



UNIVERSITÀ DI PARMA
DIPARTIMENTO DI INGEGNERIA E ARCHITETTURA

Dottorato di Ricerca in Tecnologie dell'Informazione
XXXVII Ciclo

Munsif Ali

**Towards Continual and Few-Shot Learning in Generative
Adversarial Networks (GANs)**

DISSERTAZIONE PRESENTATA PER IL CONSEGUIMENTO
DEL TITOLO DI DOTTORE DI RICERCA

APRIL 2022 - APRIL 2025

UNIVERSITÀ DI PARMA

Dottorato di Ricerca in Tecnologie dell'Informazione

XXXVII Ciclo

**Towards Continual and Few-Shot Learning in Generative
Adversarial Networks (GANs)**

Coordinatore:

Prof. Dr. Marco Locatelli

Tutore:

Prof. Dr. Massimo Bertozzi

Dottorando: *Munsif Ali*

April 2022 - April 2025

*You are the source of my happiness and the reason I strive to
achieve more each day. This work is dedicated to my beloved
daughter "Humainah Munsif".*

Contents

Abstract	1
Acknowledgment	5
1 Continual and Few Shot Learning	7
1.1 Introduction	7
1.2 Catastrophic Forgetting	9
1.3 Significance of Continual Learning	9
1.4 Importance of Few-Shot Learning	11
1.4.1 Overfitting	12
2 Related Work	13
2.1 Background	13
2.1.1 Teacher-Student Model	13
2.1.2 Low Rank Adaptation	14
2.2 Generative Models	15
2.2.1 Variational Autoencoders	15
2.2.2 Generative Adversarial Networks	17
2.2.3 Advanced Generative Adversarial Networks	18
2.3 Continual Learning Approaches	21
2.3.1 Regularization-Based Methods	21
2.3.2 Replay Methods	22
2.3.3 Expansion Approaches	23

2.4	Few-Shot Approaches	23
2.4.1	Transfer Learning	24
2.4.2	Data Augmentation	24
2.4.3	Network Architecture	25
3	CFTS-GAN	27
3.1	Abstract	27
3.2	Introduction	28
3.3	Literature Review	30
3.4	Method	32
3.4.1	Continual Few-Shot GAN	32
3.4.2	Cross Domain Consistency Loss	34
3.4.3	Teacher Student Model	35
3.5	Experiments	37
3.5.1	Qualitative Results	38
3.5.2	Quantitative Results	38
3.5.3	Ablation Study	39
3.6	Summary	41
3.7	Supplementary Materials: Progressive Modular Expansion	42
3.7.1	Progressive Modular Expansion	42
3.7.2	Experiments and Results	43
3.7.3	Discussion	45
4	CoLoR-GAN	47
4.1	Abstract	47
4.2	Introduction	48
4.3	Literature Review	50
4.4	CoLoR-GAN model	51
4.4.1	Preliminaries	51
4.4.2	StyleGAN2 LoRA Adaptation	52
4.5	Experiments	55
4.5.1	Few-shot and Continual Learning Evaluation	56

Summary	iii
4.5.2 Ablation Study:	58
4.5.3 Few-shot Evaluation	60
4.6 Summary	62
5 Conclusion and Future Work	63
5.1 Future Directions	63
Bibliography	65

List of Figures

1.1	Illustration of CL in Human: Humans learn continuously over time, for example starting with the basics of programming, then building small projects, and eventually learn more complex models.	8
1.2	Catastrophic forgetting in GANs. At the top, sequential fine-tuning of the model on two training datasets is shown. At the bottom, after training sequentially, the generated synthetic samples are shown. It is clear that the model is unable to generate samples from the previous task and indicates a loss of older knowledge.	10
1.3	Elaboration of overfitting and mode collapse in GANs. On the left, the training samples are shown for two datasets (Sunglasses and Sketches), and on the right, the corresponding fake images are given. The model generates replicas of the training samples and produces the same synthesis images multiple times.	12
2.1	The teacher-student architecture.	14
2.2	VAE Overview: A VAE consists of two components: encoder and decoder. The encoder maps the input data into distributions, while the decoder reconstructs the data from the encoded distributions. . .	16
2.3	GAN overview: GAN consists of two neural networks: a generator G and a discriminator D . The G creates synthetic data $\hat{x}$ from a random noise to fools D . While D tries to distinguish between real data x and fake $\hat{x}$	17

2.4	StyleGAN2 model architecture. The StyleGAN2 architecture consists of two main components: a mapping and a synthesis network. Where block A is the affine transformation and w represents the intermediate latent vector.	20
3.1	The teacher-student model for continual learning GAN. The dashed lines show the frozen model. a) shows the teacher model training. b) demonstrates training of the student model.	33
3.2	Qualitative results: Generated samples. In each group, the first column is for sketches (Task 1), and the second, third, fourth, and fifth are for females (Task 2), sunglasses (Task 3), males (Task 4), and babies (Task 5).	39
3.3	Parallel expansion of adapter modules to continually learn subsequent tasks. The red lines indicate the model is already trained on previous tasks, sequentially, while the blue line represents the training of the adapter for the current task.	43
3.4	Quantitative results of PME. The PME continually generates all the tasks (CelebA, Flower, and Cathedral) without affecting the previous knowledge.	45
4.1	Low-rank tensors adaptation (solid yellow) is incorporated into the pretrained StyleGan2 model (solid green) for continual few-shot image generation. For a CL FS task, only the adapted weights are trained.	53
4.2	Qualitative results. Comparison of continual image generation of different models. The few-shot GAN is not able to produce the previous samples while lifelong GANs are able to avoid catastrophic forgetting. Each pair of rows shows two samples from each domain.	57
4.3	FS samples: A qualitative evaluation of CoLoR-GAN and other models, generating Sketches, Female, Sunglasses, and Male in FS scenario. Comparatively, our model preserves the source domain information.	61

List of Tables

3.1	Quantitative results comparison in terms of FID ($\downarrow$) and B-LPIPS ($\uparrow$) for each task with state-of-the-art methods.	39
3.2	Ablation study for the teacher using w_t for sunglasses dataset. . . .	40
3.3	Ablation study for the student on α and w_s for the sunglasses dataset.	40
3.4	Ablation study for the student with freezing discriminator for sunglasses dataset.	41
3.5	CL image generation: comparison with CAM-GAN in terms of FID ($\downarrow$) across two different tasks. The results are taken after learning all the tasks.	44
3.6	An ablation study on the weight factor a for the second scenario of PME	45
4.1	Continual few-shot image generation: comparison with different methods in terms of FID ($\downarrow$) and B-LPIPS ($\uparrow$) across different tasks. The results of all models are taken after training the model on the last (Babies) task. Best and <u>2nd highest</u> results.	57
4.2	Comparison on training efficiency. TPs, ITs: number of trainable parameters and number of iterations to learn a new task of each method, respectively. Second column represents the percentage with respect to StyleGAN2.	58
4.3	Ablation study for the rank.	58
4.4	Ablation study for the activation function.	59

4.5	Ablation for α based on LPIPS distance, L_{s-t} represents LPIPS between source and target. Last row shows LPIPS value between source and target.	59
4.6	Ablation study for the LoRA tensors in mapping and synthesis part.	59
4.7	Ablation for α based on LPIPS distance from LSUN-car (used as source dataset for pretraining), and comparison with LFS-GAN. . .	60
4.8	FS image generation: comparison with different methods in terms of FID ($\downarrow$) and B-LPIPS ($\uparrow$) across different tasks. The results are taken after learning each task.	60

Abstract

Deep learning models, inspired by the human brain, aim to replicate the capabilities of continual learning and few-shot learning. Continual few-shot learning focuses on the ability to learn knowledge over time continuously from a small number of training samples (e.g., 10 samples). Although deep learning models demonstrate impressive intelligence and achieve remarkable performance, their performance deteriorates when it comes to continual and few-shot learning without proper consideration, which are important aspects for many real-world applications. Unlike humans, continual learning is challenging due to the inherent problems of catastrophic forgetting, caused by gradient updates. At the same time, few-shot learning leads to overfitting in these models. Therefore, both scenarios recently attracted attention to eliminate these issues. Compared to discriminative models, generative adversarial networks (GANs) are not explored extensively. In this thesis, we propose novel solutions for GANs that tackle the challenges of continual and few-shot learning together. Specifically, the first method proposes a teacher-student architecture to achieve more diverse image generation. In this approach, we apply cross-domain correspondence loss to learn from few-shot samples, rather than relying on a limited set of training samples (e.g., 50 to 5000 samples). We also analyze the progressive growing of adapter modules for the continual image generation in GANs. In contrast, the second technique presents a parameter-efficient method for continual learning and few-shot image generation with less parameters and more efficient training than the state of the art.

Outline

Chapter 1 provides an introduction to continual learning (CL) and few-shot (FS) learning and highlights their significance in the context of deep learning (DL) models. It discusses the fundamental concepts, challenges, and importance. The chapter explains how CL and FS learning are essential to develop intelligent systems capable of learning new information over time from FS data.

Chapter 2 discusses generative models and the two well-known approaches for synthetic data generation. It reviews various approaches and techniques that are proposed to enhance performance in CL and FS scenarios and their classification. It also provides insights into the strengths and limitations of these approaches.

In Chapters 3 and 4, we present our novel approaches to tackle the challenges of CL and FS learning using GANs. These chapters provide a detailed description of the methods we proposed, along with theoretical explanation and experimental results. Our approaches aim to overcome issues like catastrophic forgetting (CF) and overfitting, and we discuss how our solutions contribute to more efficient and robust learning in CL and FS settings.

Finally, in Chapter 5, we conclude our research and summarize the key insights. Moreover, this chapter also highlights the future directions.

Publications

- Munsif Ali, Leonardo Rossi, and Massimo Bertozzi. "*CFTS-GAN: Continual Few-Shot Teacher Student for Generative Adversarial Networks.*" In International Conference on Pattern Recognition, pp. 249-262. Springer, Cham, 2025.
- Munsif Ali, Leonardo Rossi, and Massimo Bertozzi. "*CoLoR-GAN: Continual Few-Shot Learning with Low-Rank Adaptation in Generative Adversarial Networks.*" submitted to the ICIAP conference.
- Munsif Ali, Anwar Khan, Massimo Bertozzi, Ubaid Ullah, Saleh M. Altowaijri, Ihsan Ali, and Salman Iqbal. "*Energy and Path-Aware-Reliable Routing in*

Underwater Acoustic Wireless Sensor Networks." Wireless Communications and Mobile Computing 2022, no. 1 (2022).

Acknowledgment

First and foremost, I am grateful to Almighty Allah for granting me the strength, knowledge, and perseverance needed to complete this PhD journey. It is through His endless blessings that all of this has been made possible.

I am grateful to my supervisor, Dr. Massimo Betorzzi, for his exceptional guidance, support, and assistance throughout my research journey. His mentorship has been crucial in giving me the confidence to tackle the challenges of my research work, and his faith in my abilities has continually inspired and motivated me.

I would also like to extend my heartfelt thanks to my colleagues and friends, especially Dr. Leonardo Rossi, who has been a pillar of support at every stage of my research. Leonardo's willingness to help, insightful advice, and encouragement have made a significant impact on the quality and progress of my work. Additionally, I am deeply grateful to Dr. Tomasso Fontanini, whose assistance and thoughtful input have been invaluable throughout this journey. I would also like to express my gratitude to my group members and everyone who supported me along the way. Special thanks to the Lab Head, Dr. Andrea Prati, for his continuous support, and leadership, and for providing a research environment at the Image and Mobile Processing Laboratory (IMP Lab) at the University of Parma. His guidance has played a crucial role in shaping my academic growth.

I would also like to acknowledge the invaluable resources provided by the UNIPR High-Performance Computing (HPC) facility, which were instrumental to carry out my research. The computational power and technical support they provided significantly contributed to the successful completion of my experiments.

Finally, I am forever thankful to my family, whose unwavering love, patience, and encouragement have been my greatest source of strength. Their belief in me has been a constant source of comfort, and I am deeply thankful for their support throughout this long and challenging journey. To all of you, thank you for being part of this important chapter in my life.

Chapter 1

Continual and Few Shot Learning

1.1 Introduction

Humans continuously learn new knowledge and tasks efficiently. It is important to learn and acquire new knowledge to improve our ability to be efficient, understand new concepts, and adapt to the dynamic environment [1, 2]. The illustration of human CL ability is shown in Figure 1.1. Moreover, humans also exhibit remarkable abilities in FS learning, which often requires only a single example to distinguish between objects. Even a child can differentiate between a cat and a dog with just one sample [3]. This capability stems from the effective use of prior knowledge, which enables humans to excel in FS learning scenarios. Inspired by these abilities, DL models also incorporate FS and CL to continuously acquire new knowledge from just a few samples for a given task [4].

Although deep learning models fundamentally differ from human brains and are unable to retain older knowledge while learning a new task, this limitation is termed catastrophic forgetting (CF) and makes the models particularly challenging [5]. Furthermore, the models are inherently task-specific and designed to excel at one task or a set of related tasks, but they lack the flexibility to generalize across diverse tasks without significant retraining [6]. Unlike humans, who dynamically evolve and build new learning on existing knowledge. DL models require new training for each task,

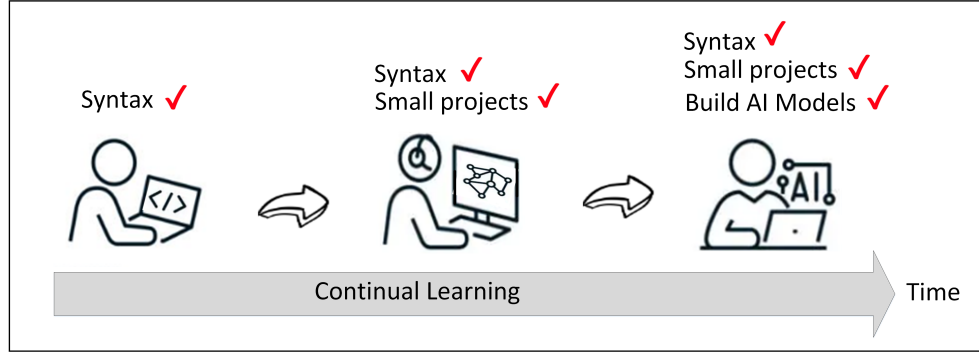


Figure 1.1: **Illustration of CL in Human:** Humans learn continuously over time, for example starting with the basics of programming, then building small projects, and eventually learn more complex models.

and this makes it difficult for them to evolve and adapt. Addressing these limitations is essential to advance DL models that mirror human-like learning processes, and evolve over time rather than being bound to static learning [7].

Similarly, DL models typically require a vast amount of data to achieve good performance [8]. This reliance on large datasets poses challenges for many real-world applications where data is limited and makes the development of DL models challenging [9]. In many practical scenarios, obtaining sufficient data is expensive, time-consuming, or simply unfeasible, particularly in specialized domains such as medical imaging or autonomous driving [10]. In such cases, the FS becomes an important consideration in DL models.

GANs have gained widespread popularity due to their fascinating applications, such as realistic image generation, creative content creation, etc. [11, 12, 13, 14, 15]. However, despite their success, GANs are less explored compared to discriminative models, especially in the area of FS and CL [16]. In this thesis, we focus on addressing these two complex tasks within the framework of GANs and aim to enhance their performance and expand their potential in FS and CL scenarios.

1.2 Catastrophic Forgetting

CF [5, 17] refers to the phenomenon in neural networks, where a model is trained sequentially on multiple tasks or datasets, it forgets the previously learned information when new tasks are introduced, as shown in Figure 1.2. It is an important consideration in CL scenarios, where models are expected to learn from a stream of data without forgetting earlier knowledge. The CF challenge was first considered in connectionist networks [17]. In more detail, when a model is trained on new data, the weights of the neural network are updated. These updates overwrite the knowledge acquired from previous tasks and lead to significant performance degradation on earlier tasks [18]. Unlike humans, who generally retain past information while learning new information, neural networks tend to prioritize the most recent information unless specifically designed to mitigate this issue.

1.3 Significance of Continual Learning

CL, also known as lifelong learning [19], incremental learning [20, 21], and never-ending learning [22], refers to a model's ability to learn new tasks or adapt to new data over time, as discussed earlier. In this thesis, we use the term CL consistently. Numerous studies explored and addressed CL over the years and recognized it as a key aspect of artificial intelligence (AI) and robotics [23, 24]. The main reason for CL is to make an autonomous system to learn over time and ability to adapt and perform effectively in a dynamic environment [25]. CL is significant for several reasons and due to limitations of DNNs.

- *Lack of adaptability:* Deep neural network (DNN) models achieved remarkable success in various domains, such as computer vision [26, 27, 28], fake image synthesis [12, 29, 30, 31], language [32, 33], etc. However, these models assume that the data distribution is static, while the data continuously evolves in a dynamic environment. Hence, the model is static and lacks a mechanism to adapt to new tasks. For instance, new situations frequently arise in real-world applications such as robotics or self-driving cars, where the models must learn

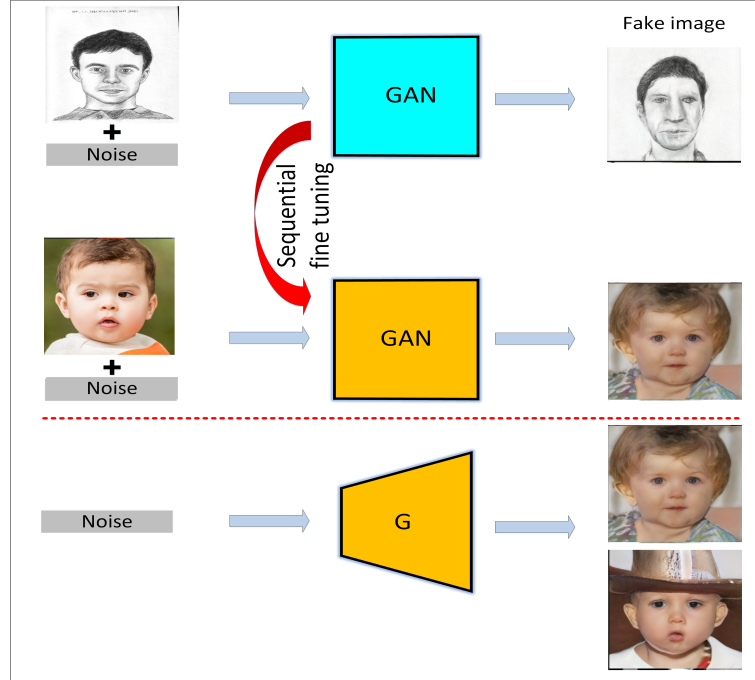


Figure 1.2: **Catastrophic forgetting** in GANs. At the top, sequential fine-tuning of the model on two training datasets is shown. At the bottom, after training sequentially, the generated synthetic samples are shown. It is clear that the model is unable to generate samples from the previous task and indicates a loss of older knowledge.

continuously and adapt to new knowledge. The gradient updates in traditional models, however, do not support this adaptability. It is because they tend to overwrite new information on the older one and prevent the model to learn effectively and adapting over time. Therefore, replicating human-like adaptability through CL [34] is essential to mimic human behavior in DL models.

- *Efficient use of resources:* The performance of deep learning (DL) models is highly dependent on a large amount of data for learning the task [35]. As a result, training a DL model requires significant resources, such as GPUs or TPUs, a vast number of training iterations, and time. Transfer learning (TL)

or fine-tuning (FT) [36] leverages pre-trained models to accelerate learning of target datasets, but it still requires efficient strategies to prevent overwriting of new information, as there is a high probability of losing prior knowledge. In contrast, CL models take advantage of TL while eliminating the risk of losing previous information [37]. CL models also remove the need to store or retrain on large volumes of past data. Such approaches learn new tasks with fewer resources and smaller datasets and make them more efficient in terms of memory usage and computational cost. Additionally, CL reduces the need to retrain models from scratch whenever new data arrives, which saves both time and computational resources.

1.4 Importance of Few-Shot Learning

Drawing inspiration from human FS learning abilities, DL practitioners and researchers attempt to replicate the FS learning process in these models. A formal definition, as presented in [10], states that a model learns from experience E in relation to task T , with performance measured by P , even that E is very scarce in FS learning [38]. In other words, FS learning enables a model to learn from a few examples (i.e., 10 samples per task). This contrasts with traditional deep learning, which typically requires large amounts of data to achieve good performance [35]. There are numerous challenges in FS learning, which highlight its significance in modern DL models. As a result, FS learning is important from three aspects [9].

- *Data scarcity*: Due to many reasons such as legal and ethical concerns, high costs, and rare cases to collect samples, it is very challenging to collect a large number of samples needed to train a DL model. Therefore, FS learning is crucial for applications where only a limited number of samples are available.
- *Reduce gap*: The fact is that DL models are data-hungry, which limits their applicability in FS learning scenarios. As a result, FS learning techniques help bridge this gap by mimicking the human ability to learn effectively from FS data.

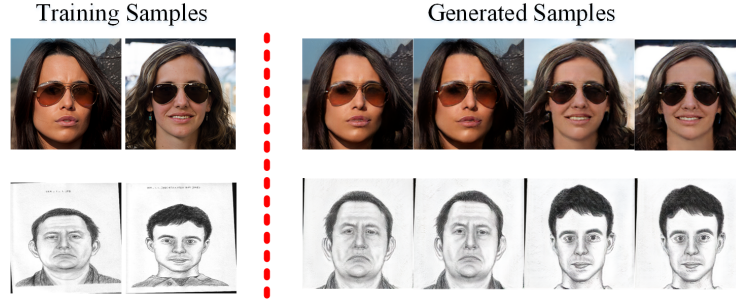


Figure 1.3: Elaboration of overfitting and mode collapse in GANs. On the left, the training samples are shown for two datasets (Sunglasses and Sketches), and on the right, the corresponding fake images are given. The model generates replicas of the training samples and produces the same synthesis images multiple times.

- *Quick deployment:* FS learning enables models to learn quickly, reduces both computational costs and training time.

1.4.1 Overfitting

As mentioned in the previous section, FS learning offers significant advantages, but it also leads to a well-known challenge called overfitting [39]. In FS learning, the model tends to memorize [40, 41] the training samples rather than learning from them. As a result, the model simply replicates the training data [42] and limits its ability to generalize to new unseen examples. Additionally, GANs are inherently susceptible to the mode collapse problem [43, 44], which limits the diversity of generated images [45]. When FS learning is applied in GANs, it intensifies the mode collapse problem and leads to even less diverse image generation as shown in Figure 1.3.

Chapter 2

Related Work

This chapter explains the background knowledge that we used in our proposed studies. Our work focuses on GANs [46], a generative model [12, 29, 47] in the context of both FS and CL. Therefore, this chapter explains generative models and their two well-known approaches [46, 48] and the advanced GANs architecture. Furthermore, we will delve into the concepts of FS and CL, explore their broad classification, and the integration of generative models within these frameworks.

2.1 Background

2.1.1 Teacher-Student Model

In DL, the authors of [49] proposed a teacher-student idea to accelerate the training process. The teacher-student model is also referred to as knowledge distillation (KD). It is a method where a smaller, simpler model is trained to mimic the behavior of a larger, more complex model. The smaller model is called the student while the larger model is called the teacher. The main goal of this approach is to transfer the knowledge learned by the teacher model into the student model, which typically has fewer parameters than the teacher and is computationally efficient.

The teacher model is typically a pre-trained, large neural network [50] that performs well on a given task. This model is then used to generate outputs, which are the

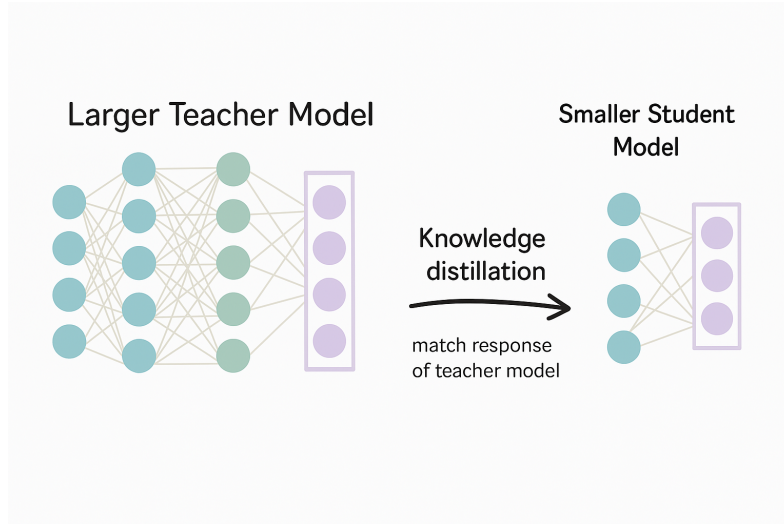


Figure 2.1: The teacher-student architecture.

predicted probabilities or output distributions from the teacher. The student model is trained to match the teacher’s predictions as closely as possible. It usually minimizes the difference between the teacher’s output and the student’s output using a loss function. The teacher-student model idea is depicted in Figure 2.1.

The teacher-student approach is beneficial in a variety of ways. The student model is used as a model compression [51, 52, 53] and makes it suitable for deployment in resource-constrained environments like mobile devices or embedded systems. Despite being smaller, the student model achieves better generalization and performance [54, 55]. It allows the transfer of complex knowledge from a large, well-trained model to a simpler one, and improves the student’s accuracy on certain tasks [56].

2.1.2 Low Rank Adaptation

Fine-tuning large pre-trained models, such as GPT-2 [57] or RoBERTa [58], for specific downstream tasks typically requires updates for a large number of parameters. This is computationally expensive and time-consuming, especially with models that have billions of parameters. Therefore, low-rank adaptation (LoRA) [59] is a method

used to adapt pre-trained models efficiently and reduces the number of parameters. LoRA addresses this issue with a focus on a more parameter-efficient way to adapt the model without modifying its entire set of parameters. It introduces low-rank matrices into the pre-trained model that capture the task-specific information. This enables the model to adapt to new tasks without a significant increase in the number of trainable parameters.

The low-rank adaptation incrementally updates internal pre-trained model weights. Consider a pre-trained weight matrix $W_0 \in \mathbb{R}^{d \times k}$. Instead of modifying the entire weight matrix during incremental fine-tuning, it approximates the update as a low-rank adaptation of two matrices $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$, where r is the rank of the matrix. The adapted weights $\hat{W}$ are given as follows:

$$\hat{W} = W_0 + \Delta W = W_0 + \frac{\alpha}{r} BA \quad (2.1)$$

where ΔW represents the weight updates and α the scaling factor that controls the LoRA influences. Selecting an r much lower than d and k , means that $d \times r + r \times k \ll d \times k$ and therefore that the additional trained parameters can be really small.

2.2 Generative Models

Generative models, a subfield of DL, have captured the attention of both researchers and industry practitioners due to their remarkable success, productivity, and applications [60]. Generative modeling refers to the task of learning a distribution $p_{data}(x)$ over data x , such that the model can generate new data $\hat{x}$ that resembles the original data [61]. In the context of DL, generative models aim to learn the underlying probability distribution from which the observed data is drawn. Two of the most well-known approaches in generative modeling are Variational Autoencoders (VAEs) [48] and GANs [46]. The following subsections explain these models in detail.

2.2.1 Variational Autoencoders

The VAE [48] is a type of generative model that learns a probabilistic mapping from data to a latent space. It builds on the concept of autoencoders [62] by introducing a

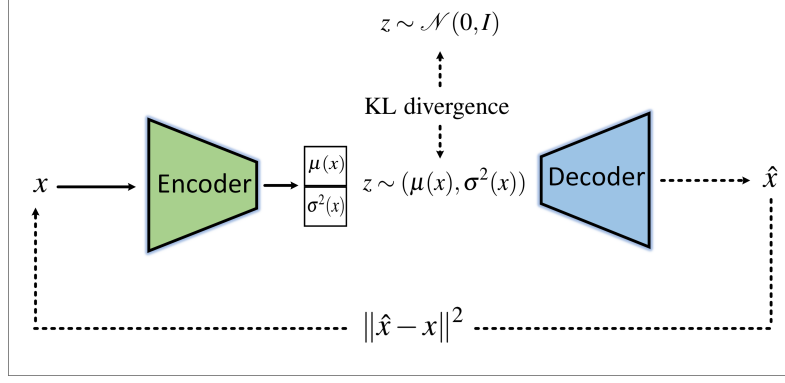


Figure 2.2: **VAE Overview:** A VAE consists of two components: encoder and decoder. The encoder maps the input data into distributions, while the decoder reconstructs the data from the encoded distributions.

probabilistic approach instead of a vector in the encoding process. A VAE consists of two primary components: encoder q and decoder p as shown in Fig. 2.2. The encoder learns latent space $q(z | x)$ while the decoder reconstructs the data x from a distribution $p(x | z)$.

Given an input x , the encoder maps it to a distribution over the latent space $q(z | x)$. Instead of encoding x to a single vector, the encoder learns a distribution $q(z | x) = \mathcal{N}(z | \mu(x), \sigma^2(x))$. In simple words, the encoder's function is to map x into $\mu(x)$ and $\sigma^2(x)$. The objective of the VAE is

$$\mathcal{L}_{VAE} = -\text{KL}(q(z | x) \| \mathcal{N}(0, I)) + \mathbb{E}_{q(z|x)} [\log p(x | z)]. \quad (2.2)$$

The first term of the equation is the KL divergence, which aims to bring the two distributions, $q(z | x)$, and $\mathcal{N}(0, I)$, closer together. The second term is the reconstruction loss, which is given by

$$\mathbb{E}_{q(z|x)} [\log p(x | z)] = \|\hat{x} - x\|^2. \quad (2.3)$$

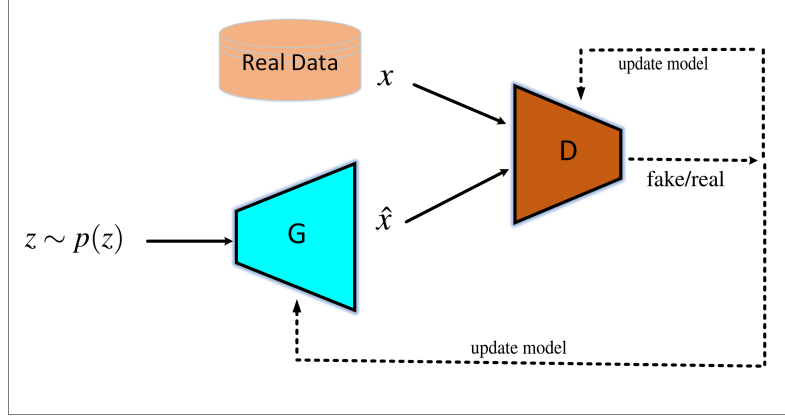


Figure 2.3: **GAN overview:** GAN consists of two neural networks: a generator G and a discriminator D . The G creates synthetic data $\hat{x}$ from a random noise to fools D . While D tries to distinguish between real data x and fake $\hat{x}$.

2.2.2 Generative Adversarial Networks

A standard Generative Adversarial Network (GAN) [46] consists of two neural networks: a generator G and a discriminator D , that are trained simultaneously in a game-theoretic framework. The G creates synthetic data $\hat{x}$ and attempts to fool D , while D tries to distinguish between real data x and fake one $\hat{x}$. The D learns to become more accurate in distinguishing the fake data from the real data. The GANs block diagram is shown in Fig. 2.3. The G takes a random noise vector z (from a prior distribution $p(z)$) and transforms it into a data $\hat{x} = G(z)$. While the discriminator either takes the generated output $\hat{x} = G(z), z \sim p(z)$ or the real data x from $p_{data}(x)$ and outputs a probability $D(x)$ to distinguish between them.

The GAN can be viewed as a two-player zero-sum game. The G tries to minimize the probability that the D correctly classifies its generated samples as fake, while the discriminator tries to maximize its ability to correctly distinguish real data from fake data. The objectives of G and D are given as:

$$\mathcal{L}_G = \mathbb{E}_{z \sim p(z)} [\log(1 - D(G(z)))] \quad (2.4)$$

$$\mathcal{L}_D = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p(z)} [\log(1 - D(G(z)))]. \quad (2.5)$$

The G and D are trained simultaneously. The combined optimization process as a minimax game is given as

$$\min_D \max_G (\mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p(z)} [\log(1 - D(G(z)))]). \quad (2.6)$$

This adversarial setup creates a system where, ideally, the generator creates indistinguishable data from real data, and the discriminator can no longer reliably tell the difference between real and synthetic data.

2.2.3 Advanced Generative Adversarial Networks

Continual Adaptation Modules for GANs

The continual adaptation modules for GANs, abbreviated as CAM-GAN [63] addresses the CF by introducing task-specific adapter modules [64]. The adapter modules, as described in [64], are typically inserted between layers of a pre-trained model. They are designed to learn new representations or transformations and enable the model to generalize better for new tasks without retraining the entire network.

Let the original pre-trained model is represented as $\theta_w(x)$, and we want to adapt it to a new task T . Instead of fine-tuning the entire model, an adapter module with parameters ϕ_v is inserted into the network, which modifies the input to each layer of the model. For instance, the output of a layer in the original pre-trained model is modified and the transformation is given as $\phi_v(\theta_w(x))$. Where x is the input to the layer and $\phi_{w,v}(x)$ is the task-specific adapter weights along with pre-trained weights. The adapter learns the task-specific modifications while the rest of the parameters $\theta_w(x)$ remain frozen. At the start, the task-specific adapters are $\phi_{w,v}(x) \approx \theta_w(x)$ to preserve the pre-trained weights. During the training, only the parameters of the adapters are updated. The choice of $\phi_v \ll \theta_w$ ensures that the number of trained parameters is very small.

The CAM-GAN [63] uses adapter modules for continual learning (CL) in GANs. It appends adapter modules to the top of the GP-GAN [65] pre-trained weights, $\theta_w(x)$.

For more efficient learning, the adapter modules consist of three specific adapter weights: ϕ_g, ϕ_p, ϕ_r . Where the first two notations represent the parameters of group-wise and point-wise convolutions, respectively, while ϕ_r represents the task-specific residual parameters. These task-specific weights are combined in parallel, and the whole transformation process is

$$\phi_{w,t}(\theta_w(x)) = \phi_g(\theta_w(x)) + \beta \phi_p(\theta_w(x)) + \phi_r(\theta_{w-1}(x)), \quad (2.7)$$

where $\phi_{w,t}(\theta_w(x))$ is the task specific adaptation on the top of original $\theta_w(x)$ for the task T and θ_{w-1} is the weights of the previous layer. The β is used as a binary selection to enable or disable the point-wise convolution. The pre-trained weights $\theta_w(x)$ remain frozen, and only the task-specific adapter weights are trained for the new task. This ensures that the knowledge of the previous task is not affected.

Style-GAN2

StyleGAN2 [29] is an advanced generative model proposed by Tero Karras et al. in 2019, as an improvement over the original StyleGAN [12]. It is primarily used for generating high-quality images by leveraging the concept of a style-based generator architecture. StyleGAN2 refines many aspects of its predecessor, leads to a more stable training process and improved image quality. The model's core innovation revolves around the manipulation of intermediate feature maps through a "style" input, which provides fine-grained control over the generated images' characteristics, such as pose, lighting, and texture. The StyleGAN2 architecture is shown in Figure 2.4.

The StyleGAN2 architecture consists of two main components: a mapping and a synthesis network. The mapping network transforms a latent vector $z \in \mathbb{R}^l$ into an intermediate latent vector $w \in \mathbb{R}^d$, where d is the dimension of the space in which the style vectors are embedded. The mapping network is expressed as:

$$w = f(z), \quad (2.8)$$

where f is a function consisting of a fully connected multi-layer perceptron.

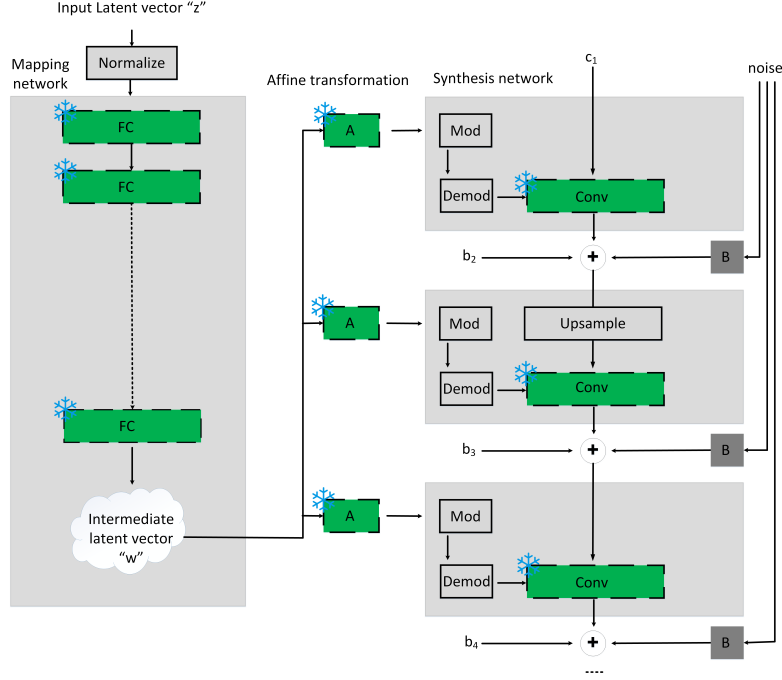


Figure 2.4: StyleGAN2 model architecture. The StyleGAN2 architecture consists of two main components: a mapping and a synthesis network. Where block A is the affine transformation and w represents the intermediate latent vector.

The synthesis network takes the vector w and generates the image progressively through a series of layers. Unlike the standard GAN models, StyleGAN2 incorporates a style input at each convolutional layer to modulate the activations non-linearly. This allows for more precise control over the image features. The generation of an image is a hierarchical process that begins with a low-resolution image and progressively increases the resolution, typically from 4×4 to 1024×1024 . At each layer, the synthesis network applies a style transformation to modify the feature maps

$$\hat{x} = G(w), \quad (2.9)$$

where G is the synthesis network, and $\hat{x}$ is the generated image.

2.3 Continual Learning Approaches

Taking an interesting scenario of human CL ability in DL model is not an easy task, as it tends to overwrite new information on the older, unlike humans [2]. Therefore, DL models struggle with the concept of CL. When a model is trained on a new task, older information tends to get wiped out as the model's weights are updated and leads to a problem known as CF [66]. CL in machine learning is the ability to learn continuously from the stream of data and preserve previously acquired knowledge. The real challenge emerges when the model starts learning new tasks, but its performance on older tasks degrades.

Initially, the concept of saving data in memory is introduced for CL, which allows it to be retrieved and combined during training on new data [5, 67]. However, these approaches aim to save tasks, which require a large memory capacity for storage, and limit the scalability as the amount of data grows. To tackle the CF issue, researchers developed several techniques, which can be grouped into distinct approaches aimed to solve the fundamental problem of forgetting. These techniques can be broadly categorized into replay, regularization, and model expansion approaches [2, 68, 6, 69].

2.3.1 Regularization-Based Methods

Regularization methods add an additional term to the final loss [70, 40, 71, 72] or penalize the gradients [73, 74, 75] of the model to prevent catastrophic forgetting by constraining the model's parameters during training. The regularization term ensures that the model does not drastically change the weights of the network that are important for previously learned tasks. The benefit of these methods is that they typically use the same architecture for subsequent tasks. However, as the number of tasks grows, the model's capacity is exhausted [63]. Many contributions that consider regularization techniques are presented in the literature, both for generative and discriminative models.

Elastic weight consolidation in short EWC [70] is one of the most famous regularization methods, which assigns a penalty to changes in parameters that are important

for previously learned tasks. It ensures that the model does not forget old tasks and minimizes changes to critical parameters. Consider a model already learned a task and is now training on a new task t . So, the objective for the new task in EWC is

$$\mathcal{L}(\theta) = \mathcal{L}_t(\theta) + \lambda \sum_i F_i (\theta_i - \theta_i^*)^2. \quad (2.10)$$

Here, $\mathcal{L}(\theta)$ represents the total loss, which includes loss for the current task $\mathcal{L}_t(\theta)$ and the regularization term. The second term penalizes large changes in the parameters that are important for the old task, where F_i denotes the Fisher information matrix, which measures the importance of each parameter θ_i . The values θ_i^* represent the optimal parameters after learning the previous tasks, and λ is a hyper-parameter that controls the strength of the regularization. By minimizing this loss, the model is encouraged to retain knowledge from previous tasks while learning new ones, thereby prevents CF. The authors in [76, 77] presented improved versions of EWC for CL. Additionally, EWC is also used in GANs [40] for FS image generation.

There are several key contributions to the advancement of regularization techniques in SOTA to avoid CF. The Synaptic Intelligence (SI) approach, presented in [78], measures the importance of model parameters for previous tasks online, incorporates regularization into the loss function while training for the current task to overcome CF. The MAS technique [72], on the other hand, calculates the importance of parameters of the previous tasks based on their sensitivity in an unsupervised and online manner. Meanwhile, the approach presented in [77] combines both SI and MAS to leverage the advantages of both strategies.

2.3.2 Replay Methods

Replay, also known as rehearsal methods, attempts to retain previously learned tasks by periodically revisiting old data while learning from new data. This prevents forgetting to recall the past tasks, and reinforces earlier knowledge [21]. The previous exemplars are either recalled from memory or generated by generative models [79] called pseudo rehearsal. The latter avoids the memory overhead of storing large datasets but still enables knowledge retention by generating data that corresponds to

earlier tasks. The author used GANs [46] to replay previous tasks' samples for classification [79]. Moreover, VAEs [80, 81] and recently diffusion models [82, 83] are also used as a pseudo rehearsal to handle CF. The overview of generative models from the perspective of CL is presented in detail in [84] and [85] provides a summary of diffusion models in CL. The MeRGAN [86] and GAN-Memory [87] are types of replay methods for image generation.

2.3.3 Expansion Approaches

Model expansion methods extend the model to accommodate new tasks without forgetting previous ones. These methods avoid CF by adding new neurons or sub-networks that are specific for new tasks [6]. The fundamental approach is to consider each task as a separate entity, with a distinct model for each task. However, this approach does not allow the new task to leverage previous knowledge and is resource-intensive. Therefore, at the beginning, researchers proposed a solution to add neurons for the new tasks. DEN [88] consists of three steps: retraining, expansion, and duplication, which help incrementally add new tasks. First, it selectively retrains the weights that are necessary for the task using sparsity regularization. In the second step, the network is expanded if the first step's sparsity regularization is not satisfied. Finally, if a neurons drift too far from its original value, it is duplicated. Another approach [89] expands the network and utilizes reinforcement learning to select the most suitable model for the new task.

Recently, in [16], LoRA [59] is added to the base network to mitigate CF in GANs. Additionally, adapter modules in [63] are appended to the base model, which modulate the model's parameters. Several other approaches based on expansion techniques for GANs are presented in [90, 91].

2.4 Few-Shot Approaches

FS learning enables models to classify tasks or generate new objects with only a few training samples [92, 93]. This challenge arises due to the scarcity of data, which contrasts with traditional DL models, where large amounts of data are available for

training [10]. This section reviews and explores FS methodologies [61] used to enhance their effectiveness, specifically focusing on transfer learning (TL), data augmentation, and network architecture.

2.4.1 Transfer Learning

TL or FT [94] is one of the most successful approaches to overcome the problem of limited data in DL models. TL aims to leverage knowledge from a source task or domain with abundant data and apply it to a target task with limited or FS data [95]. The underlying principle is that there are shared features or representations between the source and target tasks, which are transferred to the target model to improve learning efficiency. In such cases, pre-trained models, such as those trained on large-scale datasets like FFHQ [12], ImageNet [96], etc., serve as starting points for FS tasks. These models have already learned useful low-level and high-level feature representations, which are then fine-tuned using only FS examples.

The methods in [3, 97] are proposed for FS classification. Additionally, GANs are extensively explored in the context of FS. TGAN [36] is the first approach to use TL to adapt to the target, which leverages a pre-trained source model. It demonstrates that transferring both modules of the GAN is more effective than transferring only one individual module.

FreezeD [98] employs TL and trains the last few layers and freezes the other layers of the discriminator to enhance FS learning in GANs. Additionally, [99] decomposes pre-trained weights using singular value decomposition (SVD) and only trains the singular values to learn the FS target dataset. Furthermore, the methods in [41, 100, 101, 42] use TL for FS and limited image generation.

2.4.2 Data Augmentation

Data augmentation (DA) plays a pivotal role to enhance FS learning, which increases the size of the training dataset through transformations. DA techniques help to reduce over-fitting and generate more diverse training samples from the available FS data, allow the model to learn a broader range of features and improve generalization

across the target domain. The most widely used DA techniques are rotation, scaling [3], flipping, cropping [102], translation [103], and many more. However, such techniques are less effective for FS learning because they only create variations of a very limited set of examples, which do not provide enough diversity for the model to generalize well [10].

Therefore, many works [104, 105, 106] are presented to advance DA methods for limited data. In [107, 108], augmentation is applied to the input data and regularizes the discriminator’s sensitivity to these augmentations. The regularization term is added to the loss function and encourages it to produce consistent outputs for the augmented and original images. It is shown that this approach performs well in limited image generation. To handle limited data generation, the diffusion process is applied in the standard GAN to prevent over-fitting of the discriminator, resulting in what is known as Diffusion-GAN [109]. In standard training, the job of the discriminator is to classify real and fake images, while in Diffusion-GAN is to distinguish between real and noisy images. A recent survey in [110] provides a detailed review of FS learning, its classification, and recent advancements.

2.4.3 Network Architecture

This kind of method implements a distinct architecture for FS and limited data generation. It typically ensembles pre-trained models such as CLIP [111], enhances feature representation or a dynamic model to improve the model’s ability in case of FS learning [61]. The authors Kumari et al. proposed Vision-aided GAN [112], which ensembles a large pre-trained vision model with GANs. The pre-trained model is used as a feature extractor, along with the original discriminator, to distinguish real and fake images.

Chapter 3

CFTS-GAN: Continual Few-Shot Teacher Student for GANs

3.1 Abstract

Few-shot and continual learning face two well-known challenges in GANs: overfitting and catastrophic forgetting. Learning new tasks results in catastrophic forgetting in deep learning models. In the case of a few-shot setting, the model learns from a very limited number of samples (e.g. 10 samples), which can lead to overfitting and mode collapse. So, this chapter proposes a Continual Few-shot Teacher-Student technique for the generative adversarial network (CFTS-GAN) that considers both challenges together. Our CFTS-GAN uses an adapter module as a student to learn a new task without affecting the previous knowledge. To make the student model efficient in learning new tasks, the knowledge from a teacher model is distilled to the student. In addition, the Cross-Domain Correspondence (CDC) loss is used by both teacher and student to promote diversity and to avoid mode collapse. Moreover, an effective strategy of freezing the discriminator is also utilized for enhancing performance. Qualitative and quantitative results demonstrate more diverse image synthesis and produce qualitative samples comparatively good to very stronger state-of-the-art models.

3.2 Introduction

CL and FS learning are two very important problems to take into account, which affect the applicability of deep learning systems in real cases [6, 10]. Continual learning makes the model capable of learning new tasks without affecting the previously learned tasks. However, learning a new task in deep learning models leads to a well-known problem named catastrophic forgetting [113, 66]. In a few-shot setting, a model learns only from a few samples (e.g. ≤ 10), which can result in memorizing the training set and causes overfitting and mode collapse [61]. FS is also a considerable matter because of the unavailability of the full dataset due the data privacy concerns. Moreover, FS is an important consideration for such applications where very limited samples are available, such as healthcare, cybersecurity, etc. Furthermore, it is also essential due to huge resource consumption, training time, and low-power devices [6, 10].

Individually, continual and few-shot learning are very well explored; conversely, considering both together have gained more popularity recently [61]. So, we also take into consideration both together and present our contributions to the challenging tasks in this chapter.

In CL and FS there are two very distinct areas of application of the techniques: the ones that apply to discriminative models, such as classifiers, segmentation models, etc., and the ones that apply to generative models, such as GAN models. A survey article in [114] collected the works that considered both the CL and FS together for classification and segmentation. Various techniques exist that focus on both few-shot and continual learning to mitigate forgetting and overfitting in classification models. However, generative models still need more attention and consideration regarding continual and few-shot learning and have gained attention recently [61]. Therefore, we consider the CL and FS which apply to generative models, especially to GANs.

The CL approaches are grouped into regularization [40], replay [115], or expansion methods [63, 116, 117]. Regularization approaches grant to overcome forgetting but produce blurred samples after learning many tasks [63]. In replay approaches, memory consideration limits the scalability of these approaches, and data privacy is also

an important consideration where data privacy is concerned [118]. Dynamic architectures add additional parameters for CL, do not need previous data samples, and also provide good results. However, designing such architecture needs careful attention due to increasing the number of parameters [63]. On the other hand, when limited samples are available for training, instead of learning the data distribution, the model memorizes the training samples and leads to overfitting and mode collapse. Earlier approaches for FS and limited data generation used transfer learning and fine-tuning for the target domain generation [119, 120, 121]. Another approach is data augmentation either for the data or feature level for gaining diversity in the target generation [122]. However, these approaches are not a good choice in the case of very limited training samples [123]. In the knowledge distillation, another model is used to make the deep learning model efficient considering the FS setting [124]. However, all these approaches do not consider both the CL and FS image generation. Less work has been done considering both few-shot and continual learning together for GANs [125].

Our work proposes a continual few-shot teacher-student model for GANs (CFTS-GAN) considering the FS and CL together. Our method uses knowledge distillation, adding regularization terms for the few-shot image generation. Our CFTS-GAN model takes inspiration from CAM-GAN [63] for continual learning and from [126] for few-shot learning. CAM-GAN injects adapter modules on top of a generator model [127] for continual image generation. It trains only the adapters when a new task is available, preserving the ability to generate images for the previous tasks. The CAM-GAN consists of a simple generator architecture compared to StyleGAN2 [128], whereas StyleGAN2 has more control over the image generation due to latent space manipulation and adding noise in every stage. We extend the CAM-GAN training with the teacher-student architecture to improve its performance and with the CDC loss to preserve diversity and avoid mode collapse. Our final architecture consists of three generators: a source, a teacher, and a student. Starting from a source generator, previously trained on a large dataset, we train CFTS-GAN in two stages. In the first stage, the teacher model is trained on the current task, preserving the generator diversity with the help of the CDC loss [126] between the source model and itself. In

the second stage, when the student model is trained on the current task, the student takes advantage of the teacher, squeezing teacher knowledge inside the adapters and, at the same time, preserving generator diversity using CDC loss between the source model and itself. By applying CDC loss to both, we decrease the probability that the generators lose the ability to generate different images. CFTS-GAN also utilizes a simple technique of freezing the student’s discriminator to obtain much better results [98]. The quantitative results demonstrate that our approach gains more diversity and obtains comparatively higher-quality samples compared to stronger models [125], [126], [129], and [130], which are derived from a more advanced architecture [128]. The main contributions of this chapter are summarized in the following points.

- We propose a teacher-student model for continual few-shot image synthesis, to condense the knowledge of a generator into the adapters of a CAM-GAN model.
- To preserve the diversity of the image generated and prevent mode collapse due to memorization of few available examples, we employed the Cross-Domain Correspondence (CDC) loss [126] in both the teacher and student models training, introducing a source model pre-trained on a large dataset.
- To further refine the model performance, a simple strategy of freezing the discriminator in the student training is also used [98].
- To evaluate image quality and diversity, the performance of the CFTS-GAN is analyzed on different few-shot datasets using FID and B-LPIPS [125] metrics.

3.3 Literature Review

Continual Learning. Generative models have been recently analyzed to generate data continually and get rid of overriding new information on the older ones. A well-known approach in [70] is proposed for discriminative models where an additional loss term is added to retain the previously learned distribution. The idea of [70] is utilized and implemented in GAN [40] where an additional loss term is added only to

the generator of the conditional GAN to prevent previously learned distribution from drifting and avoid forgetting. However, after too many tasks the model saturates and provides an unrealistic generation of images. Another notable approach is memory replay GAN (MeRGAN) [131] in which a previous sample is recalled during training for the current task. The previous data sample is generated using the generator which solves the issue but still provides unrealistic samples after many tasks. Moreover, rehearsal-based approaches are limited to label conditional generation [132]. To implement both label and image conditional GAN, the authors in [132] proposed an approach using knowledge distillation for lifelong generation. However, the proposed lifelong model is shared across all tasks, so the previous task generation quality is degraded as new tasks arrive. In parameter isolation-based approaches, the parameters of the previous tasks are kept unchanged while learning new tasks. One way of parameter isolation is to expand the network and each task has its own parameters [133, 134]. Another way is to use the same architecture but allocate distinct parameters for each task [91, 135]. For continual classification, a model is designed using universal and parameter vectors for learning shared and task-specific domains, respectively [136]. In [116], add additional layers for learning a new task and use previously learned knowledge, and fine-tune for better convergence. CAM-GAN authors [63] proposed a continual learning approach that is based on adding adapter modules for learning the upcoming tasks without affecting the previously learned distributions. Taking inspiration from them, we extended their model to work on a more challenging scenario of CL and FS, together. We also use their adapters concept on a Teacher-Student setting to enhance generation quality and CDC loss to preserve the diversity of generated images.

Few-Shot Learning. Many methods exist for the few-shot GANs [137, 138, 139]. A simple fine-tuning FreezD method based on freezing the discriminator layers to achieve the few-shot image generation is presented in [98]. Transfer learning is also considered for the few-shot image synthesis in [121]. BSA [105] adds a small number of parameters to the source model based on the statistics of the feature map. CDC [126] preserves the diversity of the images by applying cross-domain correspondence between the source and target images. Other variants of CDC are also presented in

[129] and [130]. Similar to CDC the DCL [130] uses the mutual information between the source and target domains. These approaches consider only few-shot learning, while in our case, we take into consideration both continual and few-shot learning.

Continual Few-Shot Learning. For discriminative models, a hyper-transformer is used for few-shot lifelong learning classification in [140]. The hyper-transformer generates the weights for learning the new task. The authors in [141] proposed another approach for the few-shot class incremental classification using a neural gas network. An expansion-based idea is presented in [142] for few-shot discrimination, which tries to make the same sample closer and increase the space between different samples for few-shot settings. Conversely, generative models in terms of both continual and few-shot settings are not widely explored. According to the authors, the LFS-GAN [125] is the first approach that considers both setups together. LFS-GAN appends additional parameters for learning new tasks. The distance between generated fake samples and input noise is maximized to produce diverse images. A continual few-shot image translation is proposed in [143]. This method introduces new scale and shift parameters and modulates the older parameters to learn the newly introduced scale and shift parameters. We proposed different training approaches for continual few-shot learning utilizing the architecture of [63], introducing a teacher-student architecture. Moreover, we also applied the CDC loss in both training stages to obtain better diversity.

3.4 Method

In this section, we describe our approach for the continual few-shot image generation, which is divided into several subsections. Sections 3.4.1 and 3.4.2 give details of the preliminaries of continual and few-shot learning and CDC loss. Section 3.4.3 describes the teacher-student model and their objectives.

3.4.1 Continual Few-Shot GAN

For continual few-shot image synthesis, a set of multiple tasks $T = \{T_0, \dots, T_t\}$ is considered (Figure 3.1). Each task T_i consists of a dataset with few samples $x_0, x_2, \dots, x_n$,

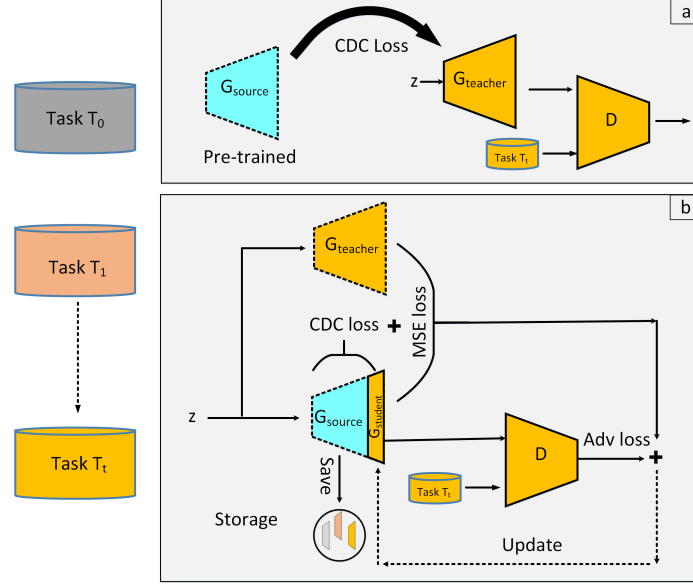


Figure 3.1: The teacher-student model for continual learning GAN. The dashed lines show the frozen model. a) shows the teacher model training. b) demonstrates training of the student model.

where n is the total number of samples in the training set. If n is low (i.e., $n \leq 10$), we are in the case of few-shot learning. As in the CL setting, when the model is trained on a new task T_i , previous training data are no longer available and then can not be used for this training step.

GANs are usually composed of two models: a discriminator D and a generator G . The objective of D is to discern real data from synthetic samples generated by G . While the objective of G is to generate samples that look real to fool D . The generator learns parameters θ from a probability distribution $p_{data}(x)$ for a given dataset to produce synthetic data. These fake images are generated from a random Gaussian distribution $p(z)$. Both D and G are trained together in an adversarial manner to achieve the following objective [144]

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [1 - \log D(G(z))] + R(\theta_d) \quad (3.1)$$

where $p_{\text{data}}(x)$ is the distribution of real data and $p_z(z)$ is a random Gaussian distribution noise, $R(\cdot)$ is a regularization term [127].

For each task T_i , our training is composed of two stages. In the first stage, a teacher model is trained on the FS dataset for task T_i . In the second stage, to exploit the knowledge distillation [145] using the teacher, the student model is trained. To enhance the diversity of the generated examples and avoid mode collapse, the Cross Domain Consistency (CDC) loss is employed for both training, teacher, and student, in addition to adversarial loss, as explained in the following paragraphs.

In the case of FS, due to limited samples, the discriminator part of the GAN also overfits and affects the generator training. So, freezing the discriminator is one of the strategies to avoid the overfitting of it [98].

3.4.2 Cross Domain Consistency Loss

Considering that few-shot learning leads to mode collapse in GANs, data augmentation is one possible solution for diverse image generation but it works better when the training samples are in a much larger number [123]. Since we are in a much more restrictive situation, to preserve the source diversity and avoid mode collapse, we took inspiration from the [126]. During the training process of the generator (teacher or student, depending on whether we are in the first or second stage) on the dataset for the task T_i , another frozen generator, called source, is incorporated into the training procedure.

As previously mentioned, we consider two generators in this scenario: G_{source} , trained on large dataset (in our case, trained on the FFHQ dataset), and G_{target} , which is trained using a few-shot learning method on dataset for the task T_i . At each training step i , two noise vectors z_j and z_k are sampled and passed to both the G_{source} and G_{target} . The probability distribution between different noise vectors for the source and target is given as

$$\begin{aligned}
y_{source} &= \text{Softmax} \left(\left\{ \text{sim} (G_{source} (z_j), G_{source} (z_k)) \right\}_{\forall j \neq k} \right) \\
y_{target} &= \text{Softmax} \left(\left\{ \text{sim} (G_{target} (z_j), G_{target} (z_k)) \right\}_{\forall j \neq k} \right)
\end{aligned} \tag{3.2}$$

where y_{source} and y_{target} are the probability of the source and target, sim denotes the cosine similarity, and $G_{source}(z)$ and $G_{target}(z)$ are the output layer for the source and target generators. We use the KL divergence to encourage the target generator to have a similar distribution of the source generator. This loss is called the CDC loss and is given as:

$$\mathcal{L}_{cdc} (G_{source}, G_{target}) = KL(y_{target} || y_{source}). \tag{3.3}$$

In our case, our target generator G_{target} will be the teacher or the student, depending if we are in the first or second stage. In the following sections, we will use the notation $G_{teacher}$ and $G_{student}$ instead of the target G_{target} for the corresponding teacher and student training.

3.4.3 Teacher Student Model

Our generator model, based on CAM-GAN [63], injects adapter modules on top of GP-GAN [127] for continual learning. The CAM-GAN consists of global weights θ_{global} and adapter weights $\theta_{adapter}$. The latter are used for learning the upcoming tasks while freezing the global weights, so as not to affect the previous tasks generation.

Our continual few-shot image generation training is composed of two stages, and takes into account three generators: source, teacher, and student. All of them have the same architecture, but each one is used in a particular setting and different purpose. At the beginning, the source is trained on a very large dataset. As shown in Figure 3.1(a), in the first stage, the teacher model is cloned from the source model and trained all its weights $\theta_{teacher}$. Because the teacher is also trained in a few-shot setting, to enforce the diversity, we support its training with the CDC loss [126], in addition to its adversarial loss.

As shown in Figure 3.1(b), in the second stage the student model is trained. To achieve the continual image generation objective, we considered the student model weights $\theta_{student}$ as the combination of global weights θ_{global} and adapter weights $\theta_{adapter}$, initially cloned from the source. To learn the current few-shot task T_i , only the adapters' weights are trained, while freezing the global weights. To make the student more effective in learning the current few-shot task T_i , it takes advantage of the source using CDC loss [126], and from the teacher model, previously trained with the same dataset, using a loss of knowledge distillation [145].

Teacher objective. The teacher learning is shown in Figure 3.1(a). The teacher model objective uses the adversarial loss as given in the equation 3.1. For diverse image generation, the CDC loss is used along with the adversarial loss. So, the objective of the teacher is the sum of the adversarial loss and CDC loss for the teacher $G_{teacher}$ and is given by

$$\mathcal{L}_{teacher} = \mathcal{L}_{adv} + w_t \mathcal{L}_{cdc}(G_{source}, G_{teacher}), \quad (3.4)$$

where w_t is a scalar weight factor for the CDC loss. Unlike starting from scratch, the teacher model's weights are initialized with the source parameters.

Student Objective. The training process of the student model is depicted in Figure 3.1(b). To train the student model $G_{student}$, the knowledge from the teacher model $G_{teacher}$ is transferred to the student. To transfer the knowledge from the teacher model to the student, we utilized the Mean Square Error (MSE) loss [145], termed as $\mathcal{L}_{kd}$, applied to the output. This loss minimizes the loss between the teacher and student, and its objective is given by

$$\mathcal{L}_{kd} = \frac{1}{N} \sum_{i=1}^N ||G_{teacher}(z) - G_{student}(z)||^2. \quad (3.5)$$

Moreover, the objective of the student adapter modules is to fool the discriminator, minimizing the standard GAN loss

$$\mathcal{L}_{adv} = \frac{1}{N} \sum_{i=1}^N \log(1 - D(G_{student}(z))), \quad (3.6)$$

where D means the discriminator network of the GANs. Lastly, the $\mathcal{L}_{kd}$ and the

standard GAN loss are combined with the CDC loss to obtain more diversity. The final objective of the student is given by

$$\mathcal{L}_{student} = \mathcal{L}_{adv} + \alpha \mathcal{L}_{kd} + w_s \mathcal{L}_{cdc} \quad (3.7)$$

where α is the weight of the loss $\mathcal{L}_{kd}$ and w_s is the weight of the CDC loss for the student.

3.5 Experiments

Datasets. This section provides details of the performed experiments and shows the qualitative and quantitative results of the CFTS-GAN for continual few-shot image synthesis. The datasets taken into consideration for the experiments are sketches [126], female [146], sunglasses [126], male [146], and babies [126]. These datasets are used as the target datasets for evaluating the efficacy of our model for the few-shot continual generation of images. Each of these datasets consists of 10 samples. While the source model is pre-trained on a large FFHQ dataset [30].

Evaluation metrics. The state-of-the-art CDC [126], RSSA [129], DCL [130], CAM-GAN [63] and LFS-GAN [125] approaches are considered for comparison. The CAM-GAN model appends adapter modules on top of [127] for continual image generation. The LFS-GAN adds more weights for the subsequent tasks using few-shot learning. Where LFS-GAN derives its model from a very strong model StyleGAN2 [128]. Moreover, it also utilizes the patch discriminator [126] to improve the target generation. While others compared state-of-the-art models are also based on a stronger model [128]. We evaluated our CFTS-GAN teacher-student model for the few-shot continual learning, which produces diverse images and has a quality near to the stronger LFS-GAN model. The Fréchet inception distance (FID) [31] and B-LPIPS [125] are used as quantitative metrics. The FID score indicates how close the synthesis images are to the real images. The lower FID means that the generator produces samples closer to the real images. While the B-LPIPS shows the diversity of the images, the higher score shows a more diverse image generation.

Training. The source model is pre-trained on the source task using a large dataset such as the FFHQ dataset. The FFHQ contains 70k high-resolution image samples. The input image dimension fed into the model is 256×256 . The other datasets mentioned above are considered as subsequent tasks. Therefore, the teacher and student models are trained to generate new tasks continually. For the training, only the data from the current task is available, while the previous data is not available. The teacher model is trained on a few-shot images. The knowledge is then transferred from the teacher to the student model using knowledge distillation while also maintaining the source diversity. So the teacher-student model correspondingly produces quality and diverse samples without affecting the previous knowledge.

3.5.1 Qualitative Results

The qualitative results are shown in Figure 3.2. Our model is able to generate different, diverse, and comparatively quality samples of images continually. The model produces the current data samples without affecting previously learned samples. Only a few generated samples from each task are presented.

3.5.2 Quantitative Results

The comparison in terms of FID and B-LPIPS scores with the state-of-the-art models is shown in Table 3.1. The bold value represents the best results while the underlined values represent the second-best results. The CFTS-GAN represents the best performance than CAM-GAN, CDC, RSSA, and DCL in terms of both B-LPIPS and FID, maintaining the same number of weights and same architecture on evaluation as CAM-GAN. Except for the FID of babies, the CDC is the second best. Compared to LFS-GAN, the teacher-student model also gives the best performance in terms of B-LPIPS. In terms of FID, the LFS-GAN [125] performs best, probably because of utilizing a very strong baseline model StyleGAN2 [128].

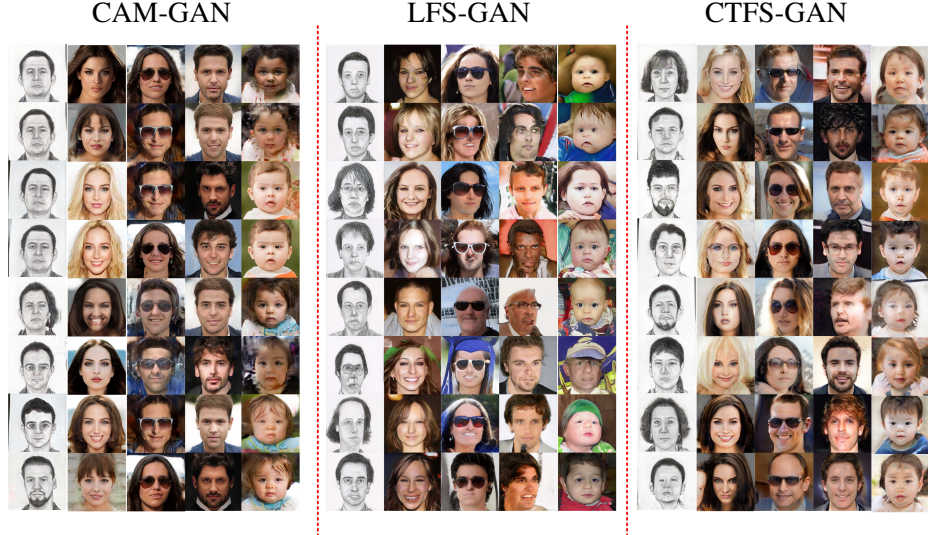


Figure 3.2: Qualitative results: Generated samples. In each group, the first column is for sketches (Task 1), and the second, third, fourth, and fifth are for females (Task 2), sunglasses (Task 3), males (Task 4), and babies (Task 5).

Table 3.1: Quantitative results comparison in terms of FID ($\downarrow$) and B-LPIPS ($\uparrow$) for each task with state-of-the-art methods.

	Sketches (T_1)		Female (T_2)		Sunglasses (T_3)		Male (T_4)		Babies (T_5)		Average	
	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS
CDC [126]	322.72	0.205	197.40	0.427	244.94	0.463	277.00	0.381	<u>69.98</u>	0.454	208.41	0.386
RSSA [129]	308.00	0.285	175.20	0.440	207.58	0.484	205.49	0.405	76.70	0.481	194.59	0.419
DCL [130]	297.73	0.307	170.31	0.435	191.54	0.490	194.42	0.443	77.22	0.487	186.25	0.432
CAM-GAN [63]	91.81	0.293	85.68	0.332	86.81	0.333	82.83	0.312	146.20	0.181	98.66	0.290
LFS-GAN [125]	34.66	<u>0.354</u>	29.59	0.481	27.69	<u>0.584</u>	35.44	<u>0.472</u>	41.48	<u>0.556</u>	33.77	<u>0.489</u>
CFTS-GAN (ours)	<u>82.49</u>	0.399	<u>62.10</u>	0.707	<u>36.03</u>	0.966	<u>66.23</u>	0.760	96.62	1.02	<u>68.69</u>	0.770

3.5.3 Ablation Study

To show the effectiveness of the CFTS-GAN, we analyzed each component of our proposed method. The details for each component are given below.

Effect of CDC on the teacher. We analyzed the effect of CDC loss using different values of w_t which are shown in Table 3.2. From the ablation, it is concluded that

it increases diversity as we give more value to it. However, increasing w_t after some point leads to a higher FID score. So, in our experiments, we use the optimal value of $w_t = 40$ for training the teacher model for all of the tasks. It is because of gaining better FID at this point.

Table 3.2: Ablation study for the teacher using w_t for sunglasses dataset.

w_t	10	20	30	40	50	60	70
FID	79.41	63.70	46.94	46.76	54.21	63.39	65.76
B-LPIPS	0.245	0.516	0.856	0.930	0.943	0.956	0.976

Effect of CDC on the student. We also inspect the CDC loss terms and see its effect on the student model. Giving more weight to the CDC loss obtains more diversity, and less weight leads to less diversity. However, increasing its effect leads to degrading the FID as we analyzed for the teacher. We performed some ablation studies on the sunglasses dataset for different values of w_s which is shown in Table 3.3. Assigning the $w_s = 20$ works better for gaining more diversity with better FID. So, for all the tasks, we used $w_s = 20$ for the student model.

Effect of $\mathcal{L}_{kd}$ on the student. The experiments are also performed by assigning different values to α . We observe that giving the value of 2 leads to better results as shown in Table 3.3. Giving more value to it has more impact on diversity. Assigning greater value to it leads to more diversity and light improvement in the FID. From this ablation study, we found that when $\alpha = 2$, we have better FID. Therefore, we use the mentioned value for the rest of the tasks.

Table 3.3: Ablation study for the student on α and w_s for the sunglasses dataset.

w_s	$\alpha = 0$				$\alpha = 2$				$\alpha = 5$				$\alpha = 10$			
	10	20	30	40	10	20	30	40	10	20	30	40	10	20	30	40
FID	56.35	39.92	40.98	53.6	50.66	37.54	42.07	54.31	50.08	38.55	39.67	47.87	46.59	42.80	39.72	46.94
B-LPIPS	0.588	0.942	0.990	1.01	0.768	0.944	1.02	1.01	0.851	0.993	0.991	1.00	0.917	0.997	1.00	1.01

Freezing student discriminator. We also analyzed the student model by freezing the layers of the discriminator of our CFTS-GAN. We inspect that training the last

24 layers of the discriminator leads to better FID scores and more diverse image generation instead of training all the layers (total layers are 36) of the discriminator. The ablation study for the student model with a frozen discriminator is shown in Table 3.4. For this ablation study, we consider the teacher with the best value as given in Table 3.2. From this analysis, it is concluded that instead of training all the layers, if some portion of the discriminator is kept frozen, it leads to better results.

Table 3.4: Ablation study for the student with freezing discriminator for sunglasses dataset.

Number of last trained layers	6	12	18	24	30	36
FID	51.22	39.20	38.98	36.03	40.38	37.54
B-LPIPS	0.902	0.910	0.922	0.966	0.971	0.944

3.6 Summary

This chapter proposes a continual learning few-shot generative adversarial network CFTS-GAN. The CFTS-GAN considers the challenging tasks of catastrophic forgetting and overfitting problems in GANs. We used the teacher-student model for the challenging task of continual few-shot image generation. For continual learning, the CFTS-GAN uses adapter modules as a student to learn new tasks while preserving previously learned knowledge. The teacher model helps the student to produce better and more diverse image generation. The CDC is used by both the teacher and student to preserve the source diversity and prevent mode collapse. Moreover, we used a simple, effective strategy of freezing the discriminator for further improvements. To show the performance of the CFTS-GAN model, it is analyzed on different datasets. The performance analysis shows better results and produces diverse images than the state-of-the-art models.

3.7 Supplementary Materials: Progressive Modular Expansion

Inspired by [147], in this study, we evaluate the expansion of the adapter modules in CAM-GAN [63], which we refer to as progressive modular expansion (PME), to support CL across multiple tasks. We introduce multiple task-specific adapters in parallel, enable the model to grow the model progressively, and learn new tasks while preserving knowledge of the earlier tasks to avoid CF. PME allows the model to scale for new tasks and maintains performance even as new tasks are added sequentially.

3.7.1 Progressive Modular Expansion

In our implementation, the CAM-GAN framework is expanded with a series of adapter modules as new tasks are encountered. Each adapter module is designed to adjust the generative model’s capacity to learn task-specific features without affecting the previous knowledge.

As discussed in section 3.4.3, the CAM-GAN consists of two building blocks: θ_{global} and adapter weights $\theta_{adapter}$. In our case, we expand the adapter for multiple tasks and learn them sequentially, represented as $\theta_{adapter}^{(i)}$. Where i represents the current task sequence number. The base task is learned by training both θ_{global} and the adapter weights $\theta_{adapter}^{(0)}$. For the subsequent task, the base task and the previous task-specific adapter are kept frozen, and the new task-specific adapter $\theta_{adapter}^{(i=1)}$ is trained, and so on for a total of i tasks. We consider three tasks, so $i = 2$, including the base task. The model for $i = 2$ is shown in Figure 3.3. The new task-specific adapter modules are introduced in one of two ways: either by adding a new adapter for each new task or by taking the weighted sum of all existing adapters to create a more unified representation for the upcoming tasks. Mathematically, both approaches are given as:

$$\theta_{adapter}^{(t)} = \sum_{i=0} \theta_{adapter}^{(i)}, \quad (3.8)$$

and

$$\theta_{adapter}^{(t)} = \sum_{i=0} a_i * \theta_{adapter}^{(i)}, \quad (3.9)$$

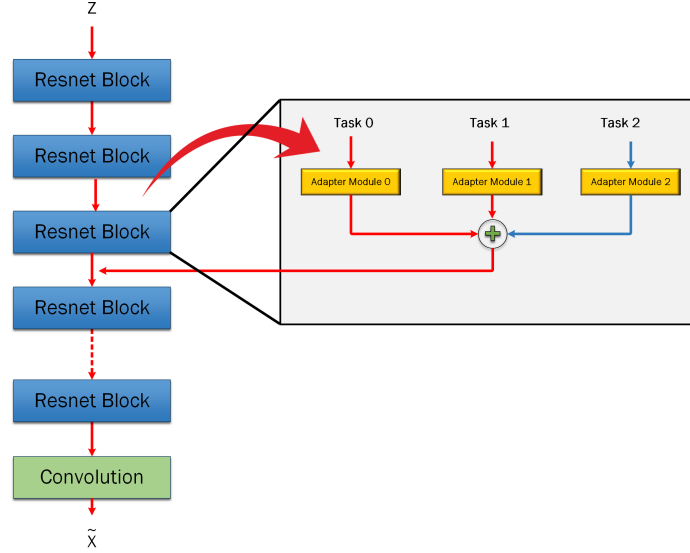


Figure 3.3: Parallel expansion of adapter modules to continually learn subsequent tasks. The red lines indicate the model is already trained on previous tasks, sequentially, while the blue line represents the training of the adapter for the current task.

where a_i is the weight factor for the adapter module and $\theta_{\text{adapter}}^{(t)}$ represents the summation of all adapters.

Each adapter incrementally learns a new task, while the shared base model and all previous adapters remain frozen. After each task is learned, the model switches among tasks and activates the corresponding adapter module, and generates the corresponding task specific images.

3.7.2 Experiments and Results

We evaluated and performed experiments on the progressive modular expansion. In this case, the model is evaluated after training on the limited datasets not on the FS dataset. It is because we follow the experimental setup of CAM-GAN. Therefore, we considered CelebA [148] dataset as a base task, while the Flower [149] and Cathedral [150] datasets are used as subsequent tasks. The FID is used as an evaluation metric.

Quantitative Results: We measured the quantitative performance (shown in Table 3.5) in terms of FID metric, which is most common for generative models. Our approach demonstrates consistent performance across tasks, with no degradation in FID as tasks are added. The task-specific adapters allow the model to maintain good quality generation for both old and new tasks. The FID for PME is worse than CAM-GAN it is because we add the current adapter with the previous ones, which leads to learn from noisy images. Therefore, our model does not converge and produces samples of lower quality compared to CAM-GAN.

Table 3.5: **CL image generation:** comparison with CAM-GAN in terms of FID ($\downarrow$) across two different tasks. The results are taken after learning all the tasks.

Method	CelebA (T_0)	Flower (T_1)	Cathedral (T_2)
	FID	FID	FID
CAM-GAN [63]	5.29	23.0	9.52
Ours	5.29	38.11	21.72
Ours (Wegithed Sum)	5.29	34.69	18.20

Qualitative Results: The visual results are shown in Figure 3.4. The generated outputs demonstrate that the model is able to maintain quality samples for earlier tasks while accurately generates new samples for subsequent tasks. The separation of task-specific knowledge in each adapter module allowed the generative model to adapt to new tasks without degradation of the previous knowledge.

Ablation study for the weight factor a : We analyzed the weighted sum and conducted an ablation study for different values of the weight factor. The results of the ablation study are provided in Table 3.6, where $a = 0.5$ the model performs better. It is because this ensures a balanced contribution of all adapters in the combined representation for each new task. This value indicates equal importance to the contribution of each adapter. As a result, it prevents the model becomes too specific for any one task.

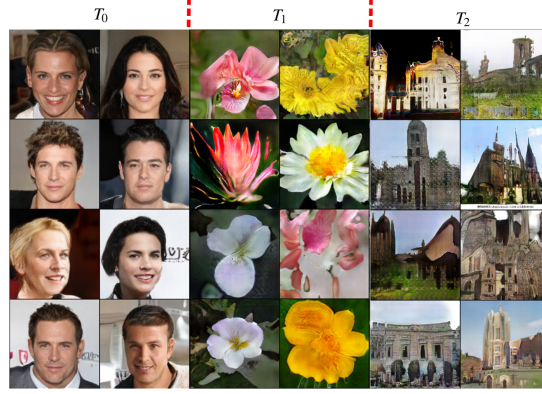


Figure 3.4: Quantitative results of PME. The PME continually generates all the tasks (CelebA, Flower, and Cathedral) without affecting the previous knowledge.

Table 3.6: An ablation study on the weight factor a for the second scenario of PME

a	0.3	0.5	0.7	1
	FID	FID	FID	FID
Flower	36.13	34.68	36.89	38.11
Cathedral	19.39	18.20	20.12	21.72

3.7.3 Discussion

The parallel expansion of adapter modules for subsequent tasks in CAM-GAN provides a solution for CL. By introducing task-specific modules that operate independently, this approach avoids the problem of CF. Additionally, the parallel expansion mechanism allows the model to scale efficiently as more tasks are added and learns them sequentially.

As discussed, PME does not result in better image quality due to the initialization of new tasks' parameters from noisy data. Therefore, in chapter 4, we introduce a more parameter-efficient method by incorporating low-rank tensors. Initially, we initialize one tensor to zero, which results in no changes to the source information, ensure that it is not affected. In this case, the target task leverages the source information as much as possible and converges quickly.

Chapter 4

CoLoR-GAN: Continual Few-Shot Learning with Low-Rank Adaptation in Generative Adversarial Networks

4.1 Abstract

Continual learning (CL) in the context of Generative Adversarial Networks (GANs) remains a challenging problem, particularly when it comes to learn from a few-shot (FS) samples without catastrophic forgetting. Current most effective state-of-the-art (SOTA) methods, like LFS-GAN, introduce a non-negligible quantity of new weights at each training iteration, which would become significant when considering the long term. For this reason, this chapter introduces continual few-shot learning with low-rank adaptation in GANs named CoLoR-GAN, a framework designed to handle both FS and CL together, leveraging low-rank tensors to efficiently adapt the model to target tasks while reducing even more the number of parameters required. Applying a vanilla LoRA implementation already permitted us to obtain pretty good results. In order to optimize even further the size of the adapters, we challenged LoRA limits

introducing a LoRA in LoRA (LLoRA) technique for convolutional layers. Finally, aware of the criticality linked to the choice of the hyperparameters of LoRA, we provide an empirical study to easily find the best ones. We demonstrate the effectiveness of CoLoR-GAN through experiments on several benchmark CL and FS tasks and show that our model is efficient, reaching SOTA performance but with a number of resources enormously reduced. Source code is available on [Github](#).

4.2 Introduction

Recent advancements [151, 12, 29, 152] in GANs show promising results on existing benchmark datasets. However, these models rely on a large amount of data for better performance. Usually, when new data is available for training, the model updates all the parameters because a trains from scratch is needed to preserve old generation capabilities and extend it with the new data. This is not always desirable, because it requires a long training and every time more longer, because the dataset increases during time. Neither, it is a feasible behavior, because not always all the data are available. In the worst case, as in the Continual Learning (CL) scenario, past data is not available anymore and training on new data only exposes the model to the phenomenon known as catastrophic forgetting [17]. Moreover, in many scenarios due to privacy concerns and domains which have limited data such as health, art, etc., only a few samples are available for each task, making it challenging for models to adapt efficiently [114]. Therefore, in Few-Shot (FS) the challenge arises from the few samples (e.g. ≤ 10) available for training which can lead to over-fitting [153].

Unlike discriminative models, GANs are not explored very well in the context of both FS and CL. GAN is known for its ability to generate high-quality synthetic data but faces significant difficulties in handling the evolving nature of tasks. Therefore, considering both FS and CL together is an emerging and challenging area [16]. In this context, the GAN have to learn sequentially with few training samples over time without affecting the previous knowledge. Existing studies such as regularization [41, 100, 101] focus only on FS and fine-tune all the parameters during the adaptation. These approaches are very good to maintain the diversity of the source domain. How-

ever, they do not consider CL and that limit the applicability of generating previously learned representations. On the other hand, many methods [154, 87, 16, 155] focus only on mitigating catastrophic forgetting by expanding the source model. Nevertheless, these models require a large number of parameters for the target tasks, leading to a storage burden to save the target parameters. A recent study [16] proposed a modulation-based efficient approach that considered both FS and CL together and produced high-quality and diverse images. Unfortunately, the adapted parameters are very complex and still need consideration to reduce them. Furthermore, a simple approach with less parameters can be effective for FS, as training with too many parameters can lead to over-fitting [156].

In response to these challenges, this chapter introduces CoLoR-GAN (continual learning with low-rank adaptation in GANs). Our approach takes inspiration from LoRA [59] and appends low-rank parameters to the source model to address the issues of catastrophic forgetting in CL scenarios. Low-rank adaptation facilitates efficient parameter updates and reduces the risk of over-fitting to the target tasks, while enabling the model to effectively learn tasks without forgetting. Furthermore, we extend LoRA methodology to reduce the parameters number in case of convolution, introducing a LoRA in LoRA mechanism. Moreover, because LoRA is usually very sensitive to hyper-parameters selection, we describe an easy approach to select the most appropriate ones. The evaluation of CoLoR-GAN on several benchmark FS datasets demonstrates its effectiveness in mitigating catastrophic forgetting while achieving comparable performance to SOTA methods but with less parameters and less training iterations.

The main contributions of our work are:

- Inspired by LoRA [59], we added the low-rank tensor on top of StyleGAN2 for CL and FS image generation; introducing also a LoRA in LoRA (LLoRA) for convolutions, in order to even further decrease the number of parameters.
- We provide an empirical study to choose the most effective LoRA hyper-parameters in generative CL and FS tasks.
- We evaluated the CoLoR-GAN through experiments on the CL and FS datasets

and demonstrated its performance in terms of quality and diversity compared the state-of-the-art.

4.3 Literature Review

FS Learning: In FS, we are in a situation where data is scarce and we have only few-sample to train a model [36]. Recent advancements leverage ensemble models [47, 157], incorporating techniques such as regularization [70, 158], expanding network [42], or contrastive learning [101, 159] to enhance performance in FS. The CDC [41] introduces a method for FS image generation by leveraging cross-domain correspondence loss. RSSA [100] introduces a relaxed spatial structural consistency loss enables the model to maintain structural information from source images. In [101] the DCL method is proposed with the aim of retaining the diversity of the source domain using the contrastive loss. AdAM [42] is a modulation-based approach. Instead of fine-tuning all the parameters, it introduces additional modulation parameters that handle FS learning, while keeping the source parameters frozen. All these approaches work very well. However, they focus solely on FS learning. In contrast, our method addresses the more challenging task of combining FS learning with CL.

CL: CL in GANs research area addresses the challenge of maintaining knowledge across multiple tasks. Various methods, including rehearsal strategies [86], regularization techniques [70], and architectural expansion [63, 160] exist for CL in GANs. The CAM-GAN [63] introduces a novel CL framework for GANs. The approach leverages adapter modules integrated into the existing architecture [65]. The adapters ensure the learning of new tasks and adapt incrementally. An alternative approach for both discriminative and generative models is presented in [160], introducing efficient parameters for CL. Recently, low-rank adaptation (LoRA) [59] gained popularity in LLMs for CL [161]. It reduces the memory and computation overhead when adapting a pre-trained model to new tasks. A recent approach utilizing LoRA for diffusion models is proposed in [162]. Although these methods achieve promising results, they do not address FS learning. In contrast, our model considers a more challenging task

of FS and CL, offering a distinct advancement over previous approaches.

CL and FS Learning: One of the most challenging task considers both FS and CL together. In this setup, models adapt to new tasks with only a few training samples while simultaneously maintaining the performance on previously learned tasks [17]. The approach for conditional GANs in [113] leverages the replay method and a novel discriminative mode affinity loss to address CL and FS. Another approach introduced in [154] for both FS and CL considers the teacher-student model with CDC loss. A recent approach named LFS-GAN [16] presents an approach that considered both FS and CL in GANs. This method introduces low-rank tensors to learn new tasks while retaining previously acquired knowledge efficiently. Additionally, the approach incorporates a loss function to enhance the diversity of synthesized images. These approaches provide the best results. However, they require more parameters and training loops compared to ours. Unlike LFS-GAN which relied on complex operations (i.e. decomposition and reconstruction of multiple tensors and modulates the pre-trained parameters), we took inspiration from [59] and incorporated simple low-rank tensors and added them to pre-trained weights. In this way, our model gives comparable results and maintains the source information while enabling more efficient training with significantly fewer parameters.

4.4 CoLoR-GAN model

4.4.1 Preliminaries

LoRA Background: Because fully training a Large Language Model (LLM) is very expensive, the LoRA [59] finetuning trick has been proposed to reduce, as much as possible, the number of weights involved in the training without performance penalties. The low-rank adaptation incrementally updates internal pre-trained model weights. Consider a pre-trained weight matrix $W_0 \in \mathbb{R}^{d \times k}$. Instead of modifying the entire weight matrix during incremental fine-tuning, it approximates the update as a low-rank adaptation of two matrices $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$, where r is the rank of the matrix. The adapted weights $\hat{W}$ are given as follows:

$$\hat{W} = W_0 + \Delta W = W_0 + \frac{\alpha}{r} BA \quad (4.1)$$

where ΔW represents the weight updates and α the scaling factor that controls the LoRA influences. Selecting an r much lower than d and k , means that $d \times r + r \times k \ll d \times k$ and therefore that the additional trained parameters can be really small.

CL and FS Definition: In the case of continual learning, different tasks $T = T_1, T_2, \dots, T_n$ are considered to train the model while maintaining performance on previously learned tasks without revisiting them. Let $D_i = \{x_i^t\}_{t=1}^k$ be the dataset for the task T_i . In FS case, the k is limited to a very small number and represents the number of training samples in the dataset for the task T_i . At time i , only the current task is available to train the model. Let W_i represents the model parameters after learning task T_i . The goal of continual learning is to optimize the weights on each task sequentially and preserves the ability to generates all the tasks.

4.4.2 StyleGAN2 LoRA Adaptation

Let's revisit the original StyleGAN2 model, which is very popular for synthesizing images. It consists of two modules: mapping and synthesis network. The first one translates a noise into a latent vector, while the synthesis network transforms the intermediate latent vector into an image. In Fig. 4.1, the StyleGAN2 is shown with dashed green boxes.

On the top of StyleGAN2, we adapted the low-rank tensor as given in Eq. 4.1 into the mapping, and synthesis modules. The LoRA adaptation for CL and FS generation is shown in Fig. 4.1 with solid yellow boxes. The pretrained StyleGAN2 weights are frozen while only the low-rank tensors are trained for the downstream tasks. By freezing the pretrained weights of StyleGAN2 and training only the low-rank tensors, we aim to adapt the model to new tasks while leveraging and retaining the source model's knowledge. Each new task involves new adapted trained weights, which are preserved to save past generation abilities when new tasks are available.

LoRA FC: In the mapping network, we extended each fully connected (FC) layer with a LoRA FC layer. A FC layer is composed of a weight tensor $W_{FC} \in \mathbb{R}^{d_{out} \times d_{in}}$.

where W_{conv} , $\hat{W}_{conv}$ and ΔW_{conv} represent the pretrained frozen convolutional weights, the convolutional weights after LoRA adaptation and the LoRA adaptation weights, respectively. All of them have the shape $\mathbb{R}^{c_{out} \times c_{in} \times k \times k}$. And, α , c_{in} , c_{out} and $k \times k$ are the hyper-parameter to control the target adaptation, the in and out channels, and the kernel size, respectively.

The $B \in \mathbb{R}^{r \times c_{out} \times k \times k}$ and $A \in \mathbb{R}^{r \times c_{in}}$ weights represents the low-rank matrices that are learned for each task. Here, differently from LoRA FC, to enhance model expressivity, we introduce also a non-linearity to compute ΔW_{conv} (a ReLU activation function), because it play a crucial role in significantly enhancing the model's performance and expressivity.

Because the number of weights in B could be relatively high, to reduce its impact we take inspiration from [16], and factorize it in the following way:

$$B = act(B' \times M_{inst}) \quad (4.4)$$

where $B' \in \mathbb{R}^{c_{out} \times r}$ and $M_{inst} \in \mathbb{R}^{r \times r \times k \times k}$ are the low rank matrices that multiplied together are forming the B . Because this low-rank parametrization is a LoRA itself, and is positioned inside a LoRA, we define this architecture as LoRA in LoRA (LLoRA) Conv. It is worth noting that another non-linearity is introduced to enhance model expressivity even further.

Scaling factor: The constant value $\frac{\alpha}{r}$ represents a scaling factor that influences how much the LoRA parameters weigh over the total. Usually, the divisor is set equal to the rank r , and there is no general rule to choose the dividend α . In literature, the choice of this value was explored in other contexts [163, 164, 165], leading to empirical choices. As noted by the authors, final performance is very sensitive to this value. We tested that, automatically learning the value does not always pay. We measured the distance between target and source (L_{s-t}), computing the average of pairwise LPIPS (Learned Perceptual Image Patch Similarity) [166] distance within each cluster. Empirically, we noted that for target domains far from the dataset on which the model was originally pretrained (source dataset), bigger α_{fc} for LoRA FC and smaller α_{conv} for LLoRA Conv work better. We suppose that it is because the FC layer handles a more abstract representation, therefore, it benefits from larger

adaptation while the convolution layers capture more spatial features. From the experiments, we notice that the higher the LPIPS distance from the source, higher α_{fc} than 1 and the lower α_{conv} than 1 work better. In most cases, best α_{fc} is around 1.5 and best α_{conv} around 0.25. More the L_{s-t} is higher and more this is true. In other words, for a dataset far from the source domain, the LoRA weight for the Mapping network need to be amplified and for the Synthesis network need to be attenuated.

4.5 Experiments

Implementation: We followed state-of-the-art [16] and utilized the pre-trained StyleGAN2 [29] on the FFHQ [12] dataset and adopted the low-rank tensors for the target domain generation on the top of StyleGAN2 [29]. The pretrained model remains frozen, while only low-rank tensors are trained to adapt to the target domains. The model is trained using Adam optimizer with a learning rate of 0.002 and a batch size of 4 due to few-shot learning, ensuring efficient updates and steady convergence [16], the training is stopped at 1500 iterations. The loss used to train the GAN is a Wasserstein Loss [167].

Datasets: The experiments are carried out on multiple datasets. The StyleGAN2 pre-trained on FFHQ [12] and the LSUN-Car [150] are considered as a source domain. The target datasets include Sketches, Females, Sunglasses, Males, Babies [96, 41], wrecked car¹, and truck datasets². Each target dataset includes 10 samples for training. These diverse target datasets help assess the performance of our model across different target domains.

Baselines: We considered several state-of-the-art models for comparison that address three distinct aspects of image generation: (i) FS image generation, (ii) CL image generation, and (iii) FS and CL together. The methods CDC [41], RSSA [100], DCL [101], and AdAM [42] belong to the first category, focusing on FS image generation. In contrast, GAN-Memory [87] and CAM-GAN [63] are designed to handle CL. While the LFS-GAN [16] considered both FS and CL similar to ours.

¹The link for wrecked car dataset is [here](#)

²The link for truck dataset is [here](#)

Metrics: The FID [31] and B-LPIPS [16] metrics are used to assess the performance of our model. The FID score measures the similarity between synthesized and real images, indicating that a lower score corresponds to better performance. On the other hand, the B-LPIPS metric evaluates the diversity of the synthesized images, and a higher value represents better results. To evaluate FID, we sampled 5000 images, while 1000 samples are used for B-LPIPS. We followed the same training and evaluation protocol and datasets of LFS-GAN to be as fair as possible.

4.5.1 Few-shot and Continual Learning Evaluation

Qualitative Results: First of all, the CoLoR-GAN model is evaluated for the FS and CL task. This means that, after training the model on the current FS dataset, all previous tasks are evaluated as well to check that the model has not forgotten the past. As shown in Fig. 4.2, our model effectively generates all the previous tasks and maintains quality and performance without affecting the previous information. It demonstrates that CoLoR-GAN is able to keep the learned information from the earlier tasks without degradation or interference like the compared state-of-the-art model.

Quantitative Results: As shown in Table 4.1, TGAN and other FS methods fail to generate quality samples due to their lack of consideration for CL. Therefore, the FID is very higher for these methods. Among FS methods, AdAM performs better than the others because it does not update all the weights during adaptation. On the other hand, CL methods such as GAN-Memory and CAM-GAN yield better results than FS methods by incorporating CL. However, these methods do not account for the FS scenario, leading them to replicate the training data and over-fit. In contrast, our model and LFS-GAN effectively address both FS and CL, producing high-quality and diverse images. Despite LFS-GAN strengths, there is still room for improvement, because our model CoLoR-GAN achieves almost the same performance with half parameters and half iterations. The training efficiency measured by the number of trainable parameters and iterations is shown in Table 4.2.



Figure 4.2: Qualitative results. Comparison of continual image generation of different models. The few-shot GAN is not able to produce the previous samples while lifelong GANs are able to avoid catastrophic forgetting. Each pair of rows shows two samples from each domain.

Table 4.1: **Continual few-shot image generation**: comparison with different methods in terms of FID ($\downarrow$) and B-LPIPS ($\uparrow$) across different tasks. The results of all models are taken after training the model on the last (Babies) task. **Best** and 2nd highest results.

Methods		Sketches (T_1)		Female (T_2)		Sunglasses (T_3)		Male (T_4)		Babies (T_5)		Average	
		FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS
Baseline (TGAN) [36]		372.89	0.157	255.33	0.238	309.13	0.247	281.43	0.129	171.19	0.203	278.02	0.195
Few-shot GANs	CDC [41]	322.72	0.205	197.40	0.427	244.94	0.463	277.00	0.381	69.98	0.454	208.41	0.386
	RSSA [100]	308.00	0.285	175.20	0.440	207.58	0.484	205.49	0.405	76.70	0.481	194.59	0.419
	DCL [101]	297.73	<u>0.307</u>	170.31	0.435	191.54	0.490	194.42	<u>0.443</u>	77.22	0.487	186.25	0.432
	Adam [42]	161.48	0.250	179.69	0.342	217.19	0.352	163.87	0.299	110.08	0.407	166.82	0.330
Continual learning GANs	GAN-Memory [87]	69.58	0.311	71.56	0.287	87.02	0.169	99.44	0.143	177.73	0.150	101.05	0.212
	CAM-GAN [63]	91.81	0.293	85.68	0.332	86.81	0.333	82.83	0.312	146.20	0.181	98.66	0.290
Continual and Few-shot GANs	LFS-GAN [16]	34.66	0.354	29.59	0.481	<u>27.69</u>	0.584	<u>35.44</u>	0.472	41.48	0.556	33.77	0.489
	ours	<u>45.03</u>	0.281	<u>35.30</u>	<u>0.448</u>	25.93	<u>0.493</u>	34.40	0.422	<u>43.83</u>	<u>0.526</u>	<u>36.89</u>	<u>0.434</u>

Table 4.2: Comparison on training efficiency. TPs, ITs: number of trainable parameters and number of iterations to learn a new task of each method, respectively. Second column represents the percentage with respect to StyleGAN2.

Method		TPs (M)	% w.r.t. backbone	ITs
Baseline (TGAN) [36]		30.0	100%	-
Few-shot GANs	CDC [41]	30.0	100%	5K
	RSSA [100]	30.0	100%	-
	DCL [101]	30.0	100%	3K
	AdAM [42]	18.9	63.0%	-
Continual Learning GANs	GAN-Memory [87]	5.3	17.7%	-
	CAM-GAN [63]	2.3	7.70%	-
Continual and Few-shot GANs	LFS-GAN [16]	0.1	0.36%	3K
	our	0.05	0.18%	1.5K

Table 4.3: Ablation study for the rank.

rank	# of Trainable Param.	Sketches (T_1)		Female (T_2)		Sunglasses (T_3)		Male (T_4)		Babies (T_5)		Average	
		FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS
1	54K	45.56	0.280	35.30	0.448	25.93	0.493	36.19	0.422	47.20	0.527	37.91	0.431
2	108K	50.59	0.163	29.96	0.449	32.66	0.482	37.77	0.439	52.38	0.430	40.62	0.306
4	216K	50.64	0.188	30.52	0.415	43.29	0.475	37.12	0.399	53.82	0.526	43.07	0.295
8	432K	49.74	0.110	36.75	0.355	58.40	0.468	36.32	0.424	59.26	0.470	48.09	0.365

4.5.2 Ablation Study:

Rank of adapted tensors: In Table 4.3, we analyze the model on different ranks r for all the datasets. For most of the case, rank 1 provides in average the best results. This result was predictable if we consider that we are in a limited-data regime. The limited number of weights to be trained allows to take under control the overfitting effects. It worth to note that, on female dataset, the rank 2 increases the number of parameters to the level of LFS-GAN, but delivering also the same performance.

Activation function: We evaluated the model’s performance with respect to the activation function. As demonstrated in Table 4.4, they plays a significant role in enhancing the performance.

Scaling factor: Table 4.5 shows results for the ablation on α value. Similar to [168], we used fixed values for alpha in both parts. On the bottom of the Table 4.5, we show LPIPS $L_{s \rightarrow t}$ distance between the Task T_i and the original dataset FFHQ, on which the StyleGan2 was pretrained. This LPIPS value is then used to select α_{fc} and α_{conv} . It

Table 4.4: Ablation study for the activation function.

Activation	Sketches (T_1)		Female (T_2)		Sunglasses (T_3)		Male (T_4)		Babies (T_5)		Average	
	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS
$\times$	57.32	0.238	40.74	0.424	50.00	0.457	45.52	0.421	70.76	0.463	52.86	0.400
$\checkmark$	45.56	0.280	35.30	0.448	25.93	0.493	36.19	0.422	47.20	0.527	37.91	0.431

Table 4.5: Ablation for α based on LPIPS distance, L_{s-t} represents LPIPS between source and target. Last row shows LPIPS value between source and target.

α_{fc}	α_{conv}	Sketches (T_1)		Female (T_2)		Sunglasses (T_3)		Male (T_4)		Babies (T_5)		Average	
		FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS
L_{s-t}	L_{t-t}	49.81	0.284	35.30	0.448	29.91	0.469	34.40	0.422	76.27	0.489	48.73	0.422
$2L_{s-t}$	$L_{s-t}/2$	47.37	0.280	38.43	0.469	27.01	0.433	39.07	0.409	50.76	0.507	40.52	0.419
$3L_{s-t}$	$L_{s-t}/3$	46.12	0.278	38.16	0.417	26.14	0.468	42.54	0.389	48.95	0.520	40.83	0.414
$4L_{s-t}$	$L_{s-t}/4$	45.03	0.281	38.90	0.394	25.93	0.493	42.64	0.385	43.83	0.526	39.26	0.415
$5L_{s-t}$	$L_{s-t}/5$	50.03	0.272	38.01	0.410	37.73	0.482	42.83	0.378	48.64	0.501	43.44	0.408
LPIPS distance		0.735		0.253		0.451		0.309		0.531			

Table 4.6: Ablation study for the LoRA tensors in mapping and synthesis part.

LoRA FC	LLoRA Conv	Sketches (T_1)		Female (T_2)		Sunglasses (T_3)		Male (T_4)		Babies (T_5)		Average	
		FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS
$\checkmark$	$\times$	50.03	0.217	41.69	0.384	27.62	0.488	51.42	0.398	50.92	0.410	44.33	0.379
$\times$	$\checkmark$	55.04	0.230	38.37	0.427	46.84	0.492	52.88	0.418	76.18	0.484	53.86	0.397
$\checkmark$	$\checkmark$	45.56	0.280	35.30	0.448	25.93	0.493	36.19	0.422	47.20	0.527	37.91	0.431

worth noting that, more the Task T_i dataset differs from FFHQ the bigger α_{fc} works better for FC, and the smaller α_{conv} works better for convolutions. Best performances are usually obtained by assigning $\alpha_{fc} > 1$ (≈ 1.5) and $\alpha_{conv} < 1$ (≈ 0.25).

LoRA FC and Conv: Furthermore, the adapted tensors in both parts are analyzed separately. We compare generation performance of our model with only LoRA FC or with only LLoRA Conv. As shown in Table 4.6, it is clear that it performs more effectively when we add LoRA in both parts.

Additional Experiments: Taking the LSUN-Car as a source, we also performed experiments on the wrecked car and truck datasets. Our model generates comparable results to LFS-GAN. It is worth noting that our model beats LFS-GAN in the truck images, as shown in the right part of Table 4.7, using α_{fc} of 1.626 and α_{conv} of 0.1, confirming our intuition.

Table 4.7: Ablation for α based on LPIPS distance from LSUN-car (used as source dataset for pretraining), and comparison with LFS-GAN.

α_{fc}	α_{conv}	Wrecked cars (T_1)		Truck (T_2)	
		FID	B-LPIPS	FID	B-LPIPS
L_{q-t}	L_{q-t}	330.7	0.232	221.1	0.229
$2L_{q-t}$	$L_{q-t}/2$	355.4	0.154	226.2	0.201
$3L_{q-t}$	$L_{q-t}/3$	356.9	0.143	230.6	0.185
$4L_{q-t}$	$L_{q-t}/4$	584.6	0.000	301.1	0.000
LFS-GAN		314.7	0.251	217.5	0.244
L_{q-t}		0.692		0.813	

α_{fc}	α_{conv}	Wrecked cars (T_1)		Truck (T_2)	
		FID	B-LPIPS	FID	B-LPIPS
L_{q-t}	L_{q-t}	330.7	0.232	221.1	0.229
$2L_{q-t}$	L_{q-t}	322.5	0.220	220.2	0.204
$4L_{q-t}$	L_{q-t}	329.5	0.202	222.8	0.189
$8L_{q-t}$	L_{q-t}	337.3	0.180	228.1	0.180

α_{fc}	α_{conv}	Wrecked cars (T_1)		Truck (T_2)	
		FID	B-LPIPS	FID	B-LPIPS
$2L_{q-t}$	$L_{q-t}/2$	355.4	0.154	226.2	0.201
$2L_{q-t}$	$L_{q-t}/4$	344.8	0.128	219.6	0.206
$2L_{q-t}$	$L_{q-t}/8$	344.6	0.198	214.4	0.215
$2L_{q-t}$	$L_{q-t}/10$	334.8	0.220	214.5	0.210

4.5.3 Few-shot Evaluation

Qualitative Results: Figure 4.3 illustrates the generated images of sketches, sun-glasses, female, and male by different models as well as by our model. As depicted in the figure, our model generates quality images while maintaining information of the source images compared to other state-of-the-art methods.

Table 4.8: **FS image generation:** comparison with different methods in terms of FID ($\downarrow$) and B-LPIPS ($\uparrow$) across different tasks. The results are taken after learning each task.

Method		Sketches (T_1)		Female (T_2)		Sunglasses (T_3)		Male (T_4)		Babies (T_5)		Average	
		FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS	FID	B-LPIPS
Baseline (TGAN [36])		60.57	0.335	67.63	0.318	72.81	0.391	73.44	0.319	114.29	0.414	77.75	0.355
FS GANs	CDC [41]	49.19	0.237	31.26	0.450	36.03	0.505	41.88	0.435	64.75	0.496	44.62	0.425
	RSSA [100]	56.25	0.251	34.24	0.467	44.01	0.501	44.83	0.434	72.45	0.486	50.36	0.430
	DCL [101]	58.60	0.353	35.19	0.468	33.05	0.517	44.19	0.436	66.10	0.508	47.43	0.456
	AdAM [42]	45.70	0.325	61.79	0.375	45.55	0.392	61.55	0.338	91.13	0.421	61.14	0.370
CL GANs	GAN-Memory [87]	69.58	0.311	71.56	0.287	87.02	0.169	99.44	0.143	177.73	0.150	101.05	0.212
	CAM-GAN [63]	91.81	0.293	85.68	0.332	86.81	0.333	82.83	0.312	146.20	0.181	98.66	0.290
CL+FS	LFS-GAN [16]	34.66	0.354	29.59	0.481	27.69	0.584	35.44	0.472	41.48	0.556	33.77	0.489
GANs	ours	45.03	0.281	35.30	0.448	25.93	0.493	34.40	0.422	43.83	0.526	36.89	0.434

Quantitative Results: Table 4.8 exposes the results of the individual training considering only the FS scenario. That is, each model is evaluated only on the current task, not taking into account the past (without CL). This justifies because now the

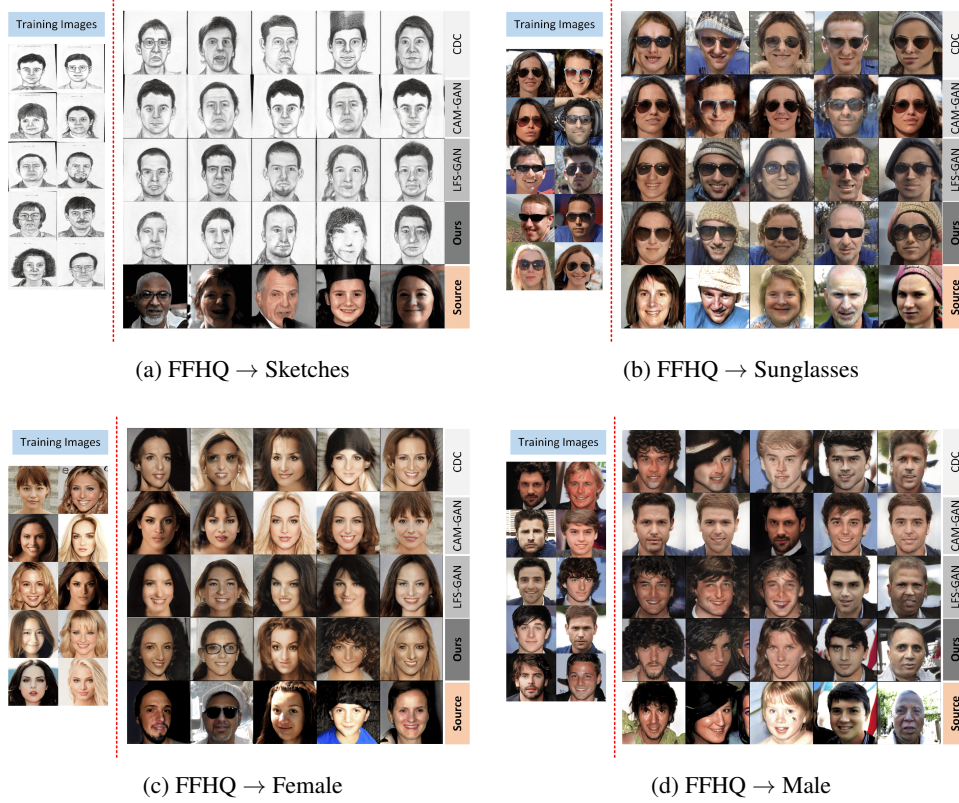


Figure 4.3: **FS samples:** A qualitative evaluation of CoLoR-GAN and other models, generating Sketches, Female, Sunglasses, and Male in FS scenario. Comparatively, our model preserves the source domain information.

FS methods obtain better performance than the methods designed to work in a CL scenario. As LFS-GAN, our CoLoR-GAN beat almost and always outperforms all FS methods. Probably thanks to the minimum quantity of trained weights, CoLoR-GAN manages to avoid overfitting better than other FS-focused methods where the network is trained almost completely [59].

4.6 Summary

This chapter presents a CL and FS image generation approach called CoLoR GAN by integrating LoRA into StyleGAN2. To better handle the problem of parameters reduction, we challenged LoRA limits introducing a LoRA in LoRA (LLoRA) technique for convolutional layers which further effectively break up its weights. Finally, aware of the LoRA sensitivity to hyper-parameters choice, we proposed a simple rule to find them. Our method enables efficient adaptation to target tasks with only few examples while preserving the quality and diversity of the source domain. Our experiments demonstrated that our model has competitive performance compared to SOTA but with half parameters and half training iterations.

Chapter 5

Conclusion and Future Work

This chapter summarizes the key findings and contributions of our thesis. We emphasized the significance of CL and FS learning in DL models, along with their broad classification. Furthermore, we shed light on CF and overfitting as critical challenges in GANs, the main focus of our thesis. Both issues significantly limit performance and generalization capabilities in GANs.

To address these challenges, we proposed two approaches. The first approach utilizes a teacher-student model, where the student incrementally learns new tasks. This method generates good quality and diverse samples while continually learning from FS datasets. It outperforms a strong baseline in terms of diversity and provides samples of comparable quality. The second approach employs parameter-efficient adaptation, allows us to achieve nearly similar results with only half the number of parameters and training iterations compared to state-of-the-art models. So our process is more computationally efficient without compromising performance too much.

5.1 Future Directions

As we know, CL and FS learning are not explored very well in GANs. Therefore, there are several directions for future research.

Distant Target Domain. Our proposed approaches, as well as other existing ap-

proaches, generally consider that the source and target domains are similar. This means that the data distribution in a source domain and a target domain shares certain common characteristics, such as feature similarity. As a result, most CL and FS learning techniques provide good performance as they are designed to handle scenarios where the shift between the source and target domains is less [61].

However, the challenge of dealing with far-domain scenarios, where the source and target domains are significantly different, is not widely explored in the context of CL and FS learning. In a far domain, the model faces a greater challenge because the knowledge learned in one domain may not transfer effectively to a very different domain, for example faces $\rightarrow$ flowers. This leads to poor generalization and low-quality image generation, where the model struggles to adapt to new tasks due to a large domain shift.

Dynamic Model Size. Most expansion techniques rely on fixed parameters for new tasks in GANs. Dynamic expansion is explored in discriminative models [6, 169], but remains underexplored in GANs. One possible future work is to develop dynamic expansion methods for GANs, particularly when tackle tasks that differ significantly from the source domain.

In scenarios where tasks are considerably dissimilar, additional parameters may be required to improve generated image quality. In our case, such as sketches, which differ significantly from the FFHQ compared to the female. As shown in Table 4.1 and Figure 4.3, our model achieves a comparable and better FID score for domains that are near to the source. However, for the sketches, our model performs less effectively compared to others. The higher FID score in this case highlights the limitations of relying on fixed parameters for tasks that differ significantly from the source.

To address this challenge, dynamically adjust parameters based on some metrics e.g., similarity of the target with the source. This would likely lead to more efficient learning and improved performance, particularly to generate higher quality samples across distant target tasks.

Bibliography

- [1] Alessandro Achille, Matteo Rovere, and Stefano Soatto. Critical learning periods in deep neural networks. *arXiv preprint arXiv:1711.08856*, 2017.
- [2] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.
- [3] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.
- [4] Hassan Gharoun, Fereshteh Momenifar, Fang Chen, and Amir H Gandomi. Meta-learning approaches for few-shot learning: A survey of recent advances. *ACM Computing Surveys*, 56(12):1–41, 2024.
- [5] Anthony Robins. Catastrophic forgetting in neural networks: the role of rehearsal mechanisms. In *Proceedings 1993 The First New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*, pages 65–68. IEEE, 1993.
- [6] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [7] Yutao Yang, Jie Zhou, Xuanwen Ding, Tianyu Huai, Shunyu Liu, Qin Chen, Yuan Xie, and Liang He. Recent advances of foundation language models-

- based continual learning: A survey. *ACM Computing Surveys*, 57(5):1–38, 2025.
- [8] Iqbal H Sarker. Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. *SN computer science*, 2(6):1–20, 2021.
- [9] Jiang Lu, Pinghua Gong, Jieping Ye, Jianwei Zhang, and Changshui Zhang. A survey on machine learning from few samples. *Pattern Recognition*, 139:109480, 2023.
- [10] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.
- [11] Joyce Henry, Terry Natalie, and Den Madsen. Pix2pix gan for image-to-image translation. *Research Gate Publication*, pages 1–5, 2021.
- [12] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.
- [13] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.
- [14] Canan Koç, Fatih Özyurt, and Lazslo Barna Iantovics. Survey on latest advances in natural language processing applications of generative adversarial networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 15(1):e70004, 2025.
- [15] Purwono Purwono, Annastasya Nabila Elsa Wulandari, Alfian Ma’arif, and Wael A Salah. Understanding generative adversarial networks (gans): A review. *Control Systems and Optimization Letters*, 3(1):36–45, 2025.

- [16] Juwon Seo, Ji-Su Kang, and Gyeong-Moon Park. Lfs-gan: Lifelong few-shot image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11356–11366, 2023.
- [17] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.
- [18] Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23638–23647, 2024.
- [19] Zhiyuan Chen and Bing Liu. *Lifelong machine learning*. Morgan & Claypool Publishers, 2018.
- [20] Alexander Gepperth and Barbara Hammer. Incremental learning algorithms and applications. In *European symposium on artificial neural networks (ESANN)*, 2016.
- [21] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.
- [22] Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bishan Yang, Justin Betteridge, Andrew Carlson, Bhavana Dalvi, Matt Gardner, Bryan Kisiel, et al. Never-ending learning. *Communications of the ACM*, 61(5):103–115, 2018.
- [23] Alan M Turing. *Computing machinery and intelligence*. Springer, 2009.
- [24] Juyang Weng, James McClelland, Alex Pentland, Olaf Sporns, Ida Stockman, Mriganka Sur, and Esther Thelen. Autonomous mental development by robots and animals. *Science*, 291(5504):599–600, 2001.

- [25] Sebastian Thrun and Sebastian Thrun. *Explanation-based neural network learning*. Springer, 1996.
- [26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- [27] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.
- [28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [29] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.
- [30] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.
- [31] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.
- [32] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048. Makuhari, 2010.

- [33] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.
- [34] Reza Shadmehr and Sandro Mussa-Ivaldi. *Biological learning and control: how the brain builds representations, predicts events, and makes decisions*. Mit Press, 2012.
- [35] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.
- [36] Yaxing Wang, Chenshen Wu, Luis Herranz, Joost Van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. Transferring gans: generating images from limited data. In *Proceedings of the European conference on computer vision (ECCV)*, pages 218–234, 2018.
- [37] Mohammadreza Iman, Hamid Reza Arabnia, and Khaled Rasheed. A review of deep transfer learning and recent advancements. *Technologies*, 11(2):40, 2023.
- [38] Yisheng Song, Ting Wang, Puyu Cai, Subrota K Mondal, and Jyoti Prakash Sahoo. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Computing Surveys*, 55(13s):1–40, 2023.
- [39] Claudio Filipi Gonçalves Dos Santos and João Paulo Papa. Avoiding overfitting: A survey on regularization methods for convolutional neural networks. *ACM Computing Surveys (Csur)*, 54(10s):1–25, 2022.
- [40] Ari Seff, Alex Beatson, Daniel Suo, and Han Liu. Continual learning in generative adversarial nets. *arXiv preprint arXiv:1705.08395*, 2017.
- [41] Utkarsh Ojha, Yijun Li, Jingwan Lu, Alexei A Efros, Yong Jae Lee, Eli Shechtman, and Richard Zhang. Few-shot image generation via cross-domain corre-

- spondence. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10743–10752, 2021.
- [42] Yunqing Zhao, Keshigeyan Chandrasegaran, Milad Abdollahzadeh, and Ngai-Man Man Cheung. Few-shot image generation via adaptation-aware kernel modulation. *Advances in Neural Information Processing Systems*, 35:19427–19440, 2022.
- [43] Ngoc-Trung Tran, Viet-Hung Tran, Ngoc-Bao Nguyen, Trung-Kien Nguyen, and Ngai-Man Cheung. On data augmentation for gan training. *IEEE Transactions on Image Processing*, 30:1882–1897, 2021.
- [44] Yunus Saatci and Andrew G Wilson. Bayesian gan. *Advances in neural information processing systems*, 30, 2017.
- [45] Ngoc-Bao Nguyen, Keshigeyan Chandrasegaran, Milad Abdollahzadeh, and Ngai-Man Cheung. Re-thinking model inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16384–16393, 2023.
- [46] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [47] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022.
- [48] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.
- [49] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- [50] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent

- embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4183–4192, 2020.
- [51] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.
- [52] Lin Wang and Kuk-Jin Yoon. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE transactions on pattern analysis and machine intelligence*, 44(6):3048–3068, 2021.
- [53] Abdolmaged Alkhulaifi, Fahad Alsahli, and Irfan Ahmad. Knowledge distillation in deep learning and its applications. *PeerJ Computer Science*, 7:e474, 2021.
- [54] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020.
- [55] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020.
- [56] Zhenyu Wang, Yali Li, Ye Guo, Lu Fang, and Shengjin Wang. Data-uncertainty guided multi-phase learning for semi-supervised object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4568–4577, 2021.
- [57] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [58] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov.

- Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- [59] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
- [60] Zeeshan Ahmad, Zain ul Abidin Jaffri, Meng Chen, and Shudi Bao. Understanding gans: Fundamentals, variants, training challenges, applications, and open problems. *Multimedia Tools and Applications*, pages 1–77, 2024.
- [61] Milad Abdollahzadeh, Touba Malekzadeh, Christopher TH Teo, Keshigeyan Chandrasegaran, Guimeng Liu, and Ngai-Man Cheung. A survey on generative modeling with limited data, few shots, and zero shot. *arXiv preprint arXiv:2307.14397*, 2023.
- [62] Achraf Oussidi and Azeddine Elhassouny. Deep generative models: Survey. In *2018 International conference on intelligent systems and computer vision (ISCV)*, pages 1–8. IEEE, 2018.
- [63] Sakshi Varshney, Vinay Kumar Verma, PK Srijith, Lawrence Carin, and Piyush Rai. Cam-gan: Continual adaptation modules for generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:15175–15187, 2021.
- [64] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019.
- [65] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.

- [66] Xiang Song, Kuang Shu, Songlin Dong, Jie Cheng, Xing Wei, and Yihong Gong. Overcoming catastrophic forgetting for multi-label class-incremental learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2389–2398, 2024.
- [67] Alexander Gepperth and Cem Karaoguz. A bio-inspired incremental learning architecture for applied perceptual problems. *Cognitive Computation*, 8(5):924–934, 2016.
- [68] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost Van De Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2022.
- [69] Zhenyi Wang, Enneng Yang, Li Shen, and Heng Huang. A comprehensive survey of forgetting in deep learning beyond continual learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [70] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.
- [71] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. *Advances in neural information processing systems*, 30, 2017.
- [72] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–154, 2018.

- [73] David Lopez-Paz and Marc’Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.
- [74] Xu He and Herbert Jaeger. Overcoming catastrophic interference using conceptor-aided backpropagation. In *International Conference on Learning Representations*, 2018.
- [75] Arslan Chaudhry, Marc’Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.
- [76] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International conference on machine learning*, pages 4528–4537. PMLR, 2018.
- [77] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pages 532–547, 2018.
- [78] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017.
- [79] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.
- [80] Martin Mundt, Iuliia Pliushch, Sagnik Majumder, Yongwon Hong, and Visvanathan Ramesh. Unified probabilistic deep continual learning through generative replay and open set recognition. *Journal of Imaging*, 8(4):93, 2022.

- [81] Guido M Van de Ven and Andreas S Tolias. Generative replay with feedback connections as a general strategy for continual learning. *arXiv preprint arXiv:1809.10635*, 2018.
- [82] Mohammad Rostami, Soheil Kolouri, and Praveen K Pilly. Complementary learning for overcoming catastrophic forgetting using experience replay. *arXiv preprint arXiv:1903.04566*, 2019.
- [83] Paweł Skierś and Kamil Deja. Joint diffusion models in continual learning. *arXiv preprint arXiv:2411.08224*, 2024.
- [84] Timothée Lesort, Hugo Caselles-Dupré, Michael Garcia-Ortiz, Andrei Stoian, and David Filliat. Generative models from the perspective of continual learning. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.
- [85] Michał Zając, Kamil Deja, Anna Kuzina, Jakub M Tomczak, Tomasz Trzciński, Florian Shkurti, and Piotr Miłoś. Exploring continual learning of diffusion models. *arXiv preprint arXiv:2303.15342*, 2023.
- [86] Chenshen Wu, Luis Herranz, Xialei Liu, Joost Van De Weijer, Bogdan Raducanu, et al. Memory replay gans: Learning to generate new categories without forgetting. *Advances in neural information processing systems*, 31, 2018.
- [87] Yulai Cong, Miaoyun Zhao, Jianqiao Li, Sijia Wang, and Lawrence Carin. Gan memory with no forgetting. *Advances in Neural Information Processing Systems*, 33:16481–16494, 2020.
- [88] Jeongtae Lee, Jaehong Yoon, Eunho Yang, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *CoRR*, 2017.
- [89] Ju Xu and Zhanxing Zhu. Reinforced continual learning. *Advances in neural information processing systems*, 31, 2018.
- [90] Mengyao Zhai, Lei Chen, Jiawei He, Megha Nawhal, Frederick Tung, and Greg Mori. Piggyback gan: Efficient lifelong learning for image conditioned

- generation. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pages 397–413. Springer, 2020.
- [91] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 67–82, 2018.
- [92] Michael Fink. Object classification from a single example utilizing class relevance metrics. *Advances in neural information processing systems*, 17, 2004.
- [93] Li Fei-Fei, Robert Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.
- [94] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. doi:[10.1109/TKDE.2009.191](https://doi.org/10.1109/TKDE.2009.191).
- [95] SJ—Yang Pan. Q.: A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.
- [96] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [97] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.
- [98] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Freeze the discriminator: a simple baseline for fine-tuning gans. *arXiv preprint arXiv:2002.10964*, 2020.
- [99] Esther Robb, Wen-Sheng Chu, Abhishek Kumar, and Jia-Bin Huang. Few-shot adaptation of generative adversarial networks. *arXiv preprint arXiv:2010.11943*, 2020.

- [100] Jiayu Xiao, Liang Li, Chaofei Wang, Zheng-Jun Zha, and Qingming Huang. Few shot generative model adaption via relaxed spatial structural alignment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11204–11213, 2022.
- [101] Yunqing Zhao, Henghui Ding, Houjing Huang, and Ngai-Man Cheung. A closer look at few-shot image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9140–9150, 2022.
- [102] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5822–5830, 2018.
- [103] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.
- [104] Tianlong Chen, Yu Cheng, Zhe Gan, Jingjing Liu, and Zhangyang Wang. Data-efficient gan training beyond (just) augmentations: A lottery ticket perspective. *Advances in Neural Information Processing Systems*, 34:20941–20955, 2021.
- [105] Atsuhiko Noguchi and Tatsuya Harada. Image generation from small datasets via batch statistics adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2750–2758, 2019.
- [106] Mengyu Dai, Haibin Hang, and Xiaoyang Guo. Adaptive feature interpolation for low-shot image generation. In *European Conference on Computer Vision*, pages 254–270. Springer, 2022.
- [107] Augustus Odena, Han Zhang, Honglak Lee, and Zizhao Zhang. Consistency regularization for generative adversarial networks. In *International Conference on Learning Representations*, 2020.

- [108] Zhengli Zhao, Sameer Singh, Honglak Lee, Zizhao Zhang, Augustus Odena, and Han Zhang. Improved consistency regularization for gans. In *Proceedings of the AAAI conference on artificial intelligence*, 2021.
- [109] Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-gan: Training gans with diffusion. In *International Conference on Learning Representations*, 2023.
- [110] Wu Zeng and Zheng-ying Xiao. Few-shot learning based on deep learning: A survey. *Mathematical Biosciences and Engineering*, 21(1):679–711, 2024.
- [111] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.
- [112] Nupur Kumari, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Ensembling off-the-shelf models for gan training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10651–10662, 2022.
- [113] Cat P Le, Juncheng Dong, Ahmed Aloui, and Vahid Tarokh. Mode-aware continual learning for conditional generative adversarial networks. *arXiv preprint arXiv:2305.11400*, 2023.
- [114] Songsong Tian, Lusi Li, Weijun Li, Hang Ran, Xin Ning, and Prayag Tiwari. A survey on few-shot class-incremental learning. *Neural Networks*, 169:307–324, 2024.
- [115] Xingyu Li, Bo Tang, and Haifeng Li. Adaer: An adaptive experience replay approach for continual lifelong learning. *Neurocomputing*, 572:127204, 2024.
- [116] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

- [117] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Life-long learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017.
- [118] Timothée Lesort, Hugo Caselles-Dupré, Michael Garcia-Ortiz, Andrei Stoian, and David Filliat. Generative models from the perspective of continual learning. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.
- [119] Abulikemu Abuduweili, Xingjian Li, Humphrey Shi, Cheng-Zhong Xu, and Dejing Dou. Adaptive consistency regularization for semi-supervised transfer learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6923–6932, 2021.
- [120] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.
- [121] Yaxing Wang, Chenshen Wu, Luis Herranz, Joost Van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. Transferring gans: generating images from limited data. In *Proceedings of the European conference on computer vision (ECCV)*, pages 218–234, 2018.
- [122] Zhendong Wang, Yifan Jiang, Huangjie Zheng, Peihao Wang, Pengcheng He, Zhangyang Wang, Weizhu Chen, Mingyuan Zhou, et al. Patch diffusion: Faster and more data-efficient training of diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.
- [123] Syed Muhammad Israr and Feng Zhao. Customizing gan using few-shot sketches. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2229–2238, 2022.
- [124] Keon-Hee Park, Kyungwoo Song, and Gyeong-Moon Park. Pre-trained vision and language transformers are few-shot incremental learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23881–23890, 2024.

- [125] Juwon Seo, Ji-Su Kang, and Gyeong-Moon Park. LFS-GAN: Lifelong Few-Shot Image Generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11356–11366, 2023.
- [126] Utkarsh Ojha, Yijun Li, Jingwan Lu, Alexei A Efros, Yong Jae Lee, Eli Shechtman, and Richard Zhang. Few-shot image generation via cross-domain correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10743–10752, 2021.
- [127] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.
- [128] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.
- [129] Jiayu Xiao, Liang Li, Chaoferi Wang, Zheng-Jun Zha, and Qingming Huang. Few shot generative model adaption via relaxed spatial structural alignment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11204–11213, 2022.
- [130] Yunqing Zhao, Henghui Ding, Houjing Huang, and Ngai-Man Cheung. A closer look at few-shot image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9140–9150, 2022.
- [131] WU Chenshen, L HERRANZ, LIU Xialei, et al. Memory replay GANs: Learning to generate images from new categories without forgetting [C]. In *The 32nd International Conference on Neural Information Processing Systems, Montréal, Canada*, pages 5966–5976, 2018.
- [132] Mengyao Zhai, Lei Chen, Frederick Tung, Jiawei He, Megha Nawhal, and Greg Mori. Lifelong gan: Continual learning for conditional image generation.

- In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2759–2768, 2019.
- [133] Abhishek Kumar, Sunabha Chatterjee, and Piyush Rai. Bayesian structural adaptation for continual learning. In *International Conference on Machine Learning*, pages 5850–5860. PMLR, 2021.
- [134] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2021.
- [135] Jathushan Rajasegaran, Munawar Hayat, Salman H Khan, Fahad Shahbaz Khan, and Ling Shao. Random path selection for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.
- [136] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8119–8127, 2018.
- [137] Yuxuan Duan, Li Niu, Yan Hong, and Liqing Zhang. Weditgan: Few-shot image generation via latent space relocation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 1653–1661, 2024.
- [138] Jiachen Shi, Wenzhen Liu, Guoqiang Zhou, and Yuming Zhou. Autoinfo gan: Toward a better image synthesis gan framework for high-fidelity few-shot datasets via nas and contrastive learning. *Knowledge-Based Systems*, 276:110757, 2023.
- [139] Vadim Sushko, Ruyu Wang, and Juergen Gall. Smoothness similarity regularization for few-shot gan adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7073–7082, 2023.
- [140] Max Vladymyrov, Andrey Zhmoginov, and Mark Sandler. Few-shot incremental learning using hypertransformers. *NA*, NA.

- [141] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12183–12192, 2020.
- [142] Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, Liang Ma, Shiliang Pu, and De-Chuan Zhan. Forward compatible few-shot class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9046–9056, 2022.
- [143] Pei Chen, Yangkang Zhang, Zejian Li, and Lingyun Sun. Few-shot incremental learning for label-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3697–3707, 2022.
- [144] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.
- [145] Angeline Aguineldo, Ping-Yeh Chiang, Alex Gain, Ameya Patil, Kolten Pearson, and Soheil Feizi. Compressing gans using knowledge distillation. *arXiv preprint arXiv:1902.00159*, 2019.
- [146] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.
- [147] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *CoRR*, abs/1606.04671, 2016. URL: <http://arxiv.org/abs/1606.04671>, [arXiv:1606.04671](https://arxiv.org/abs/1606.04671).
- [148] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

- [149] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008.
- [150] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.
- [151] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018.
- [152] Haorui Song, Yong Du, Tianyi Xiang, Junyu Dong, Jing Qin, and Shengfeng He. Editing out-of-domain gan inversion via differential activations. In *European Conference on Computer Vision*, pages 1–17, 2022.
- [153] Chenxi Zheng, Bangzhen Liu, Huaidong Zhang, Xuemiao Xu, and Shengfeng He. Where is my spot? few-shot image generation via latent subspace optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3272–3281, 2023.
- [154] Munsif Ali, Leonardo Rossi, and Massimo Bertozzi. Cfts-gan: Continual few-shot teacher student for generative adversarial networks. In *International Conference on Pattern Recognition*, pages 249–262. Springer, 2025.
- [155] Mengyao Zhai, Lei Chen, and Greg Mori. Hyper-lifelonggan: Scalable lifelong learning for image conditioned generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2246–2255, 2021.
- [156] Haoyu Dong and Jianhong Shun. Low-rank adaptation for scalable fine-tuning of pre-trained language models. *preprints.org*, 2025.
- [157] Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected gans converge faster. *Advances in Neural Information Processing Systems*, 34:17480–17492, 2021.

- [158] Liang Hou. Regularizing label-augmented generative adversarial networks under limited data. *IEEE Access*, 11:28966–28976, 2023.
- [159] Hyuk-Gi Lee, Gi-Cheon Kang, Chang-Hoon Jeong, Han-Wool Sul, and Byoung-Tak Zhang. C3: Contrastive learning for cross-domain correspondence in few-shot image generation. In *Proceedings of Workshop on Controllable Generative Modeling in Language and Vision (CtrlGen) at NeurIPS*, volume 2021, 2021.
- [160] Vinay Kumar Verma, Kevin J Liang, Nikhil Mehta, Piyush Rai, and Lawrence Carin. Efficient feature transformations for discriminative and generative continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13865–13875, 2021.
- [161] Junhao Zheng, Shengjie Qiu, Chengming Shi, and Qianli Ma. Towards lifelong learning of large language models: A survey. *arXiv preprint arXiv:2406.06391*, 2024.
- [162] Martin Wistuba, Prabhu Teja Sivaprasad, Lukas Balles, and Giovanni Zappella. Continual learning with low rank adaptation. *arXiv preprint arXiv:2311.17601*, 2023.
- [163] Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, et al. Lora learns less and forgets less. *arXiv preprint arXiv:2405.09673*, 2024.
- [164] Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with lora. *arXiv preprint arXiv:2312.03732*, 2023.
- [165] Christophe Tribes, Sacha Benarroch-Lelong, Peng Lu, and Ivan Kobzyev. Hyperparameter optimization for large language model instruction-tuning. *arXiv preprint arXiv:2312.00949*, 2023.

-
- [166] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.
 - [167] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.
 - [168] Fabian Paischer, Lukas Hauzenberger, Thomas Schmied, Benedikt Alkin, Marc Peter Deisenroth, and Sepp Hochreiter. One initialization to rule them all: Fine-tuning via explained variance adaptation. *arXiv preprint arXiv:2410.07170*, 2024.
 - [169] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *6th International Conference on Learning Representations, ICLR 2018*, 2018.

