GENERATIVE MODEL BASED APPROACHES TO LEARNING WITH INCOMPLETE SUPERVISION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SİNAN GENÇOĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

DECEMBER 2023

Approval of the thesis:

GENERATIVE MODEL BASED APPROACHES TO LEARNING WITH INCOMPLETE SUPERVISION

submitted by **SINAN GENÇOĞLU** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ______________

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** ______________

Assoc. Prof. Dr. Ramazan Gökberk Cinbiş
Supervisor, **Computer Engineering, METU** ______________

Examining Committee Members:

Assoc. Prof. Dr. Emre Akbaş
Computer Engineering, METU ______________

Assoc. Prof. Dr. Ramazan Gökberk Cinbiş
Computer Engineering, METU ______________

Prof. Dr. Erkut Erdem
Computer Engineering, Hacettepe University ______________

Date:05.12.2023

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Sinan Gençoğlu

Signature :

ABSTRACT

GENERATIVE MODEL BASED APPROACHES TO LEARNING WITH INCOMPLETE SUPERVISION

Gençoğlu, Sinan
M.S., Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. Ramazan Gökberk Cinbiş

December 2023, 46 pages

The developments in deep learning have led to great advances in a variety of recognition problems. However, predominantly the state-of-the-art models rely on the availability of large annotated training sets. Therefore, great attention has recently shifted to developing models that can incorporate concepts of interest in the absence of carefully annotated large training sets for them. Commonly referred to as learning with limited supervision, these methods vary from completely unsupervised learning to model training with noisy labels. In this thesis, we focus on the problem of learning classification models of novel classes based on a small number of training examples, also known as few-shot learning. We approach this problem from a generative perspective, where we first aim to learn a generative model and then we explore the effect of the generative model on a few-shot classification task. The primary focus of this study revolves around a generative model that is founded on diffusion principles and incorporates a transformer to manipulate latent patches. This model functions by utilizing image features acquired from a pre-trained feature extractor as its conditional input. As a result of adopting this methodology, we provide an empirical assessment of the generative model's efficacy within a few-shot learning scenario.

Keywords: Generative Models, Sample-generating, Few-shot Learning

ÖZ

EKSİK GÖZETİM İLE ÖĞRENMEYE YÖNELİK ÜRETKEN MODELE DAYALI YAKLAŞIMLAR

Gençoğlu, Sinan
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Ramazan Gökberk Cinbiş

Aralık 2023 , 46 sayfa

Derin öğrenme alanındaki gelişmeler, çeşitli tanıma problemlerinde büyük ilerlemelere imkan tanımıştır. Bununla birlikte, ağırlıklı olarak en gelişmiş modeller büyük çaplı ve etiketli eğitim kümelerinin varlığına dayanmaktadır. Bu nedenle, son zamanlarda detaylı etiketlenmiş büyük eğitim kümelerinin yokluğunda ilgilenilen konseptleri modellemeye yönelik büyük ilgi ortaya çıkmıştır. Genellikle sınırlı denetimli öğrenme olarak adlandırılan bu yöntemler, tamamen denetimsiz öğrenmeden gürültülü etiketlerle model eğitimine kadar çeşitlilik göstermektedir. Bu tezde, az sayıda eğitim örneğine dayalı olarak yeni sınıfların modellerinin öğrenilmesi sorununa odaklanıyoruz, bu aynı zamanda az örnekli öğrenme olarak da bilinir. Bu probleme üretici bir perspektiften yaklaşıyoruz; ilk olarak üretken bir model öğrenmeyi hedefliyoruz ve ardından üretken modelin az örnekli sınıflandırma görevi üzerindeki etkisini araştırıyoruz. Bu çalışmanın ana odağı, difüzyon prensiplerine dayanan ve özniteliksel gösterimleri manipüle etmek için bir dönüştürücü içeren üretken bir model tabanında oluşmaktadır. Bu model, koşul girdisi olarak önceden eğitilmiş bir özellik çıkarıcıdan elde edilen görüntü özniteliklerini kullanarak çalışmaktadır. Bu metodolojinin benimsenmesinin bir sonucu olarak, üretken modelin birkaç örnekli bir öğrenme senaryosundaki etkinliğinin ampirik bir değerlendirmesini sunuyoruz.

Anahtar Kelimeler: Üretici Modeller, Örnek-üretimi, Az Örnekle Öğrenme

To my beloved family

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my supervisor Assoc. Prof. Dr. Ramazan Gökberk Cinbiş for his great support and valuable contributions.

I would like to thank the members of the thesis committee Assoc. Prof. Dr. Emre Akbaş and Assoc. Prof. Dr. Erkut Erdem.

The work in this thesis was supported in part by the TUBITAK Grant 119E597.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

ConvNet	Convolutional Neural Network
Eq	Equation
GAN	Generative Adversarial Network
VAE	Variational Auto Encoder
NF	Normalizing Flow
DM	Diffusion Model
FSL	Few-shot Learning

CHAPTER 1

INTRODUCTION

In recent years, supervised learning has made remarkable advancements in the field of deep learning. However, these approaches heavily rely on a massive amount of precisely labeled data, which is sometimes challenging and time-consuming to acquire. Therefore, it is imperative to create methods that can successfully pick up new ideas even when the training data is sparse, noisy, or poorly supervised. Building large-scale models capable of solving real-world problems requires the capacity to efficiently harness such data. Several strategies are put forth to reduce the annotation overhead, such as few-shot learning [1] [2], unsupervised representation learning [3], [4], semi-supervised learning [5] etc.

Researchers are exploring various approaches and methods to tackle these challenges including, the use of unsupervised learning [6], semi-supervised learning such as Consistency Regularization [7] and Ladder Networks [8], and few-shot learning strategies such as Relation Network [9] and Prototypical Networks [10] to address these issues. In order to get insights into the underlying structure of the data, unsupervised learning seeks to derive meaningful representations from unlabeled data. Semi-supervised learning uses labeled and unlabeled data to enhance model performance by utilizing the available unlabeled data. Few-shot learning emphasizes learning from a small set of labeled instances to simulate how humans may extrapolate information from sparse data. All of these approaches, but particularly generative model-based methods have emerged as attractive paradigms for increasing the data size.

To tackle this fundamental problem, we employ a generative model paradigm to address the inherent challenges associated with limited labeled data and the exploration of novel image classes. Unlike traditional approaches that rely on explicit data labeling or complex conditioning mechanisms, our methodology capitalizes on the power of generative models to autonomously learn and extract meaningful features directly from the raw images themselves. Our generative model leverages the intrinsic information embedded within the images themselves as conditions. This novel approach not only allows us to effectively generate images with a small dataset but also enables the generation of images for classes that were not encountered during the training process. In this chapter, we first give an overview of the data-generating problem and our contributions to the literature before going into more depth on the specifics of our work.

1.1 Overview

Deep generative models, which encompass a variety of powerful techniques such as generative adversarial networks (GANs) [11], variational autoencoders (VAEs) [12], Normalizing Flows (NFs) [13], and Diffusion Models (DM) [14], offer a unique opportunity to capture the underlying structure and distribution of complex data.

The objective of this research is to investigate the use of generative approaches for building sample-generating models. These models would be capable of synthesizing training examples for novel concepts, even in situations where limited labeled data is available. By harnessing the power of deep generative models, we can leverage unsupervised training examples, and effectively bridge the gap in training data scarcity. For example, DatasetGan [15] uses GAN to generate realistic images with minimum human effort. This approach holds promise for expanding the applicability of deep learning models to novel domains, where acquiring labeled data is a challenge.

Through our research, we aim to address the critical need for techniques that can leverage incomplete, noisy, or weakly supervised training data. By harnessing the capabilities of deep generative models, we believe it is possible to overcome the limitations of traditional supervised learning and open doors to building large-scale models capable of tackling real-world problems.

Overall, this study presents a novel and comprehensive exploration of the role of deep generative models in learning from imperfectly supervised data. By developing noise-resistant classification approaches and sample-generating models, we aim to pave the way for advancements in large-scale model development and foster progress in addressing data scarcity and quality challenges in supervised learning.

1.2 Contribution

Our research aims to investigate the efficacy of feature conditional generative models for enhancing few-shot classification performance without relying on class labels or extensive human supervision techniques. Specifically, we propose a methodology for synthesizing additional training data using diffusion models, which are trained solely on unlabeled samples with the assistance of a feature extractor. By leveraging this approach, we aim to improve the generalization capability of few-shot classifiers in a more data-driven and unsupervised manner. More specifically, we formulate the approach: (i) In the few-shot setting, we select a subset of base classes from the entire dataset. These base classes serve as the foundation for training our feature extractor and diffusion models, (ii) we train a feature extractor using the available labeled data from the selected base classes. This feature extractor captures informative representations from the input data, enabling us to gain valuable insights without the need for class labels, (iii) we employ diffusion models to generate additional training data. Unlike traditional approaches that rely on class labels, our diffusion models are conditioned on the extracted features from the feature extractor. This conditioning allows the diffusion models to generate samples that capture the inherent characteristics of the base classes, (iv) we investigate the synthesized training data generated by the diffusion models, conditioned on the features extracted by our feature extractor, which is utilized to augment the limited labeled data in the few-shot classification task. This augmentation aims to enhance the model's ability to generalize to novel classes and variations, ultimately leading to improved few-shot classification performance.

1.3 The Outline of the Thesis

In the rest of this thesis, we present an overview of the related work in Chapter 2. In Chapter 3, we first formally define the challenge of limited labeled data in few-shot learning by leveraging feature conditional generative models. We seek to improve few-shot classification performance by synthesizing additional training data without relying on class labels or highly human-supervised techniques and express our approach and findings. In Chapter 4, we present the details of our experimental setup and a thorough experimental evaluation of widely used FSL benchmark datasets. We conclude with the final remarks in Chapter 5.

CHAPTER 2

LITERATURE REVIEW

This chapter presents recent and related work on our proposed approach. Our subsequent discussion focuses on the diffusion models we use throughout our experiments as baselines.

2.1 Related Work

Within the domain of machine learning, a generative model stands as a formidable instrument, seeking to acquire an understanding of the underlying data distribution and create new samples that exhibit similarity to the original dataset. Data scarcity, data augmentation, and data generation for novel concepts are addressed by these models that learn the underlying data distribution. A variety of generative models have been proposed, each with its own strengths and weaknesses.

Autoencoders (AEs) were developed in the 1980s and popularized by [16]. They are neural networks that compress input data to a more efficient representation while reconstructing it. However, a challenge with AEs is the unknown latent codes generated by the encoder. To address overfitting and enhance robustness, Denoising Autoencoders (DAEs) [17] introduce corruption into the input by either injecting noise or concealing values, training the model to recover the original input. These models map the input to a fixed representation, various works have proposed to solve this issue, *e.g.* Variational Autoencoders (VAEs) [18], Beta-VAE [19], Vector Quantised-Variational Autoencoder [20] rather than mapping input into a fixed vector, these models map input into a distribution.

Autoencoders have been employed in few-shot learning as a means to learn informative latent representations and aid in the generation of new samples for unseen classes. For example, Prototypical Contrastive Learning [21] presents a method that combines contrastive learning with autoencoders for unsupervised representation learning in few-shot settings. Utilized the autoencoder to learn discriminative representations through contrastive loss, improving few-shot classification performance, but it heavily depends on the negative sampling strategies. Another research open-ended context-style recombination via leakage filtering [22] presents combining content from one image with the style of another in order to increase the image size and also tests the approach on few-shot learning tasks. However, it needs a high number of labeled examples and also performs poorly for augmentation purposes.

Generative Adversarial Networks (GANs) [11] were introduced to stimulate complex, large-scale image datasets and generate diverse samples. GANs consist of a generator and a critic, engaged in a competitive process to enhance performance. However, training GAN models is challenging due to issues such as training instability and convergence failure. Techniques like feature matching, minibatch discrimination, and historical averaging [23] stabilize GAN training, addressing mode collapse and instability. Wasserstein GAN (WGAN) [24] with a gradient penalty provides a more stable training objective, addressing convergence problems. StyleGAN [25] introduces style-based synthesis and disentangled latent representations, enabling fine-grained control over generated images. Conditional GANs [26, 27] extend the standard GAN framework to incorporate conditional information, allowing control over generated samples based on specific attributes. GANs have drawbacks, including a lack of control and difficulties with conditional information. Despite these challenges, GANs have demonstrated exceptional skills in producing realistic samples.

Generative Adversarial Networks (GANs) have been widely explored in the field of few-shot learning due to their ability to generate new samples and enhance the discriminative capabilities of few-shot classifiers. A couple of attempts such as Learning to Compare [28], Task-GAN [29] tried to generate additional training examples using GANs, which were incorporated into relation networks for improved few-shot classification, [30] utilized conditional GANs to generate samples conditioned on class labels in few-shot learning scenarios, but all attempts either performed poorly or heavily depends on the class labels which is very hard to acquire. In addition to these investigations, DAGAN [31] presents a data augmentation network to increase the performance of the few-shot learning system. However, this approach introduces added complexity to the training process and exhibits challenges in generalizing to unseen examples.

Overall, GANs have shown promise in addressing the challenges of few-shot learning by generating additional samples and improving the generalization capabilities of few-shot classifiers. However, mitigating mode collapse, training instability, and depending heavily on class labels in few-shot settings remain important areas of ongoing research to fully harness the potential of GANs in few-shot learning scenarios.

Normalizing Flows (NFs) [13] were introduced as an alternative generative modeling technique to address these challenges. NFs model complex distributions by applying a series of invertible transformations to a base distribution which is simple to invert. By leveraging these flexible transformations, NFs generate diverse and high-quality samples, capturing intricate data distributions. However, a Normalizing Flow-based approach proposed by [32] for few-shot image generation still faces limitations in capturing similar sample dependencies and demonstrating limited generative ability.

Despite these advantages, Normalizing Flows have their limitations, including the prerequisites for invertibility and the ability to compute Jacobians efficiently, which can limit the expressiveness of the model compared to more flexible models like GANs.

To tackle certain challenges outlined earlier, Diffusion Models (DM) [14] were introduced. These models were designed to address the expressiveness, and evaluation issues that were commonly associated with Normalizing Flows (NFs), offering an alternative approach to generative modeling with benefits such as simplicity of training, exact likelihood evaluation, and sample generation flexibility. Diffusion Models are a group of generative models designed to model complex data distribution by iteratively transforming simple distributions into the target distribution. The principle of diffusion, where noise is added to the data on a progressively increasing basis allowing it to spread and produce different samples, will be used in these models. The diffusion model provides a solid framework for probabilistic modeling, providing benefits such as precise evaluation of likelihoods and the generation of flexible samples.

A notable achievement stemming from the diffusion model is the concept of latent diffusion model (LDM) [33]. This innovative approach leverages the pure diffusion process, not at the image level, but rather on the latent representation of images. This strategy proves to be both rapid and highly efficient, surpassing traditional methods by eliminating the need for high-resolution data. By narrowing down the problem to lower dimensions, latent diffusion streamlines processes and enhances overall effectiveness.

Given the absence of a clear-cut alternative to our feature conditional latent diffusion model, a multitude of diffusion models leverage various conditions to optimize their performance. In our initial exploration, we commenced with attempts to train unconditional diffusion models using a range of conditional ones such as Improved Denoising Diffusion Probabilistic Models [34] and Improved Diffusion Model [35] by getting rid of the conditional input. Nevertheless, these models heavily rely on class labels and a large volume of training examples, leading to unintended convergence issues.

Apart from the research mentioned earlier, the studies conducted by Dataset Augmentation in Feature Space [36] and Feature Space Transfer for Data Augmentation [37] introduce augmentation methods in feature space, showing performance improvements on certain datasets. Nevertheless, these models lack comprehensive empirical testing and focus on generating new samples within the feature space, which differs from our intended approach. Another different approach Tensor Feature Hallucination [38] is a training of a model capable of generating tensor features, departing from the conventional vector features, within the realm of few-shot learning. This innovation leverages the spatial and structural characteristics inherent in tensors.

Taking these observations into account, we simplified our approach by trimming the non-relevant components of the model proposed in Diffusion Models with Transformers (DiT) [39] which trains a latent diffusion model (LDM) [33] conditioned on classes using a transformer [40] architecture that operates on latent patches, effectively enhancing both the model's performance and scalability, and adopted it as our baseline.

Research on the application of diffusion models specifically in few-shot learning is still relatively limited. While diffusion models have gained prominence for their ability to generate high-quality and diverse data samples, their adaptation to the nuanced challenges of few-shot learning scenarios has not been extensively explored in the academic literature. For example, the Few-shot Diffusion Model [41] investigates the impact of synthetic samples on classification tasks but significantly relies on class labels for conditioning images on each class, and choosing different conditions resulting in substantial performance variations. Another study Meta-DM [42] the utilization of diffusion models in the context of few-shot learning, aiming to enhance the generalization capability of models with limited training data. However, its reliance is significantly tied to the availability of labeled data in our possession.

In this study, we aim to enhance classification performance by leveraging diffusion models. When addressing the challenges of few-shot classification, diffusion models are able to capture complex data distributions and generate diverse samples. Using diffusion models, we aim to augment the limited labeled data with synthetic samples to improve the discriminative capabilities of classifiers. Through the use of these generated samples, the classifier will be able to learn from a more diverse and representative set of examples, enabling it to generalize to unknown classes and improve overall classification accuracy. Through this research, we anticipate that diffusion models will play a crucial role in pushing the boundaries of few-shot classification and advancing the state-of-the-art in this field.

2.2 Background

To solve the problem at hand, we will focus on utilizing variational autoencoders and diffusion models. VAEs will play a key role in learning informative latent representations and aiding in the generation of new samples. By training the VAE to reconstruct input data, it will capture essential features and patterns, enabling the generation of diverse and representative synthetic samples. On the other hand, DMs will be employed to capture complex data distributions and further enhance the diversity of generated samples. The diffusion process will iteratively transform a simple distribution into samples that resemble the underlying data distribution. By leveraging the capabilities of VAEs and DMs together, we aim to improve the few-shot learning scenario by generating high-quality synthetic samples and enabling better generalization to unseen classes.

In this investigation, our objective is to explore the generation of new training samples by leveraging image features in conjunction with diffusion models. We aim to address the challenge of data scarcity and enhance the learning process of deep learning models, leading to improved performance on tasks requiring robust and generalizable models, particularly in the context of few-shot learning scenarios.

2.2.1 VAE

VAE [18] is important for diffusion models due to their generative modeling capabilities, regularization properties, stable training dynamics, and ability to learn meaningful latent space representations. An encoder and a decoder are the two main components of VAE. The encoder takes an input data point and maps it to a latent space, where it is represented by mean and variance (or covariance) vectors. These vectors define a probability distribution in the latent space. The decoder then takes a sample from this distribution and reconstructs the original input data point.

Encoder is given by;

$$z \sim q(z|x) = \mathcal{N}(\mu,\, \sigma^2) \tag{2.1}$$

Decoder is given by;

$$x' \sim p(x|z) \tag{2.2}$$

Reconstruction loss is given by;

$$L_{recon} = -\mathbb{E}[\log p(x|z)] \tag{2.3}$$

KL Divergence is given by;

$$KL(q(z|x)\|p(z)) = -0.5 \sum (1 + \log \sigma^2 - \mu^2 - \sigma^2) \tag{2.4}$$

And finally, the objective function is given by;

$$L = L_{recon} + \beta . KL(q(z|x)\|p(z)) \tag{2.5}$$

where x is the given input image, z is the Gaussian noise, and β is an adjustable hyperparameter.

2.2.2 Diffusion Model

The diffusion model, as described in [14], represents a parameterized Markov chain responsible for generating samples. matching the data over time using variational inference. As a result of these transitions, a diffusion process is reversed, which slowly adds noise to the data as the sampling direction reverses itself until the signal is destroyed.

Considering a point drawn from a real data distribution, $x_0 \sim q(x)$, a forward diffusion process is used T steps at a time to add Gaussian noise to the sample, resulting in $x_1, ..., x_T$. The step sizes are managed by a variance schedule $\beta_t \in (0,1)_{t=1}^{T}$. When the noise level is appropriate, the sampling chain transitions in the forward process can be adjusted to conditional Gaussian. When this knowledge is combined with the Markov assumption, a straightforward parameterization of the forward process results;

$$q(x_{1:T}|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1}) := \prod_{t=1}^{T} \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{2.6}$$

where $\beta_1, ..., \beta_T$ is a variance schedule.

During the reverse diffusion process, Throughout the training process, the model acquires the ability to invert this diffusion procedure, enabling it to generate new data. Beginning with pure Gaussian noise $p(x_t) := \mathcal{N}(x_t, 0, I)$ the model acquires knowledge of the joint distribution $p_\theta(x_0 : T)$ as;

$$p_\theta(x_0 : T) := p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}, x_t) := p(x_T) \prod_{t=1}^{T} \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sum_\theta(x_t, t)) \tag{2.7}$$

where the time-dependent Gaussian transition parameters are learned.

The training of a Diffusion Model involves identifying reverse Markov transitions that optimize the likelihood of the provided training data. In practical terms, training involves minimizing the upper limit on the negative log-likelihood, which is a variational measure given by;

$$\mathbb{E}[-log p_\theta(x_0)] \leq \mathbb{E}_{||} \left[-log \frac{p_\theta(x_0 : T)}{q(x_1 : T|x_0)} \right] \tag{2.8}$$

In summary, the training of a Diffusion Model is a complex task, heavily reliant on the precise identification of reverse Markov transitions to optimize the likelihood of the provided training data. This practical undertaking revolves around the frequent minimization of the upper limit on the negative log-likelihood, a variational measure that encapsulates the model's performance.

CHAPTER 3

METHOD

In this chapter, we initially provide a formal definition of the constraint posed by the scarcity of labeled data in few-shot learning by leveraging feature conditional generative models. We seek to improve few-shot classification performance by synthesizing additional training data without relying on class labels or highly human-supervised techniques and express our findings.

3.1 Problem Definition

The problem we aim to address is learning from a few data using generative models, where the availability of labeled training examples is extremely limited. Traditional machine learning algorithms heavily rely on large labeled datasets for effective training and generalization. However, in numerous real-world situations, obtaining such extensive labeled datasets is challenging or impractical.

In the context of generative models, the challenge is to produce new samples while learning a meaningful representation of the given input. that capture the underlying data distribution. In our case, we endeavor to create new data instances for previously unseen classes, utilizing only a limited set of unseen class examples.

Our goal is to create a generative model that can effectively leverage the available training data to learn representations that accurately capture the data distribution. This generative model should possess the capacity to create a wide array of diverse samples, effectively acting as a means to generate augmented versions of the provided data and enhance generalization to previously unseen instances. Through the application of the generative model, our objective is to produce novel samples for classes that have not been encountered during the training phase and assess the effectiveness of this approach in the context of few-shot classification scenarios.

3.1.1 Few Shot Learning Settings

In few-shot learning, the problem can be formulated as; given a dataset $D = (X, Y)$ containing a set of classes C and their corresponding labeled examples $X = x_i$ and labels $Y = y_i$, the goal is to train a model f that can accurately classify unseen instances x' based on a limited number of labeled examples.

The training dataset D is divided into two parts: a support set $S = (X_s, Y_s)$ and a query set $Q = (X_q, Y_q)$. The support set contains a small number of labeled examples per class, where $X_s = x_i$ and $Y_s = y_i$. The query set consists of unseen instances that need to be classified, where $X_q = x_j$ and $Y_q = y_j$.

The few-shot learning problem can be formalized as finding a function f that maps an unseen instance x' to its corresponding label y', given the limited labeled examples in the support set. In our work, we focus on using generative models to increase the number of examples in the support set with newly generated synthetic data and observe the function f behavior in terms of classification performance on *unseen* examples.

3.2 Generative Models Approach

Learning from a few data using generative models, especially in a few-shot setting is an extremely challenging problem. To facilitate the recognition of *unseen* class instances, it is crucial to possess prior knowledge about class examples. This prior knowledge serves as a foundation that allows us to make predictions when encountering *unseen* examples. By leveraging this prior knowledge, we can utilize the learned patterns and characteristics of known class instances to infer and classify *unseen* instances effectively.

Our work focuses on using diffusion models (DM) to increase the training data and investigate the effects of its few-shot classification performance. We draw upon various aspects of diffusion models, specifically Latent Diffusion Models (LDM) [33], Diffusion Models with Transformers (DiT) [39], and Improved Denoising Diffusion Probabilistic Models [34], as a baseline for our considerations. The main goal is to learn a conditional diffusion model $G : E\,x\,Z \rightarrow X$, which takes feature embedding $e \in E$ as a condition comes from pre-trained feature extractor network E trained on *seen* classes and some randomness parameter z and yields synthetic data examples $x \in X$.

In few-shot scenarios, to increase the few data samples for previously *unseen* classes, we utilize existing samples from these unseen classes as conditioning inputs for the diffusion model, subsequently generating new examples based on this provided information.

We opt for feature conditioning instead of class conditioning models in few-shot scenarios due to the absence of prior knowledge about *unseen* classes, rendering the use of class conditional models unfeasible. Additionally, we refrain from employing image conditional models as they represent a lower-level approach that doesn't align with our objectives.

The objective of our generative approach is to employ features from unseen classes as conditioning inputs for our generative model, enabling the generation of new image examples to increase the data size during the few-shot learning setting.

3.3 From class conditional generative model to feature conditional model

To establish an effective generative model, our initial approach involves creating a class conditional latent diffusion model. This model takes both images and their associated ground truth class labels as conditions during the generative process. By ensuring the functionality of this class-conditioned model, we establish a baseline for generative performance. Generated samples from the class-conditional model can be seen in the following Figure 3.1.

Following this, we proceed with experimentation aimed at eliminating the reliance on

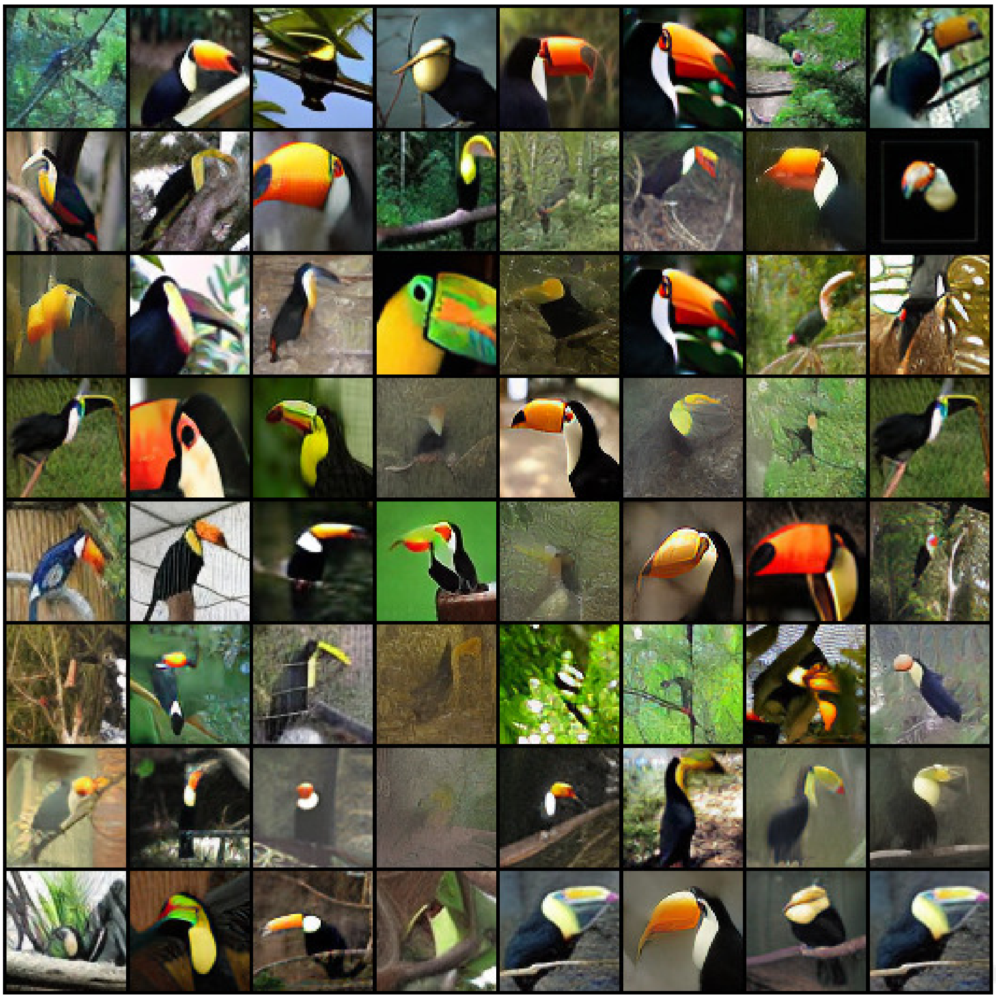

Figure 3.1: Generated synthetic samples from class-conditional generative model giving specific class label (Toucan bird).

class information. This step involves refining the model to generate diverse variations of given images without necessitating explicit class labels as conditions. The objective is to create a more versatile generative framework that can accept any image as a condition, thus enabling the synthesis of varied outputs while bypassing the need for class-specific information.

The baseline class conditional generative model involves using an embedding layer to project the number of classes into a desired hidden dimension. To refine this approach, we initiate the process by replacing the embedding layer with a linear layer. Instead of directly feeding the ground truth labels, we incorporate classifier scores derived from our feature extraction network. This involves applying softmax and argmax operations to the scores, resulting in a predicted class label. To facilitate the integration into the linear layer, we then convert this argmax prediction into a one-hot encoded representation. This refined process enhances the generative model's ability to utilize class-related information for improved sample generation.

Given the number of classes C and the desired hidden dimension H, the embedding layer maps the class index c to a latent vector z:

$$z = Embedding(c) \tag{3.1}$$

In the enhanced approach, we substitute the embedding layer with a linear transformation:

$$z = Linear(x), \tag{3.2}$$

where x represents the input data.

Given an input image x, the feature extraction network produces a set of classifier scores s:

$$s = FeatureExtractor(x) \tag{3.3}$$

Applying softmax to the scores generates a probability distribution p:

$$p_i = \frac{e^{s_i}}{\sum_{j=1}^{c} e^{s_i}} \tag{3.4}$$

The predicted class label $\hat{c}$ is obtained using the argmax operation:

$$\hat{c} = argmax(p) \tag{3.5}$$

The argmax prediction $\hat{c}$ is converted into a one-hot encoded representation y:

$$y_i = \begin{cases} 1, & \text{if } x = i \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

After training using this method, we can substantiate that the model's performance closely parallels that of the class conditional model. This implies that the substitution of the linear layer for the embedding layer does not significantly alter the model's overall performance. Furthermore, our initial approach, involving the direct integration of feature vectors into the model's intermediate layers, notably diminishes the quality of the generated samples. Figure 3.2 shows our initial approach for replacing the embedding layer with a linear layer.

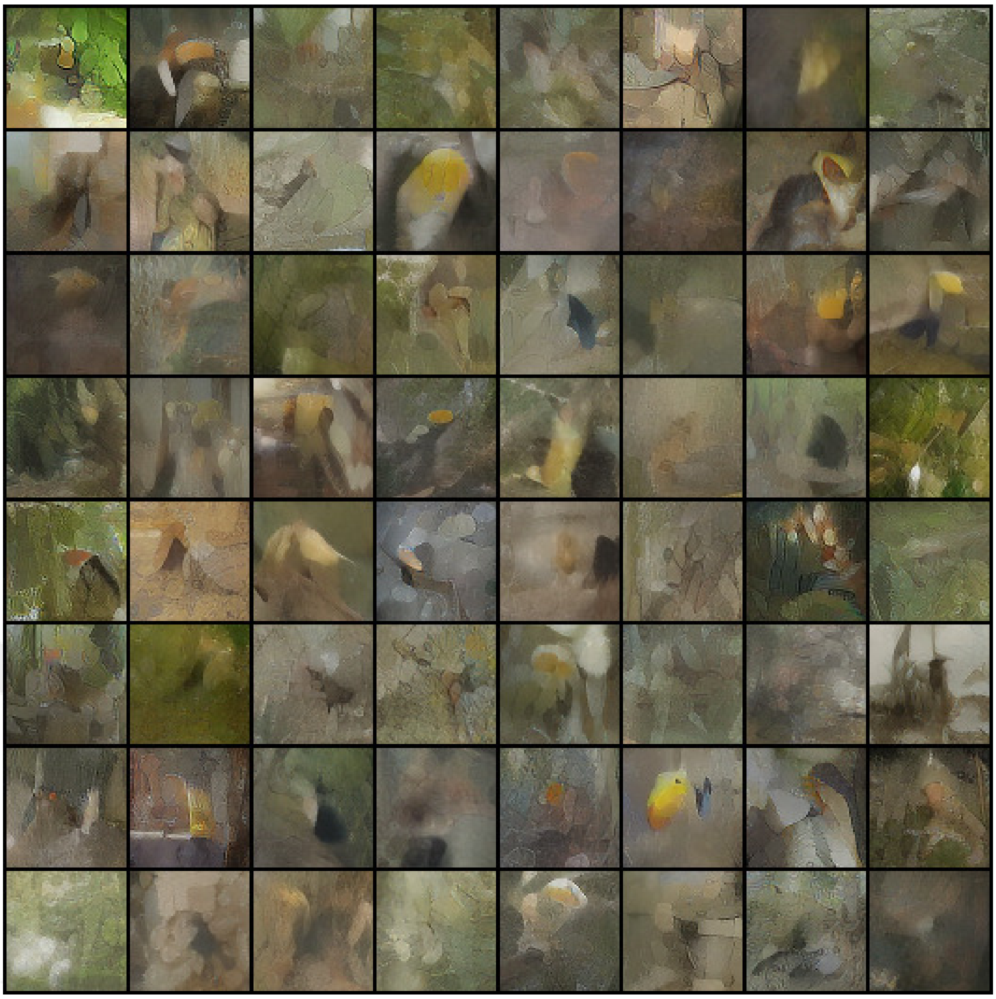

Figure 3.2: Generated synthetic samples from one hot-conditional generative model after replacement of the embedding layer with a linear layer. Given class Toucan bird.

Following the transformation of a single linear layer into a compact multi-layer perceptron (MLP) architecture, deactivating the linear layer bias, and implementing a more robust initialization scheme, notable improvements in the results were achieved. Notably, the introduction of these changes led to a significant enhancement in the model's overall performance.

In addition, a significant insight emerged from our experimentation. We observed that applying extensive augmentation techniques to the condition images had a counterproductive effect on the embedding of extractor features. This, in turn, resulted in a decline in performance. To mitigate this challenge, a strategic decision was made to eliminate heavy augmentations such as color jitter and rotation from the data pre-

Figure 3.3: Generated synthetic samples from the class-conditional generative model after replacement of the embedding with a linear layer. Given class Toucan bird.

processing pipeline.

By implementing these adjustments and fine-tuning the model accordingly, we were able to generate synthetic samples that demonstrated substantial improvements in quality and diversity. These changes collectively contributed to refining the overall generative process and enhancing the capability of the model to generate meaningful and relevant synthetic data. Despite these improvements, it's important to note that challenges persist, as a subset of the generated examples still originates from non-relevant classes. Figure 3.1 shows the samples after making the changes.

Following the series of experiments outlined earlier, our approach has provided compelling evidence that we can eliminate the embedding layer from our model's architecture. Instead, we can leverage the features extracted from earlier layers of the network to serve as more relevant conditions for our generative model.

In essence, our findings indicate that by bypassing the conventional use of class scores (p) as conditions, we can tap into the intrinsic and informative features obtained from earlier stages of the model. These features, derived directly from the input image, are shown to be highly effective in guiding the generative process. This discovery not only streamlines the architecture by removing an additional layer but also enhances the model's capacity to capture essential attributes from the input data and subsequently generate coherent and meaningful synthetic samples.

In our final set of experiments, we adopted a novel approach by utilizing a $\mathbf{R}^d$ dimensional feature vector, obtained from our feature extractor, as a crucial input for our generative models.

This feature vector, derived from the feature extractor's output, encapsulates intricate and essential information about the input image in a $\mathbf{R}^d$ dimensional space. By employing this feature vector as a conditional input for our generative models, we introduce a more focused and informative mechanism for guiding the synthesis of new data samples. This approach capitalizes on the rich insights captured by the feature extractor, resulting in a heightened ability to generate coherent and relevant synthetic samples. Figure 3.4 shows samples from the chosen class House Finch.

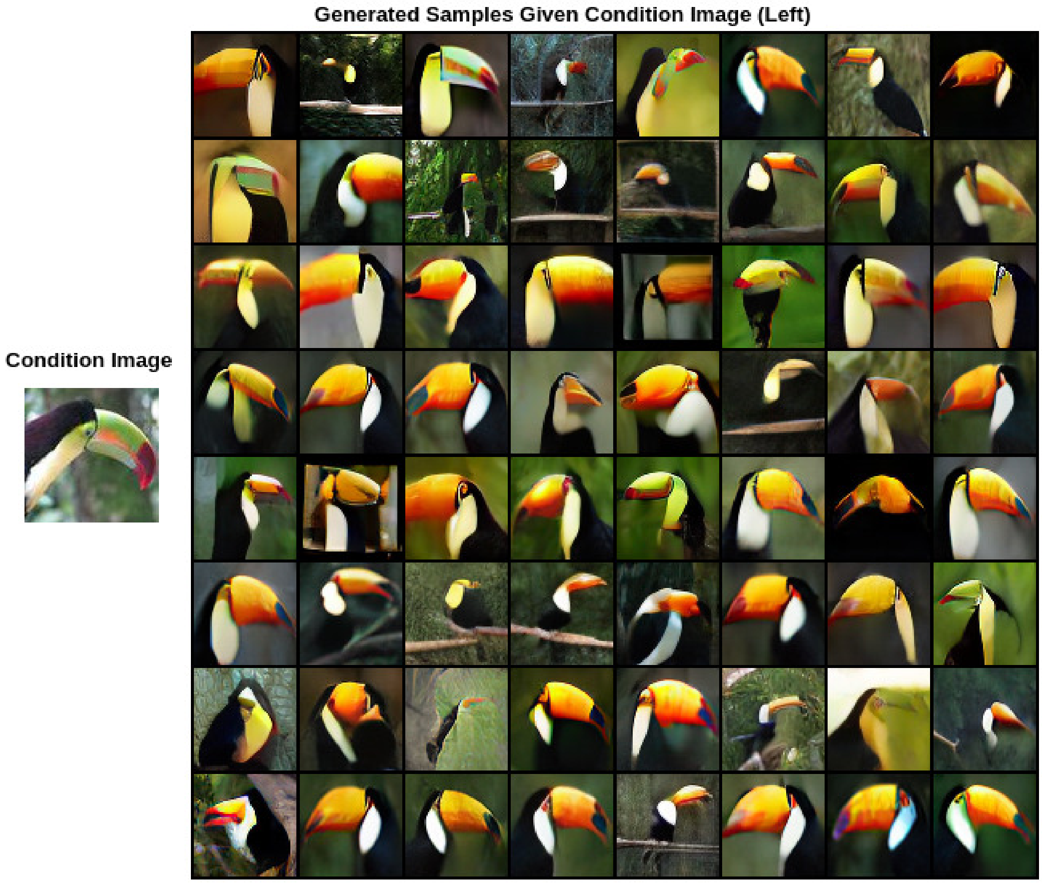

Figure 3.4: Generated synthetic samples from the feature-conditional generative model given a sample image (Toucan bird).

The outcomes derived from our latest feature conditional model unequivocally demonstrate a marked improvement in sample quality, characterized by a distinct absence of irrelevant class instances within the generated samples.

3.3.1 Feature Conditional Diffusion Model Formulation

A critical component of the synthetic data generation procedure is the choice of a suitable generative model. In our work, we adopt DMs [14] as our generative model of choice and iterate over this framework. Specifically, we use a latent diffusion model [33] as a backbone for our research.

Before formulating the whole pipeline, we provide a concise overview of fundamental concepts necessary for comprehending diffusion models. [14]. Gaussian diffusion models are based on an underlying assumption of a forward noise application process, where noise is progressively added to real examples $x_0 : q(x_t|x_0) = \mathcal{N}(x_t : \sqrt{\alpha_t}x_0, (1-\alpha_t)I)$ where α_t are hyperparameters. If we use the reparameterization trick, we can formulate the sampling as $x_t = \sqrt{\alpha_t}x_0 + \sqrt{(1-\alpha_t)}\epsilon_t$, where $\epsilon_t \sim \mathcal{N}(0, I)$.

DMs are trained in a way that they learn the reverse diffusion process so that they can fix the corrupted examples. $p_\theta(x_{t-1}|x) = \mathcal{N}(\mu_\theta(x_t), \sum_\theta(x_t))$ where x_t is the corrupted version of the real example. DMs are trained to predict a denoised version of the input x. The main objective can be defined as;

$$L_{DM} = \mathbb{E}_{x,\epsilon\sim\mathcal{N}(0,1),t}[||\epsilon - \epsilon_\theta(x_t, t)||_2^2] \tag{3.7}$$

Since most of the few shot learning settings such as [2], [43] based on doing experiments in feature space, we formulate our architecture based on this.

Our formulation has three steps. (1) We train a variant of the feature extractor network proposed in [2] using only training examples based on [44] architecture, and (2) we train an unconditional variational autoencoder to produce a lower-dimensional representational space that is perceptually equivalent to the input space called the compression step. (3) Train a diffusion model of representations $z = E(x)$ instead of a diffusion model of examples x with frozen network E. New samples can be generated by sampling a representation z from the diffusion model and then decoding it into an image using the learned decoder $x = D(z)$.

The feature extractor is responsible for extracting the class-related features from the input example x. Formally, we define the feature extractor, $\phi = f_\phi(x)$ and train it with the objective; $L(y, \hat{y}) = -\sum_i y_i log(\hat{y_i})$. In this way, the model can learn the class-relevant features from the given set of examples.

The training of the feature extractor is carried out by exposing it to a range of images from the base classes. The network learns to distill the key features and patterns inherent to these images, thus developing a comprehensive understanding of their visual characteristics. Once trained, this feature extractor can transform input images

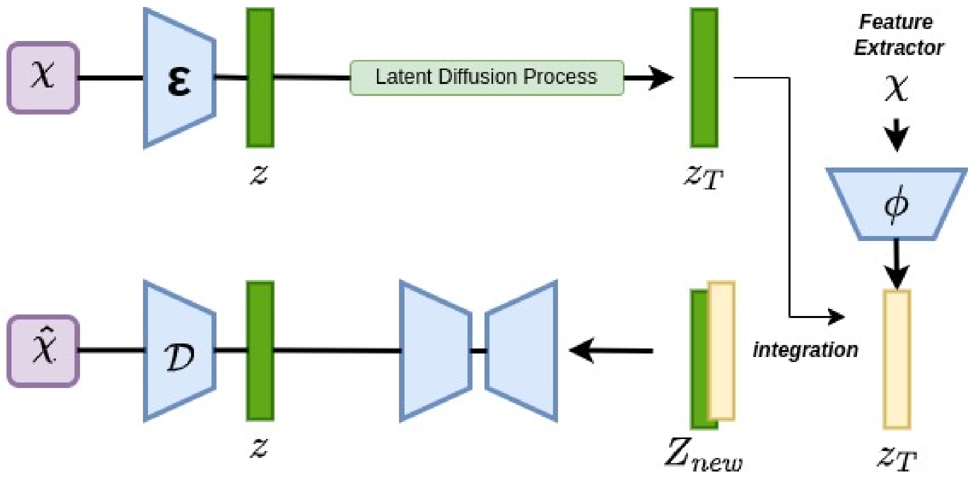

Figure 3.5: The method pipeline, at the training iteration t. The upper half shows how the encoding and diffusion models work, and the lower part shows how we combine feature vectors and noisy representations.

into meaningful and compact feature vectors. Integrating the extracted features within the latent diffusion model enhances the model's generative capabilities. During the generative process, the feature-conditioned latent diffusion model uses the learned embeddings to guide the creation of new data samples. This integration establishes a bridge between the visual attributes of the images and the latent space, facilitating the synthesis of coherent and relevant new samples.

Instead of using the input image x as it is, we use the compressed latent representation of it. More precisely, given an input image $x \in \mathbb{R}^{H \times W \times 3}$ in RGB space, the encoder which is a variant of variational autoencoder [18] (E) encodes x into a latent representation $z = E(x)$ and decoder D reconstruct the input from latent, $\hat{x} = D(z) = D(E(x))$ where $z \in \mathbb{R}^{h \times w \times c}$.

After the compression step, we train a feature conditional latent diffusion model that takes extra information which is a latent representation of the x produced by step 1. In this case, the reverse process becomes $p_\theta(x_{t-1}|x_t, \varepsilon)$. The latent diffusion model can take additional information ϕ and produce samples without using a class label.

Overall our approach, we employ a latent diffusion model that is conditioned on image features. During the training process, our model follows a specific set of steps.

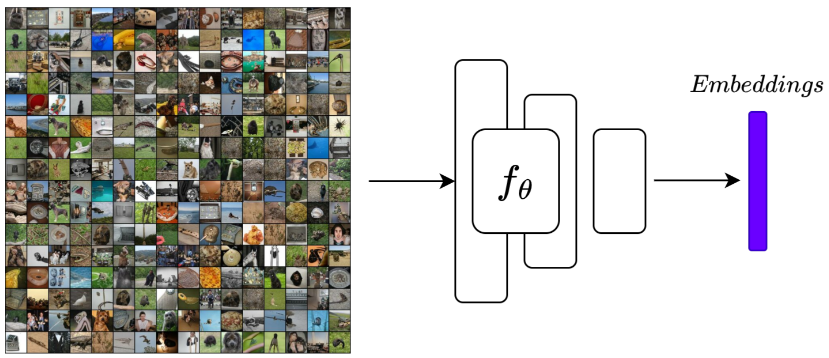

Figure 3.6: Feature extractor model. This model is used to retrieve meaningful representations of given images.

1. **Image Compression**: Initially, an image from the base classes is taken as input. This image is then compressed into a latent representation through the latent diffusion model. This step involves capturing the essential characteristics and information of the image in a more compact format.

2. **Conditioning with Features**: As a conditioning mechanism, we randomly select a sample from the same class as the previously compressed image. This selected sample is then passed through a pre-trained [44]-based feature extractor. This feature extractor is proficient at deriving high-level features from the image.

3. **Feature Integration**: The features extracted by the extractor are integrated into the network as a condition. This step involves incorporating the extracted features into the latent diffusion model, creating a link between the image's visual attributes and its latent representation.

3.3.2 Classifier-Free Guidance on Feature Level

In our training process, we employ classifier-free guidance [45] By applying a masking operation to the provided feature vector based on a predefined probability score to enhance our model's capability to generate samples for unseen classes and to increase its robustness. Notably, the utilization of classifier-free guidance leads to a substantial improvement in the quality of generated samples, consequently influencing the quality of features extracted from a pre-trained feature extractor, as the embedding derived from the feature extractor needs to be highly robust for capturing the relevant information from the provided image.

3.3.3 Feature Conditional Diffusion Model on Few Shot Learning Scenario

In the context of few-shot learning applications, we employ our feature conditional generative models. In such settings, where we have only a limited number of data examples from previously unseen classes, we initiate the process by utilizing these examples as conditions for our feature conditional generative model. This enables the generation of new examples with varying noise levels, effectively expanding the pool of available samples. Subsequently, we merge the real data samples with the generated ones. Finally, we extract features from this combined set of examples and proceed to train a classifier using these features, meticulously recording the classification accuracy.

3.4 Summary

Following the training of the aforementioned pipeline, we explore the performance of the generative model in few-shot learning scenarios. In this setting, the generative model aims to generate new examples based on input feature embeddings from unseen classes that were not encountered during training. We closely observe how this generation process impacts the classification performance and evaluate the model's behavior in handling the introduction of novel classes.

CHAPTER 4

EXPERIMENTS

In this section, we delve into an exploration of our proposed method by conducting a series of experiments. We offer an overview of the datasets used, detailing their characteristics and composition. Additionally, we introduce the evaluation metrics employed to assess our method's performance and to make meaningful comparisons with existing approaches. Using a systematic approach, we examine the effectiveness and robustness of our proposed method across different scenarios and datasets.

4.1 Experimental setup

To investigate the potential effectiveness of generative models in few-shot learning scenarios, we carefully designed our experimental setup. Our primary focus is to explore how generative models can address the challenges posed by limited labeled data. To this end, we select diverse few-shot learning datasets that reflect real-world scenarios with data scarcity. We analyze their potential to generate additional training data but want to clarify that we did not conduct experiments with these models. Our focus is on leveraging certain parts from these diffusion models to inform and inspire our approach.

We incorporate elements from diffusion models, encompassing latent diffusion; however, our model diverges significantly for several reasons. Unlike those models, which heavily depend on extensive image-text pairs and require precise yet potentially ambiguous text descriptions, our approach strives to generate meaningful examples with minimal information and a limited number of samples. While existing models often produce images that may not accurately represent the specified class in the text, our primary goal is to excel in generating relevant examples with minimal input. While those models guide us in constructing the foundational frameworks and offer insights into addressing challenges related to few-shot generation, our emphasis lies on achieving meaningful outcomes with a small amount of information and samples.

We employ established few-shot classification models as baseline specifically Rethinking Few-Shot Image Classification (RFS) [2] and evaluate its performance on the generated data using appropriate evaluation metrics. By varying factors such as the number of labeled examples per class and the complexity of the few-shot learning tasks, we aim to gain insights into the potential impact of generative models on few-shot learning performance. This investigation provides a valuable exploration of the role of generative models in mitigating the challenges of limited labeled data and sheds light on their potential benefits for improving few-shot learning outcomes.

4.1.1 Datasets

We use three widely used datasets: miniImageNet [46], CUB (Caltech-UCSD Birds-200-2011) [47] and CIFAR-FS [48] for our experiments.

Table 4.1: Statistics for miniImagenet, CUB and CIFAR-FS

Dataset	Classes	Total Images	Images Per Class	Train/Val/Test
CUB	200	6,033	30	100/50/50
miniImagenet	100	60,000	600	64/16/20
CIFAR-FS	100	60,000	600	64/16/20

CUB is a popular benchmark dataset commonly used in few-shot learning research. It comprises images representing 200 distinct bird species, with 30 images per species, totaling 6,033 images. We performed a random split of 100 classes into training sets of 64, 16, and 20 classes respectively, validation, and testing. The dataset provides a diverse range of bird species, with variations in appearance, pose, and background. Each image is associated with class labels, allowing for supervised few-shot learning experiments.

miniImagenet is another widely used dataset in few-shot learning. It is derived from the larger ImageNet dataset and contains a subset of 100 classes, with 600 images per class. The dataset is designed to simulate a more challenging few-shot learning scenario by providing a reduced number of labeled examples per class. It offers a variety of object categories, including animals, objects, and natural scenes. We adhere to the well-established splitting protocol introduced in [49], wherein 64 classes are allocated for meta-training, 16 classes for meta-validation, and the remaining 20 classes for meta-testing.

CIFAR-FS originates from the CIFAR-100 dataset, where the 100 classes are randomly divided into 64, 16, and 20 classes for training, validation, and testing, correspondingly. It is specifically tailored for few-shot learning experiments. It consists of 100 classes, with each class containing 600 images. Similar to miniImagenet, CIFAR-FS offers a reduced number of labeled examples per class to simulate the few-shot learning setting. The dataset covers various object categories, including animals, vehicles, and household items.

These datasets provide standardized benchmarks for evaluating few-shot learning algorithms and allow for fair comparisons between different approaches. Their diversity in classes, variations, and levels of data scarcity enables researchers to assess the performance and generalization capabilities of generative models in the context of few-shot learning.

Following the common experiments, we use image embeddings and the proposed methods for splitting. Table 4.1 summarizes the detailed statistics of each aforementioned dataset.

In our experiments, we use images extracted from ResNet-12 backbone pre-trained on only the train splits given in Table 4.1.

4.1.2 Evaluation Metrics

In our few-shot learning experiments, we primarily focus on evaluating the performance using two key evaluation metrics: 1-shot and 5-shot accuracy and FID scores. The terms "1-shot" and "5-shot" refer to scenarios where the model is trained with just one or five examples per class, respectively. FID metric measures the sample quality and 1-shot and 5-shot accuracy gives a proportion of correctly classified instances in the test set, considering the top-1 prediction. By focusing on these accuracy metrics, we gain insights into the model's ability to generalize and make accurate predictions with limited labeled examples, common in few-shot learning scenarios.

In the computation of FID scores for our base classes, we partition our data into an 80% training set and a 20% testing set. The testing set comprises images that the model has not encountered during training. These test images serve as a reference, and we condition the model with them, generating samples that are then compared to the test images to calculate the FID scores. Similarly, for novel classes, the procedure is repeated without splitting since none of the novel classes were seen during training.

4.2 Effects of Classifier-Free Guidance

We employ classifier-free guidance [45] for sampling and notice a significant impact on the generated samples. In Figure 4.1, by adjusting the scale of classifier-independent guidance, we achieve a dual benefit: enhancing sample quality while concurrently maintaining feature consistency across the generated samples.

Figure 4.1: Generated synthetic samples from the feature-conditional generative model using different cfg scales from left to right: no-cfg, 1.5, 2.5, 3.5, 4.5, 7.5 and 13.0 (Toucan bird).

4.3 Comparison of baselines models

In our effort to thoroughly assess the effectiveness of our proposed model, we employed Fréchet Inception Distance (FID) scores as a comparative metric against our baseline models. FID scores are widely acknowledged for quantifying the disparity in the distribution of real and generated samples, thus reflecting the quality and diversity of the generated dataset. However, a noteworthy consideration arises from our specific dataset characteristics – the limited number of examples available for each class. This inherent constraint could lead to a misleading interpretation of the FID scores. The scarcity of instances per class might not fully encapsulate the true generative prowess of our model, as the metric's sensitivity to class-wise disparities could skew its assessment. Hence, while FID scores provide valuable insights, their interpretation necessitates a contextual understanding of our data's unique distributional constraints.

Another aspect to consider is the notably low score exhibited by the CIFAR dataset. This can be attributed to its comparatively lower resolution in comparison to other datasets. Consequently, this results in dealing with a feature space of significantly reduced dimensions, potentially giving rise to mode collapse issues, particularly for certain classes.

Table 4.2: Novel classes FID statistics for miniImagenet, CUB, and CIFAR-FS

Model	**Image Resolution**	**FID Score**
miniImagenet		
Class-conditional	64x64	92.3
Onehot-conditional	64x64	243.4
Feature-conditional	64x64	74.1
CUB		
Class-conditional	64x64	66.9
Onehot-conditional	64x64	112.5
Feature-conditional	64x64	47.7
CIFAR-FS		
Class-conditional	32x32	9.5
Onehot-conditional	32x32	25.7
Feature-conditional	32x32	8.2

Table 4.3: Base Classes FID statistics for miniImagenet, CUB, and CIFAR-FS

Model	Image Resolution	FID Score
miniImagenet		
Class-conditional	64x64	53.1
Onehot-conditional	64x64	122.4
Feature-conditional	64x64	36.6
CUB		
Class-conditional	64x64	35.4
Onehot-conditional	64x64	75.3
Feature-conditional	64x64	28.8
CIFAR-FS		
Class-conditional	32x32	18.7
Onehot-conditional	32x32	48.4
Feature-conditional	32x32	13.2

To compute the FID for the base class, our initial step involves training the model using support examples from the base class. We then employ the query examples as a reference for FID calculation, providing them as conditions for our model. More specifically, In Table 4.2, among all test images, for each class k, we take the test images (query images), and we further divide them into two sets: $SetA\,(30\%)$ and $SetB\,(70\%)$. For each image in $SetA$, we use it as a conditioning input to generate five novel images. Subsequently, we calculate the FID score between all the generated images and the entire $SetB$. This process allows us to assess the variability and quality of novel image generation conditioned on the images in $SetA$.

In Table 4.3, we focus on the base classes, using their test images as the conditioning input. The FID scores are calculated by comparing the distribution of generated samples conditioned on these query images to the distribution of the actual test images for the base classes. Due to the limited number of samples, FID scores can vary depending on the selected support and query examples.

While the model excels in generating high-quality examples in certain classes, there is a notable risk of generating nonsensical images at times, potentially impacting the FID score. To mitigate the impact of randomness, random images are sampled for each class. This process is repeated multiple times, and the results are averaged to reduce the influence of randomness.

For instance, the images in Figure 4.2 and 4.3 exemplify situations where we may encounter a quality that significantly impacts the scores. This scenario is particularly pronounced when dealing with out-of-distribution examples. While the visual appearance may not seem problematic, it has a notable impact on the classification score.

Figure 4.2: Generated synthetic samples for given random military tank image (leftmost).

This particular example serves as a clear failure in providing an accurate representation of a spider image. Instances like this have the potential to drastically alter the classification score.

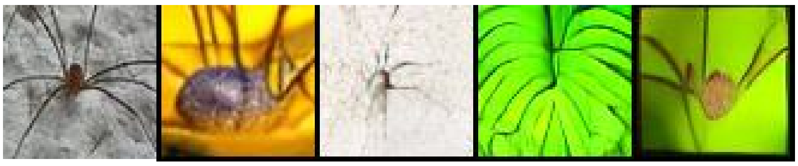

Figure 4.3: Generated synthetic samples for given random spider image (leftmost).

In order to attain consistent scores, we note that generating a larger number of examples and selecting relatively superior samples has a positive impact on classification scores. It is crucial to consider not only visually appealing samples but also the underlying representation for a more comprehensive evaluation.

It is important to note that excessive training of our model with a small dataset can result in the mode collapse problem, especially with images that have very low resolution. Therefore, maintaining a balance between training and subsequent sampling is crucial and a key consideration.

In addition to the described procedure, we compare our results with the SinGan [50] model. Figure 4.4 illustrates the comparison: the top row displays the output of our model, while the bottom row presents the results from SinGan. Each column corresponds to the output given a specific leftmost reference image as a conditioning input.

Figure 4.4: Generated synthetic samples from our model and SinGan [50] model for a given random image (leftmost).

In Table 4.4, we present the FID scores obtained from comparing the generated samples using base classes between our model and the SinGan [50] model. We observe a notable difference in fid scores. While SinGan [50] demonstrates visually appealing results with in-distribution data, it falls short in out-of-distribution cases, which is an important part of our study.

Table 4.4: Base classes FID statistics for miniImagenet

Model	Image Resolution	FID Score
miniImagenet		
Feature-conditional	64x64	94.1
SinGan [50]	64x64	196.7

4.4 Application to few-shot training

In the realm of existing literature, while an exact version for direct comparison with our feature conditional models is absent, a handful of studies [51], [52], [53] have illuminated the impact of synthetic samples on classification tasks. It is important to clarify that these studies primarily operate within a supervised setting, which differs from the few-shot learning context that our feature conditional models address. Also, it is crucial to emphasize that our specific focus lies in image-level generation.

It is worth noting that training classification models solely on synthetic datasets can precipitate a notable decline in accuracy unless it is not employed for pre-training or fine-tuning purposes. Despite the challenges we face, there's a hint of hope. With time, it becomes possible to capture important features, leading to a slight improvement in the ability of classification models to generalize.

It is important to note that our approach differs from prior work that primarily focuses on synthetic image data. While those studies explore the impact of synthetic samples on classification tasks in general, our emphasis is specifically on addressing the challenges posed by few-shot learning scenarios, where the model must learn to generalize effectively with minimal labeled examples. Our approach involves training a feature extractor network from scratch, utilizing solely the base classes. Subsequently, we immerse ourselves in a 5-way 5-shot and 5-way 1-shot experiment during testing. Here, we select five classes for each few-shot task sample, each equipped with a solitary example. To bolster this limited dataset, we strategically employ our feature conditional diffusion model to generate additional examples for these novel classes. This experimental setup aims to provide insights into the potential augmentation benefits of our approach in the context of few-shot classification scenarios. For this experiment, we take RFS [2] as our baseline and report the best scores which include relatively best synthetic examples generated by our model.

Table 4.5: Effects of synthetic samples on few-shot classification task

Model	**5-way 1-shot Accuracy**	**5-way 5-shot Accuracy**
miniImagenet		
RFS [2]	62.0	79.6
RFS [2] + synthetic features	64.8	81.2
CUB		
RFS [2]	75.7	82.3
RFS [2] + synthetic features	73.1	81.5
CIFAR-FS		
RFS [2]	71.5	86.0
RFS [2] + synthetic features	72.4	89.1

This table essentially demonstrates that when we incorporate synthetic samples, ranging between 1 and 5, generated by our feature-conditioned few-shot generative model, and utilize their features via the feature extractor network of the RFS [2], we observe an improvement in overall classification accuracies.

Another notable aspect is the slight reduction in accuracy scores observed in the CUB dataset. This can be attributed to the unique nature of CUB, which exclusively features birds, setting it apart from other datasets. While the model excels in generating high-quality images for various bird species, it encounters challenges in accurate classification, struggling to differentiate between them.

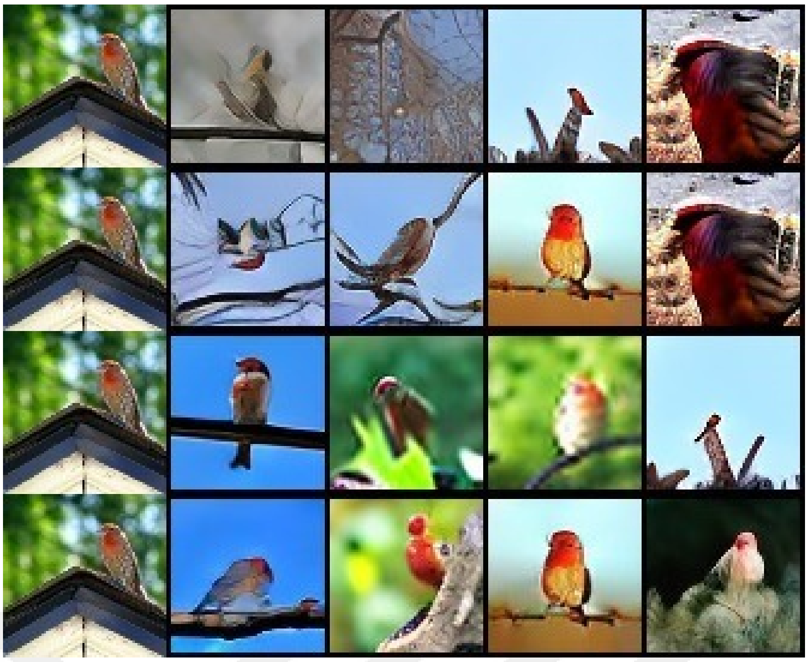

Figure 4.5: Effects of different CFG values on few-shot sampling image quality for a given random image (leftmost).

It is important to note that while the model can attain a high score as we show in Table 4.5, it is essential to note that sampling very few images and neglecting to adjust the cfg parameter may lead to a significant drop in accuracies for example for the samples in Figure 4.2 and Figure 4.3, up to 20 percent. For example, in Figure 4.5, synthetic images are sampled using the same condition image. Starting from the first row, we progressively increase the "cfg" parameter value. Interestingly, when evaluating the few-shot accuracy, we initially observed an approximately 20% decrease. However, as we proceed step by step through the bottom rows, there is a positive impact, leading to a gradual increase. Remarkably, in the last row, we observe a noteworthy 2-point improvement in the few-shot accuracy. Conversely, with proper tuning, it is possible to achieve a relatively high percentage score.

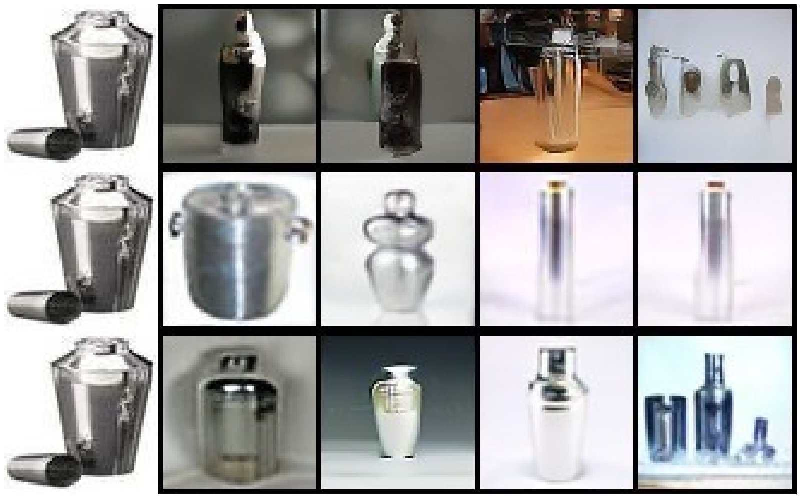

Figure 4.6: Effects of different CFG values on few-shot classification accuracy for a given random image (leftmost): top = 38.2%, middle = 55.7%, bottom = 61.4%

In Figure 4.6, it becomes evident that, despite producing visually appealing examples, the few-shot classification accuracy can vary significantly. Therefore, selecting the appropriate cfg value is crucial for achieving commendable classification scores.

CHAPTER 5

CONCLUSION

This chapter marks the culmination of the comprehensive investigations outlined in this thesis. Within its pages, we encapsulate the essence of our research endeavors and delve into prospective avenues for future exploration. The discourse extends to promising directions that have the potential to amplify and enrich the proposed approach, thereby elevating the overall value of this thesis.

5.1 Conclusion

In this thesis, we embarked on an exploration of generative models and their applications in enhancing few-shot learning scenarios. Through a series of systematic experiments and methodological refinements, we aimed to harness the potential of generative models to overcome challenges related to limited data availability and incomplete supervision.

Our investigation began by delving into the realm of conditional generative models, particularly focusing on the interplay between various generative techniques and their efficacy in few-shot learning contexts. We established that the incorporation of feature conditions within the generative process holds immense promise in expanding the scope of data synthesis. By seamlessly integrating extracted features as conditions, we not only achieved a more nuanced representation of the data but also demonstrated the capability to generate diverse and meaningful samples.

Building upon this foundation, our study revealed the intricate interplay between generative models and classification tasks. Through different experimentation, we showcased that the replacement of an embedding layer with a compact multi-layer perceptron, coupled with robust initialization strategies, offers a viable pathway to achieve competitive performance. Moreover, our findings underscored the delicate balance required in data augmentation, wherein excessive augmentation measures can detrimentally affect the quality of feature embeddings, leading to suboptimal outcomes.

Furthermore, our exploration culminated in a pioneering revelation – the effective utilization of $\mathbf{R}^d$ dimensional feature vectors extracted from the feature extractor. Leveraging this vector as a conditional input for generative models proved a pivotal stride, offering an innovative avenue for enhancing the generative process. This approach leveraged the inherent richness of feature embeddings to guide the synthesis of new data instances, yielding remarkably coherent and relevant results.

In conclusion, this thesis advances our understanding of the generative model's potential to address challenges within few-shot learning scenarios. Through our series of experimentation, careful analysis, and practical approaches, we shed light on the potential of generative models. We showcase how these models can surpass limitations and enhance the realm of few-shot learning by skillfully blending feature conditions, refining model architectures, and incorporating data augmentation techniques. This work not only contributes to the theoretical discourse but also lays a robust foundation for future research endeavors in the realm of generative models and their profound impact on enhancing learning paradigms.

5.2 Future Work

While this thesis presents a substantial advancement in the realm of generative models and their role in augmenting few-shot learning scenarios, several compelling avenues remain for further exploration and refinement.

Building upon the success of feature-conditioned generative models, future research could delve deeper into devising advanced feature extraction techniques. Exploring methods that extract more nuanced and domain-specific features could potentially enhance the generative process even further, leading to more refined and diverse synthetic samples.

Exploring the integration of semi-supervised learning techniques within the generative framework could open up new dimensions for improving few-shot learning. Leveraging the synergy between generative models and semi-supervised learning might enhance data utilization and enable better exploitation of limited labeled examples.

Investigating the application of generative models in transfer learning and domain adaptation scenarios holds promise. Adapting the proposed approach to domains with limited labeled data could pave the way for addressing real-world challenges across diverse applications.

REFERENCES

[1] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," 2017.

[2] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, "Rethinking few-shot image classification: a good embedding is all you need?," 2020.

[3] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," 2020.

[4] Y. Bengio, A. C. Courville, and P. Vincent, "Unsupervised feature learning and deep learning: A review and new perspectives," 2012.

[5] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," 2014.

[6] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," 2014.

[7] H. Zhang, Z. Zhang, A. Odena, and H. Lee, "Consistency regularization for generative adversarial networks," 2020.

[8] A. Rasmus, H. Valpola, M. Honkala, M. Berglund, and T. Raiko, "Semi-supervised learning with ladder networks," 2015.

[9] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," 2018.

[10] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," 2017.

[11] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[12] J. Schmidhuber, "Deep learning in neural networks: An overview," 2014.

[13] D. J. Rezende and S. Mohamed, “Variational inference with normalizing flows,” 2016.

[14] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” 2015.

[15] Y. Zhang, H. Ling, J. Gao, K. Yin, J.-F. Lafleche, A. Barriuso, A. Torralba, and S. Fidler, “Datasetgan: Efficient labeled data factory with minimal human effort,” in *CVPR*, 2021.

[16] G. Hinton and R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” 2006.

[17] 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*, pp. 1096–1103, 2008.

[18] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” 2022.

[19] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, “beta-VAE: Learning basic visual concepts with a constrained variational framework,” in *International Conference on Learning Representations*, 2017.

[20] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, “Neural discrete representation learning,” 2018.

[21] J. Li, P. Zhou, C. Xiong, and S. C. H. Hoi, “Prototypical contrastive learning of unsupervised representations,” 2021.

[22] K. Ridgeway and M. C. Mozer, “Open-ended content-style recombination via leakage filtering,” 2018.

[23] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” 2016.

[24] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein gan,” 2017.

[25] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” 2019.

[26] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” 2014.

[27] A. Casanova, M. Careil, J. Verbeek, M. Drozdzal, and A. Romero-Soriano, “Instance-conditioned gan,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[28] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, “Learning to compare: Relation network for few-shot learning,” 2018.

[29] J. Ouyang, G. Wang, E. Gong, K. Chen, J. Pauly, and G. Zaharchuk, “Task-GAN for improved GAN based image restoration,” 2019.

[30] C. P. Le, J. Dong, A. Aloui, and V. Tarokh, “Few-shot continual learning for conditional generative adversarial networks,” 2023.

[31] A. Antoniou, A. Storkey, and H. Edwards, “Data augmentation generative adversarial networks,” 2018.

[32] G. H. Linhao Luo, Yuan-Fang Li and S. Pan, “Normalizing flow-based neural process for few-shot knowledge graph completion,” in *The 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023.

[33] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” 2022.

[34] A. Nichol and P. Dhariwal, “Improved denoising diffusion probabilistic models,” 2021.

[35] P. Dhariwal and A. Nichol, “Diffusion models beat gans on image synthesis,” *CoRR*, vol. abs/2105.05233, 2021.

[36] T. DeVries and G. W. Taylor, “Dataset augmentation in feature space,” 2017.

[37] B. Liu, X. Wang, M. Dixit, R. Kwitt, and N. Vasconcelos, “Feature space transfer for data augmentation,” 2019.

[38] M. Lazarou, T. Stathaki, and Y. Avrithis, “Tensor feature hallucination for few-shot learning,” 2022.

[39] W. Peebles and S. Xie, “Scalable diffusion models with transformers,” 2023.

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[41] G. Giannone, D. Nielsen, and O. Winther, "Few-shot diffusion models," 2022.

[42] W. Hu, X. Jiang, J. Liu, Y. Yang, and H. Tian, "Meta-dm: Applications of diffusion models on few-shot learning," 2023.

[43] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," 2020.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[45] J. Ho and T. Salimans, "Classifier-free diffusion guidance," 2022.

[46] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," 2017.

[47] C. Wah, S. Branson, P. Welinder, P. Perona, and S. J. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

[48] L. Bertinetto, J. F. Henriques, P. H. S. Torr, and A. Vedaldi, "Meta-learning with differentiable closed-form solvers," 2019.

[49] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *International Conference on Learning Representations*, 2017.

[50] T. R. Shaham, T. Dekel, and T. Michaeli, "Singan: Learning a generative model from a single natural image," 2019.

[51] M. B. Sariyildiz, K. Alahari, D. Larlus, and Y. Kalantidis, "Fake it till you make it: Learning transferable representations from synthetic imagenet clones," 2023.

[52] S. Ravuri and O. Vinyals, "Classification accuracy score for conditional generative models," in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.

[53] V. Besnier, H. Jain, A. Bursuc, M. Cord, and P. Pérez, "This dataset does not exist: training models from generated images," 2019.