (spc).





FID: 128 ; CAS: 25

FID: 103 ; CAS: 59



FID: 240 ; CAS: 18

FID: 138 ; CAS: 42