ADVANCE: ADVERSARIAL COLLABORATIVE LEARNING FOR DETECTION AND VERIFICATION
OF ARTIFICIALLY CREATED EXAMPLES

A Thesis
presented to
the Faculty of the Graduate School
at the University of Missouri-Columbia

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
CHLOE JONES
Dr. Chi-Ren Shyu, Thesis Supervisor
JULY 2023
The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

ADVANCE: ADVERSARIAL COLLABORATIVE LEARNING FOR DETECTION AND VERIFICATION OF ARTIFICIALLY CREATED EXAMPLES

presented by Chloe Jones,
a candidate for the degree of Master of Science in Computer Science,
and hereby certify that, in their opinion, it is worthy of acceptance.

Professor Chi-Ren Shyu

Professor Dong Xu

Professor Ilker Ersoy
I would like to take this opportunity to express my gratitude and deep appreciation to all those who have supported me throughout my journey in completing this master’s thesis. Their unwavering encouragement, guidance and contributions have been instrumental in shaping my research and academic growth.

Foremost, I extend my sincerest thanks to my esteemed advisors, Dr. Chi-Ren Shyu and Dr. Dan Lin. Their exceptional expertise, insightful guidance and unwavering mentorship and support have been instrumental in every aspect of my graduate career. Their dedication to higher education inspires me to be a lifelong learner, specifically in the field of computer science and cybersecurity.

I am immensely grateful to the members of my committee, Dr. Dong Xu and Dr. Ilker Ersoy. I appreciate you for taking the time to be a part of this monumental moment in my educational career. I would like to thank you for your discussions, assistance and encouragement throughout this process. You all challenge me to be a better researcher and student.

I would also like to extend my appreciation to my peers and professors who have been an incredible source of support and inspiration throughout my academic journey. Their encouragement, intellectual discussions and shared passion for knowledge have had a profound impact on my growth as a researcher and student.

Furthermore, I am deeply thankful to the National Science Foundation (NSF) and their Scholarship for Service (SFS) Award NSF DGE-1946619. This funding helped make my academic aspirations a reality. I am particularly grateful to the advisors who serve on the board of the SFS at the University of Missouri and for their investment in my educational pursuits. Your support has not only provided financial assistance but has also instilled in me a commitment to public service and the advancement of cybersecurity.

Lastly, I would like to express my gratitude to all those who have supported me along this journey, including friends, family and loved ones. Their unwavering belief in my abilities, encouragement during challenging times and understanding of the demands of my research have been invaluable.

Completing this master’s thesis has been a profound and rewarding experience, made possible
by the support and contributions of the individuals and organizations mentioned above, as well as countless others who have touched my life in meaningful ways. I am truly grateful for their presence in my academic endeavors.

In closing, I am honored to have had the opportunity to undertake this research and contribute to the body of knowledge in my field. I am excited about the potential impact of my work and look forward to future opportunities for further growth and exploration in this field.
# TABLE OF CONTENTS

ACKNOWLEDGEMENTS ii

TABLE OF CONTENTS iv

LIST OF TABLES v

LIST OF ILLUSTRATIONS viii

ABSTRACT ix

1 INTRODUCTION 1

2 Literature Review 4
   2.1 Adversarial Learning . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
   2.2 Generative Modeling . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
   2.3 Deepfake Generators . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
   2.4 Deepfake Detectors . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

3 Methodology 16
   3.1 Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
   3.2 Methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
   3.3 Experimental Settings . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24
   3.4 Experimental Results and Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31

4 Conclusion and Future Work 49

BIBLIOGRAPHY 59
LIST OF TABLES

1 Summary of Adversarial Attacks Using Adversarial Examples. Application represents the type of system to be attacked, the study represents a specific study, the method is the method of attack used, the attack type represents a white-box or black-box attack, and the frequency shows if an attack is iterative or only happens one at a time. 6

2 Summary of Countermeasures for Adversarial Examples. The defense type categorizes the defense methods as reactive or proactive, the study represents a real-world study, the application shows the type of defense method, and the attack method shows which methods this application defends against. 7

3 System parameters used for training the StyleGAN3 generator. Note these parameters were utilized per the original paper recommendations. 30

4 System Parameters used for training the CNNDetection detector. Note these parameters were utilized per the original paper recommendations. 30
**LIST OF ILLUSTRATIONS**

1. A conceptual representation of the reactive (left) and proactive (right) arms races for machine learning systems in computer security. [1, 2] ........................................ 5

2. Example of how FGSM works to generate adversarial examples. Left: a clean image of a dog; middle: the perturbation; right: one sample adversarial image, classified incorrectly as a panda. .................................................. 6

3. Example of adversarial attacks on facial/object detection systems using adversarial glasses. .................................................. 7

4. Overview of a vanilla GAN’s two-network architecture featuring a generator and discriminator .................................................. 9

5. Taxonomy of GANs. The taxonomy has three divisions for classifying the contribution each GAN has provided: architecture, conditioning goal and adversarial loss. ........ 11

6. Vanilla CNN deepfake detection system that consists of an input layer, convolution layer and pooling layer for feature extraction, and a fully-connected MLP that works as a classifier to determine whether an images is real or fake. .................. 13

7. Taxonomy of Deepfake Detection Methods using CNNs .................................................. 14

8. High-level overview of my system, ADVANCE. ADVANCE uses an iterative approach to simulate an arms race between two systems. ................................. 17

9. Low-level details of ADVANCE translated from the high-level overview including the taxonomy of variables .................................................. 18

10. StyleGAN3’s architecture. Left: full architecture including the input latent vector \( z \), the mapping network of fully-connected layers, the associated vector weights \( w \), the synthesis network which maps \( z \) to the fake images as output; middle: an expansion of the synthesis network showing how the style layers and ToRGB layer maps the input to the output of fake images; right: the discriminator architecture within the GAN consisting of convolutional layers featuring skip connections and down sampling. .. 19
11 ResNet50 architecture used in CNN Detection to classify images as real or fake.
12 FFHQ Dataset utilized within training and representing the real image dataset.
13 Overview of the iterative process within ADVANCE alternating between training the
   generator (left) and the detector (right).
14 Generator benchmark data: Left: Real images for visual comparison; Right: fake
   images initially trained with only real images as a baseline FID score.
15 Round 1: Establishing the benchmark data set trained on 70k real FFHQ images.
   Step 1: Train the generator. Step 2: generate 10k fake images. Step 3: Input 10k real
   and 10k fake images into the detector. Step 4: Evaluate the detector results. Step
   5: Filter images. This figure shows a sample of the 10k fake images generated. FID
   score: 4.5. Images that fooled the detector: 621. Detector Accuracy: 94.59%.
16 Round 2: First training round on fake images. Step 1: Train the generator on 621
   filtered images from Round 1. Step 2: generate 10k fake images. Step 3: Input 10k
   real and 10k fake images into the detector. Step 4: Evaluate the detector results. Step
   5: Filter images. This figure shows a sample of the 10k fake images generated. FID
   score: 38.4. Images that fooled the detector: 2,620. Detector Accuracy: 75.41%.
17 Round 3: Step 1: Train the generator on 2,620 filtered images from Round 2. Step
   2: generate 10k fake images. Step 3: Input 10k real and 10k fake images into the
   detector. Step 4: Evaluate the detector results. Step 5: Filter images. This figure
   shows a sample of the 10k fake images generated. FID score: 9.4. Images that fooled
   the detector: 2,620. Detector Accuracy: 68.40%.
18 Round 4: Step 1: Train the generator on 2620 filtered images from iteration 2. Step
   2: generate 10k fake images. Step 3: Input 10k real and 10k fake images into the
   detector. Step 4: Evaluate the detector results. Step 5: Filter images. This figure
   shows a sample of the 10k fake images generated. FID score: 8.4. Images that fooled
   the detector: 7,927. Detector Accuracy: 64.24%.
19 Round 5: Retrain the detector. Step 1: Train the detector on 10k fake images from
   Round 4. Step 2: Evaluate the detector results by inputting 10k real images and 10k
   fake images from Round 3. Detector Accuracy: 100%.
20 Round 6: Train the generator on 7,927 filtered images from Round 4. Step 2: generate
   10k fake images. Step 3: Input 10k real and 10k fake images into the detector. Step
   4: Evaluate the detector results. This figure shows a sample of the 10k fake images
   generated. FID score: 4.0. Detector Accuracy: 96%.
21 Detector results represented by the detector’s accuracy vs. round. These results were as expected and imply the ability to conduct an arms race.

22 Generator results represented by the FID score vs. rounds/training iterations. These results were as expected and imply the maintaining of realism throughout the generated fake image distribution.

23 Rounds 2-3 & 6 generated image evolution for visual analysis. This allows us to see how the images change in realism despite continuous training on fake images.

24 Round 1-4, images that fooled the detector. These results show us the type of images that fooled the detector and potentially point out features that distinguish fake from real images.

25 Round 6, more samples of images that fooled the detector.
ADVANCE: ADVERSARIAL COLLABORATIVE LEARNING FOR DETECTION AND VERIFICATION OF ARTIFICIALLY CREATED EXAMPLES

Chloe Jones
Dr. Chi-Ren Shyu, Thesis Supervisor

ABSTRACT

Adversarial learning methods have gained significant popularity in generating deceptive yet convincingly authentic data. While these techniques have proven beneficial for advancing artificial intelligence, they also give rise to a pressing concern regarding the authenticity of information consumed by the general public, exemplified by the prevalence of deepfakes. Consequently, various approaches have been proposed to detect adversarial generated data, aiming to address this challenge. However, a significant proportion of these innovations rely on non-iterative feedforward designs, leading to the overarching concept of an arms race between machine learning and detection systems. In the context of my research, I introduce ADVANCE: Adversarial Collaborative Learning For Detection And Verification Of Artificially Created Examples, an adversarial recognition pipeline that embraces a continuous and collaborative learning paradigm to facilitate an arms race between a deepfake generator and deepfake detector. Specifically, given a generative system and a detection system, this research aims to discover whether it is possible to create a stable pipeline to look for an endpoint to this arms race. These experiments embody a computational analysis of the pipeline and ultimately, results indicate that it is stable enough to be used for further research in improvements of the detection of adversarial generated data.
Chapter 1

INTRODUCTION

In recent years, adversarial learning methods have gained popularity for generating fake yet highly realistic data. Specifically, Generative Adversarial Networks (GANs) have emerged as powerful frameworks capable of producing data that can deceive human perception [3, 4, 5]. GANs have found successful applications in various domains, including text and art generation, as well as supporting data augmentation in research [6, 7, 8]. However, the potential for malicious use of adversarial networks presents a significant challenge, as the authenticity of information consumed by the average person is threatened. Consequently, there has been a surge of research efforts focusing on the detection of adversarially generated data [9]. However, as machine learning techniques advance, the methods of attack also evolve, leading to an ongoing arms race between machine learning and detection systems [10].

The remarkable progress in adversarial learning methods, particularly GANs, has sparked both excitement and concern, specifically in the domain of deepfake generation. On one hand, the ability to generate deepfakes has opened new possibilities for artificial intelligence applications, creative expression and research advancements [11, 12]. On the other hand, the misuse of these techniques poses serious threats to the integrity and trustworthiness of information. Deepfakes are a “manipulation of existing media (image, video and/or audio) or generation of new (synthetic) media using deep learning-based approaches” [13] and serve as a prominent example of the risks associated with adversarial generated data. Deepfakes have raised concerns in areas such as privacy, misinformation and trust in digital media [14]. Therefore, it is imperative to develop robust detection mechanisms to mitigate the potential harm caused by malicious actors leveraging this technology.

The emergence of deepfakes in 2017 brought significant attention to the dangers associated with generative technologies, particularly in the form of maliciously generated content [15]. Deepfakes utilize advanced machine learning algorithms to create highly realistic but fabricated audiovisual
content that can convincingly impersonate individuals or fabricate events. The impact of deepfakes on society has been substantial with instances of blackmail through the creation of fake pornographic material, bots creating fake profiles to influence election outcomes and the spread of false information online [16].

To combat the growing threat of deepfakes, researchers have focused on developing detection methods that can distinguish between real and manipulated content. Current works in the field of adversarial detection primarily rely on machine learning algorithms and techniques: feature-based approaches extract distinctive features from the data and utilize classification models to identify anomalies, pixel-based methods focus on analyzing pixel-level inconsistencies or artifacts introduced during the generation process and model-based approaches leverage statistical analysis and domain-specific models to detect irregularities in the data distribution [17, 18, 19, 20].

Despite the progress made in detecting adversarial generated data, the need for more robust and iterative systems remains crucial. Adversarial attacks continue to evolve, leading to the development of more sophisticated generation methods that can bypass existing detection mechanisms. Furthermore, existing detection systems often suffer from high false positive rates, limited generalization and vulnerability to adversarial attacks themselves [21]. Therefore, there is a pressing need for continuous innovation and collaborative approaches to enhance the detection and mitigation of adversarial generated data.

In this thesis, I propose ADVANCE: Adversarial Collaborative Learning For Detection And Verification Of Artificially Created Examples, a novel continuous and collaborative learning pipeline for adversarial data detection by simulating an arms race between a deepfake generator and a deepfake detector. The objective is to investigate the possibility of creating a stable pipeline that can contribute to the discovery of the end of the arms race between to systems. The pipeline operates by iteratively improving both the generator and the detector. This reciprocal feedback loop aims to enhance the detection capabilities of the system by using the generator’s ability to generate more convincing adversarial data.

The proposed pipeline introduces a continuous and collaborative learning paradigm that facilitates an ongoing interaction between the generator and the detector. Through this iterative feedback loop, the generator learns to produce more sophisticated and challenging adversarial data and the detector adapts to recognize and discriminate these adversarial samples more effectively. The collaboration between the generator and the detector creates a dynamic and evolving environment that simulates the arms race between generative and detection systems.

To implement the continuous and collaborative learning pipeline, I use a hybrid architecture that
combines state-of-the-art deep learning models for both generation and detection. The generator utilizes a GAN-based architecture to generate synthetic data samples that resemble real data. The detector, on the other hand, employs advanced machine learning techniques to classify the input data as real or adversarial. The outputs generated by the generator serve as training data for the detector, while the feedback from the detector guides the generator's learning process. This bidirectional information flow allows both systems to improve iteratively, pushing the boundaries of detection and generation capabilities.

The primary objective of this research is to address the critical need for more robust and iterative systems for detecting adversarial generated data. To achieve this, I have taken the following steps in my research:

- Conducted a comprehensive literature review of adversarial learning methods, covering the underlying principles, current state-of-the-art techniques, challenges and proposed approaches for adversarial data detection and generation. This review provides an understanding of the research landscape and highlights the strengths and weaknesses of existing methods.

- Determined the methodology employed in ADVANCE, providing a detailed explanation of the system's architecture, components and training procedures. This section elucidates the design choices, model architectures and evaluation metrics used in this research.

- Presented the results of the experiments, including a quantitative and qualitative evaluation of the pipeline's performance. This section includes an analysis of the accuracy, robustness and efficiency of the system in detecting adversarial generated data.

- Concluded by summarizing my findings, discussing the implications of this work and providing recommendations for future research in the field of adversarial data detection. This section will include a reflection on the challenges encountered during the research, identifying potential limitations and suggestions for avenues for further exploration and improvement.

By exploring ADVANCE, this research aims to contribute to the development of more robust systems capable of detecting adversarial generated data. This research has broader implications for enhancing security, trustworthiness and integrity in an era where the arms race between generative and detection systems continues to evolve.
Chapter 2

LITERATURE REVIEW

2.1 Adversarial Learning

Adversarial machine learning has emerged as a crucial field, particularly in the context of Generative Adversarial Networks (GANs). However, in relation to this research, its impact extends to not only GANs, but also encompassing security. This section aims to explore adversarial learning in the context of security by conducting a literature review, laying the foundation for my research methodology and experiments.

Adversarial learning can be best understood through the lens of security, often compared to an arms race between system designers and adversaries [10]. By focusing on understanding attacks and defenses against machine learning algorithms, adversarial learning enables researchers to enhance the robustness and security of these models. At its core, adversarial learning involves the interplay between an adversary and a system. The adversary iteratively challenges the system, leading to improvements for both parties. This concept can be categorized into two models: reactive learning and proactive learning, as depicted in Fig. 1. While this concept originates from the context of security, it extends to all areas where adversarial learning is employed.

The reactive approach follows an iterative process where the adversary analyzes classifier defenses and devises attack strategies to overcome them, while the designer reacts by updating the classifier to counter these novel attacks. However, to achieve enhanced system security, the proactive approach, known as security by design, should be considered. Security by design entails anticipating adversary behavior and developing countermeasures before deploying the model, making it more challenging for adversaries to discover and exploit vulnerabilities [10].

Adversarial attacks gained significant attention in 2014 when Szegedy et al. [22], followed by subsequent works [23, 24, 25], exposed the vulnerability of machine learning models to misclassifying
adversarial examples. Adversarial examples are slightly perturbed instances from the data distribution and serve as the primary method for carrying out adversarial attacks. This awareness enabled researchers to uncover blind spots in their training algorithms, leading to the integration of adversarial examples in network training to enhance resistance against attacks.

Although adversarial learning is commonly associated with this 2014 discovery, it has been evolving independently since 2004. In 2004, Dalvi et al. [26] and subsequently Lowd and Meek [27, 28] explored adversarial learning in the context of spam filtering. Lowd and Meek demonstrated how malicious actors can deceive linear classifiers used for spam email filtering by creating adversarial examples of spam emails through careful content modification to evade detection. Since then, extensive research has been conducted to develop attacks against machine learning models during both training and testing phases, systematic methodologies for security evaluation against these attacks and design effective defense mechanisms to mitigate threats.

Now that the concept of adversarial learning and its application in security have been established, let us delve into the current literature on adversarial attacks and defenses. Adversarial attacks can be classified into two categories: white-box attacks and black-box attacks. White-box attacks assume full access to the target model, while black-box attacks only allow querying the input and output of the network, or even having no knowledge of the network [29]. It is crucial to defend against both types of attacks when designing a system. Table 1 provides a summary of various adversarial attacks utilizing adversarial examples that have influenced this research area.

1. FGSM (Fast Gradient Sign Method): FGSM was initially proposed by Goodfellow et al. as a general technique to generate adversarial examples for neural networks [23]. By calculating the gradient after forward propagation with respect to the dataset, FGSM shows that you can perturb the data ever so slightly to maximize loss and fool the detection system. Lowd and
Reinforcement Learning [30] FGSM White-box & Black-box One-time  
Generative Modeling [31] Feature Adversary White-box Iterative  
Face Recognition [32] Feature Adversary White-box One-time  
Object Detection [33] Data Poisoning White-box Iterative  
[34] Data Poisoning Black-box Iterative  
Malware Detection [35] GAN Black-box Iterative  
[36] GAN White-box Iterative  

Table 1: Summary of Adversarial Attacks Using Adversarial Examples. Application represents the type of system to be attacked, the study represents a specific study, the method is the method of attack used, the attack type represents a white-box or black-box attack, and the frequency shows if an attack is iterative or only happens one at a time.

Meek extended this method to the domain of spam emails by applying reinforcement learning techniques [27, 28]. Huang et al. utilized these two studies to show that adversarial attacks are also effective when targeting neural network policies. Specifically, they show existing adversarial example crafting techniques can be used to significantly degrade test-time performance of trained policies by adding small perturbations on the input of the policy by calculating the gradient of the cross-entropy loss function [30].

Figure 2: Example of how FGSM works to generate adversarial examples. Left: a clean image of a dog; middle: the perturbation; right: one sample adversarial image, classified incorrectly as a panda.

2. Adversarial Examples for Generative Modeling: Kos et al. proposed a method to generate adversarial examples for generative models [31]. They demonstrated that by adding perturbations to the input image of the encoder in an autoencoder, they could misguide the autoencoder to generate a targeted adversarial image. This approach leverages the decoder of an autoencoder to generate the adversarial example.

3. Adversarial Attacks on Facial/Object Detection Systems: Illustrated in Fig. 3, S. Zhao et al. utilized a feature adversary method to implement adversarial eyeglass frames within a dataset to evade facial recognition systems [33]. They used a softmax loss function to generate the
adversarial examples. S. Thys et al. employed a similar technique to create an adversarial patch to evade object detection systems [34]. They optimized a total loss function to generate the patch.

![Face Detected](image1) ![Nothing Detected](image2)

Figure 3: Example of adversarial attacks on facial/object detection systems using adversarial glasses.

4. GANs for Adversarial Examples in Malware: W. Hu and Y. Tan utilized GANs to generate adversarial domain names to evade detection by domain generation algorithms [35]. J. Spooren et al. proposed MalGan, a GAN-based algorithm, to generate malware examples and evade black-box detection. They used a substitute detector to simulate the real detector and leveraged the transferability of adversarial examples to attack the real detector. MalGan was trained using reinforcement learning, with the evasion rate considered as a reward [36].

These examples highlight the diverse applications of adversarial attacks and the use of different techniques, ranging from gradient-based methods to GANs, in generating adversarial examples to deceive various systems and models.

As adversarial attacks have evolved, defense mechanisms have also advanced, employing both reactive and proactive strategies. Table 2 provides notable examples:

<table>
<thead>
<tr>
<th>Defense Type</th>
<th>Study</th>
<th>Application</th>
<th>Attack Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reactive</td>
<td>[37]</td>
<td>Adversarial Detecting</td>
<td>FGSM</td>
</tr>
<tr>
<td></td>
<td>[38]</td>
<td>Input Reconstruction</td>
<td>Feature Adversary</td>
</tr>
<tr>
<td></td>
<td>[39]</td>
<td>Network Verification</td>
<td>Feature Adversary</td>
</tr>
<tr>
<td>Proactive</td>
<td>[40]</td>
<td>Network Distillation</td>
<td>Data Poisoning</td>
</tr>
<tr>
<td></td>
<td>[23, 30]</td>
<td>Adversarial Retraining</td>
<td>GAN</td>
</tr>
<tr>
<td></td>
<td>[41]</td>
<td>Classifier Robustifying</td>
<td>GAN</td>
</tr>
</tbody>
</table>

Table 2: Summary of Countermeasures for Adversarial Examples. The defense type categorizes the defense methods as reactive or proactive, the study represents a real-world study, the application shows the type of defense method, and the attack method shows which methods this application defends against.
1. Adversarial Detecting: Deep neural network-based binary classifiers have been trained to detect adversarial examples. These classifiers classify input data as either legitimate or adversarial. Metzen et al. introduced an auxiliary network as a detector for adversarial examples, achieving successful detection of various attack methods such as FGSM [37].

2. Input Reconstruction: Adversarial examples can be transformed into clean data through reconstruction, rendering them ineffective against deep learning models. Gu and Rigazio proposed a deep autoencoder, a variant of the autoencoder network with a penalty, to enhance the robustness of neural networks [38].

3. Network Verification: Network verification verifies the properties of deep neural networks to defend against adversarial examples. It checks whether an input violates or satisfies certain properties, serving as a promising solution for detecting new and unseen attacks [39].

4. Network Distillation: Originally designed to reduce the size of deep neural networks, network distillation has been repurposed for defending against adversarial examples [40]. It involves training a second neural network using the probability of classes produced by the first network. This approach was shown to enhance the robustness of deep neural networks against adversarial attacks.

5. Adversarial (Re)training: Adversarial retraining involves training neural networks with adversarial examples to improve their robustness. Goodfellow et al. and Huang et al. demonstrated the effectiveness of including adversarial examples in the training stage [23, 30]. Adversarial training methods have shown increased robustness against one-step attacks but may be less effective against iterative attacks. Ensembling adversarial training methods have also been proposed to train models with adversarial examples generated from multiple sources.

6. Classifier Robustifying: Robust architectures of deep neural networks have been designed to prevent adversarial examples. Bayesian classifiers have been leveraged to build more robust networks by considering the uncertainty introduced by adversarial examples. Separating classes into sub-classes and ensembling results from all sub-classes through voting have also been explored to prevent misclassification of adversarial examples [41].

These defense strategies demonstrate the ongoing efforts to mitigate the impact of adversarial attacks on deep learning systems by developing resilient architectures, leveraging input manipulation and incorporating adversarial examples in the training process. In this research, an ensembling of these defense mechanisms is used to create a more robust way to improve detection systems.
2.2 Generative Modeling

Adversarial learning plays a pivotal role in some prominent generative models, spanning tasks such as image, video, text and speech synthesis across various domains. Generative modeling coupled with adversarial learning has experienced a remarkable surge in popularity due to its ability to learn and generate data with minimal supervision. The core concept behind generative modeling is to train a model that produces samples following the same distribution as the training data. The most prominent generative modeling technique is a generative adversarial network, otherwise known as a GAN. A GAN is a type of machine learning model that utilizes adversarial learning to generate fake yet realistic looking data [42]. A GAN’s architecture consists of two main components: a generator and a discriminator, as shown in Fig 4.

Figure 4: Overview of a vanilla GAN’s two-network architecture featuring a generator and discriminator.

The goal of a GAN is to generate new data samples that are similar to a given training dataset with respect to a minmax loss function, such as binary cross entropy (BCE), shown in Eq. 1:

$$\min_{G_{gan}} \max_{D_{gan}} l[\log D_{gan}(x)] + (1 - l)[\log(1 - D_{gan}(G_{gan}(z)))]$$

(1)

Binary cross entropy utilizes the truth labels from the images, $l$ of $x_{gan}$, the fake images from the generator, $G_{gan}(z)$, and the classification output of the detector, $D_{gan}$, to improve the generative network. Using BCE, the discriminator’s goal is to correctly identify which samples are real, while the generator’s goal is to fool the detector by making its images seem real. Each system utilizes Eq. 1 to achieve their goal. For the detector, the left side of the equation is used to classify an image as real, while the right side of the equation is used to classify an image as fake. Therefore, the
discriminator \( (D_{gan}) \) works to push predictions of real images towards one and fake images towards zero, while the generator \( (G_{gan}) \) works to push the prediction of fake data towards 1, fooling the detector. The generator creates new data by mapping the latent space into realistic looking images. These generated samples are then presented to the discriminator along with real data samples from the training dataset. The loss from the minimax function is back propagated to the generator and discriminator for learning.

Initially, both the generator and discriminator are relatively poor at their tasks. However, as the training progresses, they continuously improve by learning from the loss function. The generator aims to generate samples that are so realistic that the discriminator cannot distinguish them from real data. Meanwhile, the discriminator becomes more adept at identifying the generated samples.

This training process is often likened to a competitive game between the generator and the discriminator. The generator acts as the adversary, striving to produce more convincing samples, while the discriminator constantly hones its ability to differentiate between real and generated samples.

A GAN continues to train until a balance is achieved, where the generator can create highly realistic samples that fool the discriminator. The generated samples can be used for various applications like generating new images, creating synthetic data for training other models, or even producing artistic and creative outputs.

It’s important to note that GANs can be challenging to train and require careful tuning and optimization [43, 44]. If training becomes unstable, GANs can experience mode collapse, meaning the generator cannot produce new, unique images. However, when successful, they can produce impressive results and have been used in various fields, including computer vision, natural language processing and art generation. GANs can also be used to create adversarial examples for adversarial learning.

### 2.3 Deepfake Generators

Deepfake generation methods typically rely on deep learning algorithms, with (GANs) being the most popular approach due to their ability to generate high-quality and diverse images. This research employs a GAN as a key component in the continuous learning pipeline to generate deepfakes. However, the selection of the appropriate GAN necessitates a comprehensive examination of existing models.
Fig. 5 shows the taxonomy of GANS. These GANs are grouped into three sections. The architecture section groups GANs with notable architectures that helped establish a “backbone” for GANs. GAN architectures are generally difficult to adopt. The next section, conditioning goal, has to do with GAN models that are able to synthesize an intended style or image through training. Finally, the adversarial loss section has to do with models that have a vanishing gradient problem. We will describe a few relevant models:

1. DCGAN: The first prominent results in GAN development came from adopting a backbone architecture with DCGAN - as opposed to a multi-layer perceptron. Although simple, DCGAN is capable at producing recognizable images and stabilizes the dynamic between the generator and discriminator [45].

2. StyleGAN: As deep learning accelerated from DCGAN, StyleGAN became a novel architecture,
specializing in generating high quality images with low inter-class variation and more inductive bias for stabilization within the model. StylGAN is now one of the most state-of-the-art deepfake generators [4].

3. ACGAN: cGANs stand for conditional image GANs. ACGAN adopts an auxiliary classifier on top of the discriminator to try and minimize loss and the objective function [46].

4. Vanilla GAN: The Vanilla GAN is the most basic GAN architecture and suffers from an unstable characteristic - the vanishing gradient problem. This problem, often called mode collapse, happens when the GAN tries to minimize divergence between the real and fake distributions, therefore, it cannot provide helpful gradient signals to the discriminator and the network cannot learn. This discovery was integral to the development of more advanced GANS [47, 48].

In conclusion, we have explored several notable deepfake generators, each contributing to the advancement of deepfake synthesis. It is important to note the complexities of GANs when trying to work with them. Despite their incredible performance, it is difficult to create a stable GAN from scratch. That is why many people utilize state-of-the-art methods, such as StyleGAN, due to its state-of-the-art architecture, high-quality visual output, fine control over synthesis, improved diversity, robustness to image artifacts, and its strong presence within the research community. Its capabilities and advancements have propelled the field of deepfake generation, paving the way for exciting applications and driving the progress of the technology.

2.4 Deepfake Detectors

Deepfake detection methods have evolved with the ever-changing generation methods. According to the literature, traditional detection methods for identifying deepfakes primarily relied on various media manipulation techniques. These techniques aimed to analyze and detect anomalies or inconsistencies in the visual content, audio signals, or metadata associated with the media. While these traditional methods laid the foundation for detecting manipulations, they faced limitations in terms of accuracy and robustness, especially with the emergence of sophisticated deepfake technologies [49].

Media manipulation techniques encompass a range of approaches including: forensic analysis, statistical modeling and rule-based algorithms. Forensic analysis involves examining the media at a pixel level, searching for artifacts or traces left behind during the manipulation process [50]. Statistical modeling techniques employ statistical features or models to identify statistical deviations
from genuine media. Rule-based algorithms, on the other hand, rely on predefined rules or heuristics to flag suspicious patterns or characteristics in the media [49].

While these traditional methods contribute valuable insights into identifying manipulated media, they often struggle to keep pace with the rapid evolution and sophistication of deepfake techniques. Deepfakes leverage deep learning algorithms, particularly GANs, to generate highly realistic and convincing synthetic media that can deceive traditional detection methods.

Similar to advancements in generative networks, with the advent of deep learning, the field of deepfake detection has also witnessed significant advancements. Deep learning detection methods harness the power of convolutional neural networks (CNNs) and other deep learning architectures to automatically learn and extract intricate patterns and features from the media [51]. By training on large datasets of both genuine and manipulated media, these models can discern subtle inconsistencies or artifacts introduced by deepfake techniques.

![Figure 6: Vanilla CNN deepfake detection system that consists of an input layer, convolution layer and pooling layer for feature extraction, and a fully-connected MLP that works as a classifier to determine whether an image is real or fake.](image)

Shown in Fig. 6, general CNN deepfake detection architectures use convolutional layers, pooling layers and fully connected layers to learn hierarchical representations of the input data - starting from low-level features (e.g., edges, textures) and progressively learning higher-level features (e.g., facial features, object shapes). After training, you can input an image to be classified. This image passes through convolutional and pooling layers for feature extraction. Then, this data is input to a fully connected network that is used to classify if the image is real (one) or fake (zero). By training on a large dataset containing both genuine and manipulated media, CNNs can learn to identify subtle visual artifacts, inconsistencies, or statistical anomalies introduced by deepfake techniques [51].

Overall, deep learning-based detection methods have demonstrated improved accuracy and
robustness in identifying deepfakes across various modalities, including images, videos and audio. Additionally, deep learning models can adapt and evolve as new deepfake techniques emerge, making them more resilient to evolving threats.

Figure 7: Taxonomy of Deepfake Detection Methods using CNNs.

1. MesoNet is a CNN-based deepfake detection method that focuses on micro-expression analysis to identify manipulated facial videos. It operates on the premise that deepfake videos often lack subtle facial movements present in genuine videos. MesoNet extracts features from multiple scales and uses a deep neural network for classification. Despite its high accuracy in detecting subtle movements and being robust against deepfake techniques, MesoNet’s performance may degrade when videos involved more complex manipulation such as face swap or lip syncing.[52, 53]

2. CNNDetection is another CNN-based approach designed for deepfake detection. It employs a
deep neural network architecture to analyze facial features and learn discriminative patterns indicative of deepfake manipulations. CNNDetection utilizes a large dataset of both real and manipulated videos for training. CNNDetection has a high accuracy in detecting various types of deepfakes and has the ability to learn complex features, making it effective in capturing subtle artifacts. However, its performance can be easily impacted by adversarial attacks. [54]

3. FFD (Feature-Fusion-based Detection) combines features extracted from multiple modalities, such as color, texture, and motion, using a fusion mechanism. By integrating information from different modalities, FFD aims to enhance the detection performance and robustness against various deepfake manipulation techniques. It leverages multi-model features which enhances its accuracy and resilience. However, it relies heavily on the selection of modalities which can require extensive optimization.[55, 56]

4. FALDetector focuses on inconsistencies in the Fourier phase spectrum of manipulated images, which are often altered during deepfake generation. It provides a unique approach for detection by incorporating both spatial and frequency manipulations. However, it is also vulnerable to adversarial attacks targeted at the Fourier phase spectrum.[57, 58]

It is worth noting that traditional detection methods, although surpassed in accuracy by deep learning methods, still play a role in the deepfake detection landscape. They can complement deep learning approaches by providing additional insights, performing post-processing tasks, or contributing to ensemble-based detection systems. Furthermore, traditional methods continue to be valuable for analyzing non-deepfake manipulations and serve as a foundation for understanding the broader context of media forensics.

In conclusion, while traditional detection methods encompassed various media manipulation techniques and laid the groundwork for deepfake detection, the advancements in deep learning have significantly enhanced the accuracy and effectiveness of detecting deepfake manipulations. Deep learning-based approaches have shown remarkable potential in addressing the challenges posed by increasingly sophisticated deepfake technologies and are paving the way for more robust and reliable detection systems.
Chapter 3

METHODOLOGY

3.1 Overview

Current research suggests that generative networks are constantly evolving, making it crucial for detection methods to keep pace with these advancements. However, detection methods often lag behind in development due to a reactive approach to improvement, reiterating the point that this phenomenon can be likened to an arms race, wherein as generative methods progress, detection methods must also enhance their capabilities. This research seeks to explore whether ADVANCE, a continuous and collaborative learning pipeline, can simulate an arms race between a deepfake generator and deepfake detector to contribute to determining which system will converge first.

To address this question, this methodology revolves around employing adversarial learning to mutually enhance a generator and a detection system. Specifically, this research utilizes a state-of-the-art deepfake image generator and a cutting-edge detection algorithm, complemented by an intelligent reasoning block that leverages the outputs of these networks to facilitate their performance. I have chosen to focus on the problem of deepfake image generation as it is a relevant and evolving field where the detection methods struggle to keep up with the advancements in generative techniques. Additionally, the visual nature of deepfake images offers valuable insights into the learning process of the entire pipeline.

By establishing a continuous and collaborative learning environment, this research aims to investigate if a pipeline exists to simulate the endpoint to the arms race between generative networks and detection methods. This environment harnesses the power of adversarial learning, where the generator and detection system engage in a dynamic interplay, constantly challenging and improving each other’s capabilities. This iterative process seeks uncover whether these networks can propel each other to new heights of performance or if they eventually encounter a limit beyond which further
improvement becomes unattainable.

This section aims to define the methods used within ADVANCE as well as the experimental settings and results. The aim of this research contributes to understanding the evolving dynamics between generative networks and detection methods, shedding light on the potential outcomes of the ongoing arms race in the field of artificial intelligence.

3.2 Methods

To facilitate this experiment, I devised a method for constructing ADVANCE to enabling an arms race between a deepfake generator and detector.

![Diagram of ADVANCE](image)

**Figure 8:** High-level overview of my system, ADVANCE. ADVANCE uses an iterative approach to simulate an arms race between two systems.

Fig. 8 presents a high-level overview of the continuous and collaborative adversarial pipeline employed in the experiments. Note that this figure resembles a GAN system, however, it is not a GAN. The Generator and the Detector are two separate systems with different purposes.

The pipeline begins with training the generator, where the generator’s output is subsequently fed into the detector. The detector’s outputs undergo intelligent reasoning to filter the data and determine whether the detector or generator should be retrained. The retraining process will utilize the images that were filtered, otherwise known as the “optimal images” that managed to deceive the detector. This process iterates until the threshold is met based on the results of the detector and generator.

This experiment has three main components: a Generator System we have labeled $G$, a Detection System labeled $D$, and an Intelligent Reasoning Algorithm.

Fig. 9 shows the low-level details of the pipeline. The input to the system is a random latent vector, $z$. This vector is initially input into the generator system, $G$. $G$ produces fake images, $G(z)$,
by mapping $z$ to $G(z)$. $G(z)$ and real images, $x$, are used as input into the detection system, $D$, which classifies the images as real or fake. $D$ outputs various metrics that are used to determine what happens next in the cycle. $D(x)$ and $D(G(z))$ represent the detector’s accuracy on classifying the real images and fake images respectively. $D(x, G(z))$ represents the systems overall accuracy while $C$ represents a set of confidence scores, $c$, associated with each fake image. The Intelligent Reasoning Algorithm undergoes an Intelligent Selection process and an Intelligent Filtering process. Intelligent Selection uses $D(x, G(z))$. If $D(x, G(z))$ is less than or equal to 60%, then the pipeline will retrain $D$, or else it will retrain $G$. The Intelligent Filtering uses $C$ to choose which fake images fool the detector and filters them with the filtering function $F(G(z), C)$. These images are used to retrain the chosen system.

The deepfake generator utilized in this research, labeled as $G$, is StyleGAN3, an advanced generative adversarial network (GAN) architecture renowned for its ability to produce highly realistic and diverse images [59, 4]. StyleGAN3 builds upon the success of its predecessors, incorporating several features and improvements that enhance the quality and control of generated deepfake images.

Shown in Fig. 10, at the core of StyleGAN3 is a two-network architecture consisting of a generator and a discriminator. The generator network operates by mapping latent variables, often represented as random vectors or codes, into the image space. It leverages a mapping network and a synthesis network. The mapping network takes the random latent vectors, $z$, and maps them to disentangled latent vector weights, $w$, that control various aspects of image generation, such as pose, expression and style. The mapping network consists of several fully connected, FC, layers that process the input vectors and transform them into meaningful latent representations. Each fully connected layer performs a linear transformation on the input data by multiplying it with weight matrix $w$ and
Figure 10: StyleGAN3’s architecture. Left: full architecture including the input latent vector \( z \), the mapping network of fully-connected layers, the associated vector weights \( w \), the synthesis network which maps \( z \) to the fake images as output; middle: an expansion of the synthesis network showing how the style layers and ToRGB layer maps the input to the output of fake images; right: the discriminator architecture within the GAN consisting of convolutional layers featuring skip connections and down sampling.

adding a bias term \( b \). The output of each neuron in the fully connected layer is computed using a weighted sum of the inputs followed by an activation function. An affine transformation, \( A \), and a fourier feature mapping is applied to the weight matrix to extract important features which will be used within the synthesis network.

The synthesis network consists of multiple style layers, \( L \), that play a crucial role in controlling the style and appearance of the generated images. Each style layer corresponds to a particular resolution level in the image synthesis process. The main purpose of these style layers is to modify the input noise vector to produce different styles and variations in the generated images.

The style layers in StyleGAN3 operate by manipulating the intermediate feature maps in the synthesis network. These feature maps capture the complex hierarchical representations of the images at different resolutions. By applying modulation and demodulation to the feature maps, the style layers can modulate the mean and standard deviation of the feature maps, effectively controlling the appearance and style of the generated images. The style vector, often referred to as the latent code, determines the values for the adaptive parameters in each style layer, enabling precise control over various attributes such as colors, textures and shapes.

The ToRGB layer is another key component of StyleGAN3. This layer is responsible for transforming the intermediate feature maps into the final RGB image output. Each resolution level in the synthesis network has a corresponding ToRGB layer that applies a linear transformation to the feature maps to generate the RGB values for the corresponding resolution. The ToRGB layers allow
the model to generate high-quality images by combining the modulated feature maps from the style layers with the appropriate color information.

Finally, these high-quality images are put into the GANs discriminator network. The discriminator in StyleGAN3 consists of multiple residual blocks, each containing convolutional layers and skip connections. The skip connections are responsible for the residual connections in the network, allowing the model to learn residual mappings rather than direct mappings. This approach helps address the vanishing gradient problem and enables the discriminator to effectively capture fine-grained details and features in the input images.

One of the key strengths of StyleGAN3 lies in its unparalleled control over various image attributes. Through the introduction of style modulation, the generator network can manipulate specific aspects of the generated images independently, including facial features, pose, expression and level of detail. This control enables researchers to investigate the impact of different attributes on the detection performance and explore the limits of deepfake detection methods. Moreover, StyleGAN3 incorporates a style-mixing technique that allows for the blending of multiple latent codes during the generation process. This results in the creation of hybrid images that possess characteristics from different latent spaces, producing a diverse range of deepfakes that exhibit a remarkable level of realism and variability.

To train the StyleGAN3 model, a large-scale dataset of real images is utilized. The generator network is trained to learn the underlying patterns and distributions present in the dataset, while the discriminator network is simultaneously trained to distinguish between real and generated images. This adversarial training process fosters the improvement of both networks, leading to the production of increasingly convincing deepfake images.

Overall, StyleGAN3 represents a state-of-the-art deepfake generation method that offers advanced control, diversity and realism. Its ability to create highly detailed and visually coherent deepfake images sets the stage for an effective and challenging arms race between the generator and detection systems, facilitating the investigation of detection methods in the face of rapidly advancing generative networks.

The detection system utilized in this research is CNNDetection by Adobe, a powerful and cutting-edge deep learning model specifically designed for the identification and classification of deepfake images [54, 60]. Developed by Peter Wang and his team, CNNDetection leverages state-of-the-art convolutional neural network (CNN) architectures and techniques to accurately differentiate between real and generated images.

CNNDetection is built upon a deep neural network, ResNet [61], that is trained on a diverse and
extensive dataset comprising both deepfake and authentic images. Shown in Fig. 11, the network architecture incorporates multiple convolutional layers, pooling layers and fully connected layers, enabling it to learn discriminative features and capture complex patterns indicative of deepfakes.

Figure 11: ResNet50 architecture used in CNNDetection to classify images as real or fake.

One notable feature of CNNDetection is its ability to extract high-level semantic information from images, allowing for robust and accurate classification. By leveraging the hierarchical nature of CNNs, the model learns to automatically identify and extract relevant features at various levels of abstraction, enabling it to distinguish subtle artifacts and inconsistencies present in deepfake images.

To train CNNDetection, a large labeled dataset is utilized, consisting of deepfake images generated by state-of-the-art methods, as well as authentic images sourced from various reliable and diverse sources. The model is trained using a combination of supervised learning techniques and advanced optimization algorithms, such as stochastic gradient descent, to iteratively adjust the network’s parameters and optimize its performance. During the inference phase, CNNDetection takes a candidate image as input and passes it through the trained network. The model then computes a confidence score indicating the likelihood that the image is a deepfake. This score serves as a reliable measure for decision-making and subsequent steps within the intelligent reasoning block of the adversarial learning pipeline.

CNNDetection by Adobe has gained recognition in the research community for its high accuracy and robustness in detecting deepfake images. Its ability to effectively identify manipulated content contributes to the ongoing efforts to combat the spread of misinformation and protect against the potential misuse of deepfake technology.

CNNDetection by Adobe has gained recognition in the research community for its high accuracy and robustness in detecting deepfake images. Its ability to effectively identify manipulated content contributes to the ongoing efforts to combat the spread of misinformation and protect against the potential misuse of deepfake technology.

By integrating CNNDetection into the adversarial learning pipeline, this research aims to explore its performance in the face of evolving deepfake generation methods. Additionally, the experiments seek to analyze the impact of the generator’s advancements on the detection system’s accuracy and determine if the detection system can iteratively improve and adapt to the changing landscape of deepfake technology.
deepfake techniques.

Overall, CNNDetection by Adobe represents a state-of-the-art deep learning-based detection system that plays a critical role in the arms race against deepfake generation methods. Its ability to accurately discern between real and generated images provides a strong foundation for investigating the limits of deepfake detection and further enhancing the security and trustworthiness of digital media.

The intelligent reasoning algorithm employed in this research is a custom-designed approach tailored to suit the specific output and input requirements of the generator and detector. This algorithm plays a crucial role in guiding the decision-making process within the iterative pipeline, enabling effective training of the networks.

Algorithm 1: Intelligent Reasoning Algorithm

<table>
<thead>
<tr>
<th>Function F(G(z), C):</th>
</tr>
</thead>
<tbody>
<tr>
<td>images ← null</td>
</tr>
<tr>
<td>for c in C do</td>
</tr>
<tr>
<td>if c ≤ 60% then</td>
</tr>
<tr>
<td>images ← G(z);</td>
</tr>
<tr>
<td>else</td>
</tr>
<tr>
<td>continue to the next confidence score c</td>
</tr>
<tr>
<td>return images;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function Main:</th>
</tr>
</thead>
<tbody>
<tr>
<td>optimalImages ← F(G(z), C)</td>
</tr>
<tr>
<td>if D(x,G(z)) ≤ 65% then</td>
</tr>
<tr>
<td>train ← D</td>
</tr>
<tr>
<td>else</td>
</tr>
<tr>
<td>train ← G</td>
</tr>
<tr>
<td>return optimalImages, train</td>
</tr>
</tbody>
</table>

Alg. 1 utilizes the output of the detection system to make its decision on what system to train and what images to use in training. The detection system provides two key metrics as input to this algorithm: the overall accuracy of real and fake image detection, $D(x,G(z))$ and a set of confidence scores assigned to each individual image, $C$. These metrics serve as valuable indicators of the detection system’s performance as well as the images generated by the generation method.

The overall accuracy reflects the detector’s ability to correctly classify both real and fake images. By monitoring this metric, the system is able to determine whether the detection system is performing at an acceptable level or if it requires retraining. For instance, if the overall accuracy drops below a predetermined threshold it suggests that the detector’s performance has deteriorated, indicating the
need for retraining. On the other hand, if the overall accuracy remains consistently high, it may be an opportune time to focus on improving the generator.

Complementing the overall accuracy, the confidence score assigned to each image provides a deeper understanding of the detector’s level of certainty in its classification. This score represents the detector’s confidence in labeling an image as either real or fake, with higher scores indicating a stronger belief that the image is fake and lower scores suggesting a higher likelihood of perceiving the image as real. The utilization of the confidence score in the intelligent reasoning algorithm enables researchers to identify images that pose challenges to the detector. By prioritizing images with lower confidence scores, researchers can ensure the generation of diverse and challenging samples that push the limits of the detection system. This approach fuels the continuous development and refinement of both the generator and detector, leading to the creation of more realistic deepfake images and more robust detection methods.

First, optimalImages used for re-training the networks are determined by the filtering function $F$. $F$ takes in the fake images $G(z)$ and the set of confidence scores with their corresponding fake image using $C$. Iterating through $C$, $F$ returns the images that have a confidence score of being synthetic of 60% or less. These images are the fake images that the detector classified as real.

The Main function returns the optimalImages from $F$ that will be used for training as well as the system that is to be trained next, labeled as train. If the detector’s accuracy $D(x, G(z))$ is less than or equal to 65%, then train = $D$ or else train = $G$. When the if statement is true, then the detector is no longer able to tell the difference between real or fake images and $D$ must be trained. Otherwise, the generator $G$ will be trained.

The intelligent reasoning algorithm, incorporating both the overall accuracy and the confidence scores, drives the iterative cycle of improvement between the generator and detector. While the overall accuracy guides the decision on whether to retrain the detector or generator, the confidence score allows for a deeper analysis of misclassifications and the selection of optimal training data. This combined approach ensures a comprehensive understanding of the detection system’s performance and facilitates targeted enhancements to create more robust and effective deepfake detection systems.

While the generation method we utilized, StyleGAN3, has shown effectiveness in producing realistic deepfake images, it is important to acknowledge certain limitations that we encountered. One significant limitation is the reliance on a substantial amount of diverse and representative training data. Insufficient or biased training data could impact the realism and diversity of the generated deepfake images, potentially affecting the generalization of our findings. Additionally, the generation process may inherit biases present in the training data, introducing potential content
and characteristic biases in the synthetic images produced. Another consideration is the significant computational resources required for training the generator, which may pose limitations in terms of time and availability of computing power.

In addition to the generation method showing effectiveness, the detection system we employed, CNNDetection, has demonstrated effectiveness in distinguishing between real and fake images. However, it is also crucial to acknowledge limitations associated with its use. One potential limitation lies in the interpretation and utilization of confidence scores. While lower confidence scores indicate images that challenge the detector, careful analysis and interpretation of these scores are necessary, as they can be influenced by factors such as image quality and dataset biases. It is essential to consider these nuances in order to avoid potential misinterpretation of the detection results. Furthermore, the effectiveness of the detection method may vary depending on the sophistication of the pre-trained network. Choosing this network is integral to the overall accuracy of the system.

Finally, our custom-designed intelligent reasoning algorithm has demonstrated its ability to facilitate the iterative improvement of the generator and detector. However, one consideration is the subjectivity introduced by the chosen thresholds and criteria used to determine network retraining. The decision-making process relies on these predefined thresholds, which may introduce biases and influence the results. It is crucial to carefully select and justify these thresholds based upon your problem set to mitigate any potential biases and maintain the validity of our findings.

In summary, the methods chosen encompass the utilization of the StyleGAN3 generation method, the CNNDetection system for deepfake detection and an intelligent reasoning algorithm to guide the iterative cycle between the generator and detector. While acknowledging the limitations associated with each component, these methods align with the research objective and contribute to addressing the research question at hand. By transparently addressing potential biases, constraints and challenges, we enhance the credibility and robustness of our findings, ultimately advancing the understanding and development of deepfake generation and detection techniques.

### 3.3 Experimental Settings

The dataset utilized in this experiment is the FFHQ dataset, represented as $x$ within Fig. 9. This dataset consists of 256x256 images of faces to train StyleGAN3. The FFHQ dataset is a widely recognized and commonly used dataset in the field of computer vision and generative modeling. It was curated by Nvidia and consists of high-quality, human face images collected from various sources [62].
The FFHQ dataset shown in Fig. 12 is renowned for its large size, containing over 70,000 images, making it suitable for training deep learning models that require a substantial amount of data. The dataset encompasses a diverse range of facial attributes, including different genders, ages, ethnicities and expressions, providing a broad representation of human faces. This diversity ensures that our models have exposure to various facial characteristics and can generate and detect deepfake images that capture the nuances of different individuals.

![Figure 12: FFHQ Dataset utilized within training and representing the real image dataset.](image)

During the initial training round of StyleGAN3, the FFHQ dataset consisting of real images was exclusively employed to establish a set of authentic and visually realistic images. The images generated from training on this dataset serve as the input for the detector in our research. When selecting a dataset, it is crucial to consider factors such as dataset size, class distribution and preprocessing requirements. In the case of FFHQ, which primarily comprises facial images, there was no need for additional preprocessing or class labeling due to the nature of the dataset.

The choice of 256x256 sizing for the images was deliberate, as it allowed for efficient training and seamless integration with the detector after the generation process. This sizing decision balanced the computational resources required for training with the need to maintain image quality and compatibility. It facilitated a smooth workflow between the generator and the detector, ensuring that the generated images could be readily processed and analyzed.

It is worth noting that StyleGAN3 offers a range of pre-trained networks trained on different datasets. While FFHQ was selected as the dataset for our experiment, alternative datasets and pre-trained networks could have been utilized for this purpose.
The CNNDetection system is a universal deepfake detection method. Built upon the widely-used ResNet50 architecture, this convolutional neural network (CNN) has exhibited remarkable performance in discerning between genuine and manipulated images. ResNet50 is renowned for its exceptional capability to extract intricate visual features, making it a prominent choice in various computer vision applications. The CNNDetection system leverages this robust architecture along with advanced techniques to thoroughly analyze image characteristics and identify potential indicators of deepfake manipulation.

To train the CNNDetection system, Wang and his team curated a meticulously designed dataset comprising a diverse array of real and fake images. The fake images used in the training set were generated using a specific type of generative adversarial network (GAN) called ProGAN [63, 64]. Notably, the research findings demonstrate the system’s ability to generalize effectively to unseen architectures, datasets and training methods, including the popular StyleGAN3, through careful preprocessing, post-processing and data augmentation techniques. This suggests that the CNNDetection system exhibits a high level of adaptability and universality in detecting deepfakes across different contexts and variations in generative models.

By utilizing the CNNDetection system in our research, we benefited from its extensive training and the expertise embedded within its design. This allowed us to focus on investigating the collaborative learning between the generator and detector without the need to allocate significant time and resources to train a detection model from scratch. The robustness and adaptability of the CNNDetection system were instrumental in aligning our chosen methods with our research objectives, providing a solid foundation for addressing our research question regarding the iterative improvement between generative networks and detection methods.

In our research, a comprehensive set of evaluation metrics were employed to assess the progress and performance of both the generator and detector throughout the iterative process. These metrics allow us to compare the results at each iteration and gain insights into the improvement and effectiveness of the respective models.

Evaluating the performance of a generator can be challenging due to its complex nature, especially when assessing the realism of fake images. Traditionally, fidelity, diversity and inherent qualities of the generated dataset are crucial aspects. However, quantifying these qualities requires reliable metrics.

In our evaluation of the generator’s performance, we employed the Fréchet Inception Distance (FID) score, which is a widely-used metric in generative modeling [65, 66, 67]. The FID score captures both the quality and diversity of generated images by quantifying the dissimilarity between the
distributions of real and generated images. The FID score is calculated by taking the difference between the mean and covariance of the real and fake image distributions. A lower FID score indicates a closer resemblance, on a feature level, between the generated and real images, signifying higher levels of realism and quality. Equation 2 shows how the FID score is calculated using $\mu(z)$ as the mean of the distribution of real images and the mean of the distribution of fake images as $\mu(G(z))$. The difference of covariance of the distributions is obtained by taking the trace of the difference of the covariances. Labeled as $\Sigma(z)$ symbolizing the covariance of the distribution of real images and $\Sigma(G(z))$ representing the covariance of fake images.

$$FID = |\mu_x - \mu_{G(z)}| + \text{tr}\left(\Sigma_x + \Sigma_{G(z)} - 2\left(\Sigma_x \Sigma_{G(z)}\right)^{\frac{1}{2}}\right)$$ (2)

By monitoring the variations in the FID score at each iteration, we were able to evaluate the generator’s continuous improvement in generating realistic images. This evaluation framework offered valuable insights into the generator’s progress and its capability to produce high-quality outputs.

Additionally, the FID score is complemented by the use of visual analysis in evaluating the generator’s output. Human visual perception is invaluable in evaluating the overall quality, authenticity and coherence of the generated images. Visual analysis shows a qualitative assessment of the generated images and ensures that they maintained a high level of realism throughout the iterative process [67]. By combining quantitative metrics, such as the FID score, with subjective visual analysis, a more comprehensive understanding of the generator’s performance was achieved, along with its capacity to consistently produce compelling deepfake images. However, it is important to acknowledge that the objective of visual analysis is not necessarily to generate more realistic-looking images, but rather to maintain realism within the generated data.

Regarding the detector, the primary evaluation metric used was the overall accuracy, shown in Eq. 3. Using the true and false positives (TP, FP) and negatives (TN, FN), accuracy measures the detector’s ability to correctly classify an image as real or fake. A higher overall accuracy indicates a more reliable and effective detection performance. By monitoring the changes in the overall accuracy at each training round, trends and patterns in the detector’s classification abilities and its capacity to distinguish between real and fake images were discerned.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$ (3)

By utilizing the FID score and visual analysis for the generator, along with the overall accuracy for the detector, valuable insights were obtained regarding the interplay between the improving
quality of fake images and the detection performance. The evaluation metrics facilitated tracking and analysis of trends, ensuring iterative progress for both the generator and detector in terms of performance. However, it is important to acknowledge that these evaluation metrics provide indications of performance but may not capture all aspects of the models’ capabilities or potential limitations.

Now that the key components of the system have been established, we can expand upon the iterative process of the learning pipeline. ADVANCE consists of a series of rounds, each contributing to the generator and detector.

![Figure 13: Overview of the iterative process within ADVANCE alternating between training the generator (left) and the detector (right).](image)

Fig. 13 shows how each round focuses on training either the detector or the generator. The generator’s objective is to produce synthetic images that can successfully deceive the detector into classifying them as real. Each round, the generator is retrained on synthetic images output from the previous iterations. These images are the optimal images that fooled the detector. Therefore, creating a constant cycle of data that fools the detector. Through this iterative process, the generator becomes more adept at generating deepfake images that fool the detector.

Once the generator reaches a certain level of proficiency in fooling the detector, the next round involves retraining the detector. The detector is retrained on a combination of the previous iterations fake images as well as real images from the FFHQ dataset. Retraining the detector allows it to adapt and become more robust in distinguishing between real and fake images, even as the generator continues to improve its generation capabilities.

By following this iterative approach, we aim to achieve a continuous cycle of improvement in
both the generator and detector in order to determine whether these systems follow the assumption that the generator always wins against the detector. Each round brings us closer to refining the deepfake generation process and enhancing the detection capabilities, creating a more robust and effective system.

In summary, ADVANCE follows a cyclical process of iterations and rounds, where the generator and detector continually improve their respective capabilities. This iterative approach, depicted in Fig. 13, forms the foundation of my research and aligns with the objectives of addressing the research question. It allows for continuous refinement and development of deepfake generation and detection methods, bringing us closer to understanding the dynamics and potential thresholds in this arms race.

To replicate this experiment, we provide the system parameters used for training in Table 3 and 4. Training StyleGAN3 involves a significant number of parameters, considering its versatility and extensive processing requirements. Our parameter selection was influenced by the FFHQ dataset, particularly due to the choice of smaller 256x256 images. For this reason, we opted for the ”stylegan3-r” model configuration, which was recommended for our dataset. This configuration ensures rotational equivariance in the generated images, enhancing the detector’s vulnerability.

The choice of the number of GPUs, batch size, gamma, batch GPU, mirror, Kimg and Cbase parameters were based on the available resources at the time of conducting the experiment as well as recommended configurations for good performance. We also considered the number of snapshots to facilitate network training monitoring, while the initial network used for further training was compatible with our model and dataset. These considerations were made to strike a balance between computational feasibility and achieving desirable outcomes.
### Generator Training: StyleGAN3 System Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
<th>Description</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Configuration</td>
<td>-cfg</td>
<td>Version of StyleGAN3 to use as model</td>
<td>stylegan3-r</td>
<td>x</td>
</tr>
<tr>
<td>Initial Dataset</td>
<td>-data</td>
<td>Dataset to train the network</td>
<td>FFHQ</td>
<td>70, 000 training images</td>
</tr>
<tr>
<td>GPUs</td>
<td>-gpus</td>
<td>Maximum GPU memory usage</td>
<td>2</td>
<td>x</td>
</tr>
<tr>
<td>Batch Size</td>
<td>-batch</td>
<td>Evenly divided batches amongst GPUs</td>
<td>32</td>
<td>x</td>
</tr>
<tr>
<td>Gamma</td>
<td>-gamma</td>
<td>R,1 regularization weight</td>
<td>2</td>
<td>x</td>
</tr>
<tr>
<td>Batch GPU</td>
<td>-batch-gpu</td>
<td>Uses multiple passes to avoid running out of memory</td>
<td>8</td>
<td>x</td>
</tr>
<tr>
<td>Mirror</td>
<td>-mirror</td>
<td>Augments the dataset with random x-flips</td>
<td>1</td>
<td>x</td>
</tr>
<tr>
<td>Kimg</td>
<td>-kimg</td>
<td>Controls the total number of training iterations</td>
<td>5000</td>
<td>x</td>
</tr>
<tr>
<td>Snapshots</td>
<td>-snap</td>
<td>Specifies the number of network snapshots during training</td>
<td>20</td>
<td>x</td>
</tr>
<tr>
<td>Cbase</td>
<td>-cbase</td>
<td>Speeds up the training by decreasing network capacity</td>
<td>16384</td>
<td>x</td>
</tr>
<tr>
<td>Initial Network</td>
<td>-resume</td>
<td>Specify network to retrain</td>
<td>stylegan3-r-ffhqu-256x256.pkl</td>
<td>This value changes every training round</td>
</tr>
</tbody>
</table>

Table 3: System parameters used for training the StyleGAN3 generator. Note these parameters were utilized per the original paper recommendations.

### Detector Training: CNNDetection Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blur Probability</td>
<td>-blur_prob</td>
<td>Sets the probability of applying blur to the images during training</td>
<td>0.5 and 0.1</td>
</tr>
<tr>
<td>Blur Range</td>
<td>-blur sig</td>
<td>Defines the range of blur magnitude values to be randomly applied to the images during training</td>
<td>0.0,0.3,0.6</td>
</tr>
<tr>
<td>Compression Probability</td>
<td>-jpg_prob</td>
<td>Sets the probability of applying JPEG compression to the images during training</td>
<td>0.5 and 0.1</td>
</tr>
<tr>
<td>Compression Method</td>
<td>-jpg_method</td>
<td>Specifies the methods to be used for JPEG compression</td>
<td>cv2,pil</td>
</tr>
<tr>
<td>Image Quality</td>
<td>-jpg_qual</td>
<td>Defines the range of JPEG quality values to be randomly applied to the images during training</td>
<td>30,100</td>
</tr>
</tbody>
</table>

Table 4: System Parameters used for training the CNNDetection detector. Note these parameters were utilized per the original paper recommendations.
Regarding the detector, CNNDetection is less computationally demanding compared to StyleGAN3. Therefore, the parameters used for the detector were relatively straightforward. To generate deepfake images, we employed both models provided by CNNDetection, with the only distinction being the blur applied to the images during pre-processing. The parameter choices for this experiment were based on the recommended parameters from the original experiment, ensuring the preservation of high-quality performance. It is important to keep in mind that these parameters can be slightly changed to speed up training time based upon your available resources.

3.4 Experimental Results and Analysis

The experiments conducted in this research aim to investigate the evolving dynamics between generative networks and detection methods in the field of deepfake image generation. The increasing sophistication of generative networks poses a significant challenge for detection methods, which often lag behind in development due to a reactive approach to improvement. To address this challenge, this study employs ADVANCE, a continuous and collaborative adversarial learning pipeline between a deepfake image generator and a detection system.

The experiments focus on the arms race between generative networks and detection methods, seeking to explore whether there exists a threshold beyond which these systems can no longer improve or if they have the potential for iterative enhancements and report our findings. The methodology revolves around the utilization of state-of-the-art deepfake image generation and detection techniques, complemented by an intelligent reasoning algorithm that facilitates the mutual improvement of the generator and detector.

We start by training the deepfake image generator using the StyleGAN3 architecture, which is renowned for its ability to produce highly realistic and diverse deepfake images. The generator leverages a two-network architecture and advanced control mechanisms to create visually appealing and coherent deepfakes. The detector used in this research is CNNDetection, a cutting-edge deep learning model specifically designed to identify and classify deepfake images. CNNDetection utilizes advanced convolutional neural network architectures to extract discriminative features and accurately differentiate between real and generated images.

We incorporate an intelligent reasoning algorithm that analyzes the outputs of the generator and detector, including overall accuracy and confidence scores, to guide the decision-making process within the iterative pipeline. This algorithm plays a crucial role in determining whether the generator or detector should be retrained based on the performance metrics and the challenges posed by the
generated deepfake images.

The experimental settings, listed in the Methodology section, encompass the use of the FFHQ dataset for training the generator, which provides a diverse range of high-quality human face images. The dataset ensures exposure to various facial attributes and allows for the generation of deepfakes that capture the nuances of different individuals. The CNNDetection system is trained on a curated dataset comprising real and fake images generated using ProGAN [64], demonstrating its ability to generalize effectively across different generative models.

Through these experiments, our aim is to gain valuable insights into the performance and limitations of generative networks and detection methods in the realm of deepfake image generation. The findings from this research will contribute significantly to our understanding of the ongoing arms race between these systems, providing valuable knowledge for iterative enhancements.

The primary goal of this research is to explore the possibility of establishing a stable pipeline that simulates which system, the generator or the detector, will converge first. This exploration is crucial for advancing the development of robust deepfake detection methods and, in turn, improving the security and trustworthiness of digital media.

This section begins by providing a comprehensive overview of the experiments conducted. It explains the utilization of evaluation metrics within the experiments and their role in benchmark comparisons. Furthermore, it presents a detailed step-by-step account of each experiment. Finally, a comparative analysis is provided to evaluate the performance of both the generator and detector throughout the experimental process.

As explained in the Methodology section, we employ evaluation metrics to compare the performance of the generator and detector in this experiment. For assessing the generator, we utilize the Frechet Inception Distance (FID) score along with visual analysis. The FID score provides a quantitative measure of image quality by considering the diversity of generated photos as well as the quality. Lower FID scores indicate better image quality. The visual analysis ensures that the generator produces images with a realistic appearance.

Regarding the detector, its performance is determined based on its accuracy in distinguishing between real and fake images. As the generator undergoes training, the detector’s accuracy should decrease, indicating that the generator is becoming more successful in fooling the detector. Once the detector’s accuracy reaches a threshold of no further improvement, we retrain it using new images, expecting its accuracy to improve. This iterative process helps us evaluate system performance.

To assess the proper functioning of the detector and generator, a benchmark analysis is conducted using the evaluation metrics. The generator is initially trained on real images from the FFHQ dataset
for a specified number of iterations, establishing a baseline for quality performance. During this training, Stylegan3 generates high-quality and realistic images. The deepfake images produced in this training phase serve as the baseline for fake image quality. Our objective is to achieve this baseline by training the generator to generate fake images that successfully evade detection by the CNNDetection model.

In Figure 14, we observe a comparison between real images, which have an FID score of 0, and benchmark deepfake images, which have an FID score of 4.5. The FID score serves as a metric to assess the quality of image generation, where a lower score indicates higher fidelity. Our objective is to generate images with a high level of fidelity, and thus, a decrease in the FID score signifies an improvement in the system’s performance.

The presented benchmark data serves as a reference for evaluating the image quality of the generated images in our experiment. It includes both real images and the deepfake images trained solely on real images. By comparing their respective FID scores, we can assess the fidelity of the generated images. The real images, being compared to themselves, achieve an FID score of 0. The deepfake images trained exclusively on real images exhibit remarkably low FID scores, which we will adopt as our benchmark for comparison.

Regarding the detector, we utilize CNNDetection’s initial accuracy on Stylegan3 as a benchmark. Initially, we evaluate the detector’s accuracy by testing it on a set of real and fake images generated
by Stylegan3. Upon evaluation, the detector achieves an accuracy of 92%. This initial accuracy serves as a baseline to compare whether the detector’s performance improves or deteriorates throughout the experiment. By monitoring changes in the detector’s accuracy, we can assess the effectiveness of both the generator and the detector in their respective tasks. The generator aims to produce realistic and high-quality images, while the detector aims to accurately identify fake images.

With a clear understanding of the evaluation criteria for our experiments, let us now delve into the step-by-step process of the conducted experiments.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Train StyleGAN3 on real images</td>
</tr>
<tr>
<td>2</td>
<td>Generate 10k images</td>
</tr>
<tr>
<td>3</td>
<td>Input 10k real and 10k fake images into detector</td>
</tr>
<tr>
<td>4</td>
<td>Evaluate detector results, accuracy &gt;65%</td>
</tr>
<tr>
<td>5</td>
<td>Filter images for next round</td>
</tr>
</tbody>
</table>

Figure 15: Round 1: Establishing the benchmark data set trained on 70k real FFHQ images. Step 1: Train the generator. Step 2: generate 10k fake images. Step 3: Input 10k real and 10k fake images into the detector. Step 4: Evaluate the detector results. Step 5: Filter images. This figure shows a sample of the 10k fake images generated. FID score: 4.5. Images that fooled the detector: 621. Detector Accuracy: 94.59%

In Round 1, we established the benchmark for comparing results at each training iteration. We
produced 10k fake images that were trained solely on real images. Since this round was trained only on real images, this round acts as the best comparison for the deepfake quality. We want to be able to produce deepfakes that are similar to this set, however, do it using the fake images generated each iteration. A sample of the deepfakes produced this round are shown in Fig. 15. These images have a low FID score of 4.5. Next, we input 10k real images and these 10k fake images into the detector to see the detector’s initial accuracy. The detector produced an accuracy of 94.59% for detecting fake images. Of these fake images, 621 images fooled the detector. These 621 images that fooled the detector will be used within the next round to train StyleGAN3.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Train StyleGAN3 on 621 fake images from Round 1</td>
</tr>
<tr>
<td>2</td>
<td>Generate 10k images</td>
</tr>
<tr>
<td>3</td>
<td>Input 10k real and 10k fake images into detector</td>
</tr>
<tr>
<td>4</td>
<td>Evaluate detector results, accuracy &gt;65%</td>
</tr>
<tr>
<td>5</td>
<td>Filter images for next round</td>
</tr>
</tbody>
</table>

Figure 16: Round 2: First training round on fake images. Step 1: Train the generator on 621 filtered images from Round 1. Step 2: generate 10k fake images. Step 3: Input 10k real and 10k fake images into the detector. Step 4: Evaluate the detector results. Step 5: Filter images. This figure shows a sample of the 10k fake images generated. FID score: 38.4. Images that fooled the detector: 2,620. Detector Accuracy: 75.41%
Round 2 consists of resuming training on the StyleGAN3 network with the 621 images that fooled the detector from the previous round. Fig. 16 shows a sample of the 10k fake images produced from training StyleGAN3’s network on the optimal fake images. These images have an FID score of 38.4, higher than the previous round. These images were input into the detector, which had an accuracy of 75.41%, lower than the previous round. Since the detectors accuracy is >65%, we will retrain the generator. Out of this fake image dataset, 2,620 images fooled the detector and will be used to train the generator in the next iteration.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Train StyleGAN3 on 2620 fake images from Round 2</td>
</tr>
<tr>
<td>2</td>
<td>Generate 10k images</td>
</tr>
<tr>
<td>3</td>
<td>Input 10k real and 10k fake images into detector</td>
</tr>
<tr>
<td>4</td>
<td>Evaluate detector results, accuracy &gt;65%</td>
</tr>
<tr>
<td>5</td>
<td>Filter images for next round</td>
</tr>
</tbody>
</table>

Figure 17: Round 3: Step 1: Train the generator on 2,620 filtered images from Round 2. Step 2: generate 10k fake images. Step 3: Input 10k real and 10k fake images into the detector. Step 4: Evaluate the detector results. Step 5: Filter images. This figure shows a sample of the 10k fake images generated. FID score: 9.4. Images that fooled the detector: 2,620. Detector Accuracy: 68.40%

Round 3 follows the same steps as the previous round. The StyleGAN3 network resumed training
on the 2,620 images that fooled the detector from the previous round. Fig. 19 shows a sample of the 10k fake images produced from training StyleGAN3’s network on the optimal fake images. These images have an FID score of 9.4, significantly lower than the previous iteration. These images were input into the detector, which had an accuracy of 68.4% on this dataset, lower than the previous round. Out of this fake image dataset, 2,620 images fooled the detector and will be used to train the generator in the next round.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Train StyleGAN3 on 2620 fake images from Round 3</td>
</tr>
<tr>
<td>2</td>
<td>Generate 10k images</td>
</tr>
<tr>
<td>3</td>
<td>Input 10k real and 10k fake images into detector</td>
</tr>
<tr>
<td>4</td>
<td>Evaluate detector results, accuracy &lt;65%</td>
</tr>
<tr>
<td>5</td>
<td>Filter images for next round</td>
</tr>
</tbody>
</table>

Figure 18: Round 4: Step 1: Train the generator on 2620 filtered images from iteration 2. Step 2: generate 10k fake images. Step 3: Input 10k real and 10k fake images into the detector. Step 4: Evaluate the detector results. Step 5: Filter images. This figure shows a sample of the 10k fake images generated. FID score: 8.4. Images that fooled the detector: 7,927. Detector Accuracy: 64.24%

Round 4 is the last generator training for this cycle. StyleGAN3 network resumed training on the 2,620 images that fooled the detector from the previous round. Fig. 18 shows a sample of the
10k fake images produced from training StyleGAN3’s network on the optimal fake images. These images have an FID score of 8.4, lower than the previous iteration. These images were input into the detector, which had an accuracy of 64.24% on this dataset, lower than the previous round. Since the detector’s accuracy has reached below the threshold of 65%, the next round will consist of retraining the detector. Out of this fake image dataset, 7,927, or the majority of fake images, fooled the detector and will be used to train the detector in the next round.

<table>
<thead>
<tr>
<th>Round 5: Train the Detector</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step</strong></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

Figure 19: Round 5: Retrain the detector. Step 1: Train the detector on 10k fake images from Round 4. Step 2: Evaluate the detector results by inputting 10k real images and 10k fake images from Round 3. Detector Accuracy: 100%

Round 5 consists of retraining the detector. Since the detector has reached a threshold that is deemed inconclusive for being able to detect fake images, we will retrain it to improve its detection accuracy. The detector is trained on 10k fake images from Round 4, the dataset with the worst detector accuracy. After training, we input the dataset from Round 3 into the detector for validation. The detector’s accuracy went up to 100%.

Round 6 is the final round of experiments for testing ADVANCE. Despite the detector’s improved accuracy, we will retrain StyleGAN3 on the images that fooled the detector through all the iterations. Fig. 20 shows the images generated from training StyleGAN3 on the optimal images. These images achieve an FID score of 4.0. After inputting these images into the detector, the detector’s accuracy reached 96%.
Round 6

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Train StyleGAN3 on 7,927 fake images from Round 4</td>
</tr>
<tr>
<td>2</td>
<td>Generate 10k images</td>
</tr>
<tr>
<td>3</td>
<td>Input 10k real and 10k fake images into detector</td>
</tr>
<tr>
<td>4</td>
<td>Evaluate detector results, accuracy &gt;65%</td>
</tr>
</tbody>
</table>

Figure 20: Round 6: Train the generator on 7,927 filtered images from Round 4. Step 2: generate 10k fake images. Step 3: Input 10k real and 10k fake images into the detector. Step 4: Evaluate the detector results. This figure shows a sample of the 10k fake images generated. FID score: 4.0. Detector Accuracy: 96%

Overall, this section focuses on the experiments conducted to evaluate ADVANCE, a continuous and collaborative adversarial learning pipeline. The objective of this pipeline is to simulate an arms race between a detection system and a generation system. The Methodology section provides a detailed analysis of the experimental results.

The experimental results analysis aims to provide a comprehensive overview of the findings obtained from the conducted experiments described in the next section, Experiments. This analysis delves into the performance and effectiveness of ADVANCE in simulating an arms race between the detection and generation systems.

The analysis begins by evaluating the performance of the detection system. The detector’s
accuracy is computed to assess the system's ability to accurately identify and classify deepfake images produced by the generator. The results are presented and compared to benchmark scores established in previous studies, highlighting any improvements or shortcomings in the detection system's performance.

Next, the focus shifts to the generation system and its ability to create realistic deepfake images. The quality of the generated images is assessed through a visual analysis as well as using the Frechet Inception Distance (FID) score, which measures the similarity between the generated images and real images. The analysis examines the image quality and FID scores across different iterations of the adversarial learning process, indicating whether the generation system successfully improves its image generation capabilities over time.

Furthermore, the analysis explores the dynamic nature of the arms race between the detection and generation systems. It investigates the effectiveness of the collaborative learning approach employed in the ADVANCE pipeline, wherein the detection system provides feedback to the generation system to enhance its ability to evade detection. The impact of this collaborative feedback loop on the performance of both systems is analyzed and discussed.

Lastly, the limitations and potential future directions of the ADVANCE pipeline are discussed in the analysis. Areas for improvement, additional experiments and potential extensions of the pipeline are explored, aiming to provide insights for further research and development in the field of adversarial learning.

Overall, the experiment results analysis provides a comprehensive evaluation of the ADVANCE adversarial learning pipeline, shedding light on its performance, effectiveness and potential applications in simulating an arms race between detection and generation systems.

First, we will start by presenting and analyzing the results obtained from the detection system across multiple rounds. Particularly noteworthy are the results depicted in Fig. 22, which illustrate the fluctuations in accuracy throughout the experiments. Accuracy serves as a critical metric for evaluating the detector’s performance. According to our hypothesis, as the generator underwent training on optimal fake images, it was anticipated that the detector’s accuracy would decline. Once the detector reached a threshold accuracy of 65%, we initiated a retraining process focused on the images where the detector had performed inadequately. The expectation was that this retraining would lead to an increase in accuracy, thereby restarting the process anew.

Figure 22 illustrates the accuracy of the generator during each round. Round 1 consisted of deepfakes generated solely from real images sourced from the FFHQ dataset. As anticipated, the detector exhibited performance in line with the expectations outlined in the detector’s research paper,
achieving an accuracy of 94.59%. This value serves as a reference point for assessing the accuracy of subsequent images generated using deepfakes. It is expected for this number to decrease each round until re-training. After re-training, it is expected for this number to maintain or go above this 94.59%.

During Rounds 2-4, a noteworthy trend emerged, revealing a gradual decline in accuracy as the generator underwent training exclusively on optimal fake images. This anticipated outcome can be attributed to the deliberate focus on images that effectively deceive the detector. The selection of these images was determined based on the confidence scores, denoted as $c$, generated by the detector at each round. These confidence scores serve as a measure of the detector’s assessment of the images’ authenticity, reflecting the extent to which an image is perceived as synthetic.

To identify the optimal images, a threshold criterion was applied, classifying images with confidence scores below 60% as candidates for retraining the generator. This careful selection process aimed to extract the most effective adversarial examples—images that convincingly fooled the detector into perceiving them as authentic despite their synthetic nature. By retraining the generator using these optimal images, the subsequent iterations intentionally aimed to challenge the detector’s robustness, fostering an adversarial relationship between the generator and the detector.

It is important to note that this decrease in accuracy during the training process does not indicate
a failure on the part of the detector. On the contrary, this decline is an inherent aspect of the adversarial training paradigm, where the generator continually strives to produce more convincing deepfakes while the detector adapts to detect them. The purpose of this process is to ultimately enhance the detector’s ability to identify and distinguish between real and fake images, as it is continually exposed to increasingly sophisticated adversarial examples.

Now, let us shift our attention to evaluating the performance of the generator by examining the FID score across the different rounds. The FID score serves as a quantitative measure to assess the quality of generated images, with lower scores indicating better performance. However, it is essential to acknowledge the inherent challenge of qualitatively measuring image quality in generative applications. The ultimate objective of the generator is to produce images that not only demonstrate a generally low FID score but also maintain a sense of realism within the generated photos.

Figure 22 presents the FID score of the generator throughout the rounds, along with the corresponding number of training iterations. It is important to note that Round 1 serves as the benchmark for image analysis and Round 5 did not include training, therefore, these rounds were omitted.

While the FID score does not necessarily have to decrease, the expectation is for it to eventually converge to a score around the benchmark value of 4.5.

In Round 2, as the training set transitions to fake images, a substantial increase in the FID score is expected, which aligns with our observations of an FID score of 38.4. This increase is attributed to the inherent challenge of generating realistic images solely from fake sources.

However, the noteworthy discovery lies in the subsequent rounds, specifically Rounds 3-6, where the FID score exhibits a gradual decline. This decrease signifies that the generator is progressively improving in maintaining realism within its generated images. The FID score experiences a decline from 9.4 in Round 3 to 8.4 in Round 4, eventually reaching an impressively low score of 4.0 after continuous training. These results indicate that the generator successfully enhances its capability to generate images that closely resemble real ones.

It is important to emphasize that while the FID scores show improvement, it is crucial to consider alternative units of measurement for evaluating the problem at hand, as visual analysis may suggest different conclusions. The subjective nature of image quality necessitates a comprehensive assessment that incorporates both quantitative metrics like FID and qualitative evaluation through visual inspection. By combining various evaluation approaches, we can gain a more comprehensive understanding of the generator’s performance and its ability to produce high-quality and realistic images.
Figure 22: Generator results represented by the FID score vs. rounds/training iterations. These results were as expected and imply the maintaining of realism throughout the generated fake image distribution.

Figure 23 provides a visual analysis of the generated images used in each round, shedding light on the evolution of the generator’s performance over time. A carefully selected sample of 16 images from training rounds 2-4 and 6 is presented, showcasing the generator’s behavior and ability to maintain realism throughout the training process. It is important to note that the grid numbers assigned to the images from each round correspond to images produced by different trained networks but belonging to the same distribution.

Observing the visual results, we can discern interesting patterns in the generator’s behavior. It is crucial to reiterate that our primary objective is not solely to generate visually superior images but rather to produce images that exhibit a sense of realism while effectively fooling the detector. Consequently, it is anticipated that the generated images may not possess the same level of visual appeal as those trained on real images. However, the generator demonstrates commendable performance in maintaining realism across each round.

Analyzing the corresponding photos, we can identify consistent facial features in images (1,6),
Figure 23: Rounds 2-3 & 6 generated image evolution for visual analysis. This allows us to see how the images change in realism despite continuous training on fake images.
(1,7), (3,5) and (4,6), indicating the generator’s ability to maintain consistency within a specific facial structure throughout the training process. Additionally, images (1,8), (3,7) and (4,7) exhibit variations in facial features while still preserving a notable level of realism.

On the other hand, images corresponding to (1,5), (2,5), (2,8), (3,6) and (4,4) demonstrate some realism but may be susceptible to an issue within GANs known as mode collapse. Mode collapse occurs when the diversity of the generated image set is limited due to a constrained generation distribution. Consequently, these images display similarities and may lack the desired variety.

Finally, there are instances where certain photos, such as (2,6), (3,1) and (4,8), do not exhibit a realistic appearance. These examples highlight the challenges and limitations of the generator in consistently producing convincing images throughout the training process.

Despite encountering some undesirable behavior, the generator still demonstrates its capability to generate images that maintain a sense of realism using the proposed method. However, it is important to highlight that the generator’s primary objective is to create images that increasingly fool the detector, ultimately compromising its performance.

Fig. 24 presents a curated set of images that successfully deceived the detector in each round of the experiments. These images do not exhibit a direct correlation to one another, indicating some variations in the distribution of selected photos across rounds. For instance, certain images that fooled the detector in Round 2 also manage to deceive it in Round 3, while some images from Round 1 may not be present in subsequent rounds. This lack of consistency in the selection of deceptive images suggests that the generator’s ability to fool the detector does not solely rely on a specific set of visually similar images.

Furthermore, the images featured in Fig. 24 do not necessarily have to maintain a high level of realism to successfully deceive the detector. Instead, the focus is on generating images that exploit the vulnerabilities of the detector’s black-box model.

It is noteworthy to observe that as the generator is trained on images that have previously fooled the detector, the number of images successfully deceiving the detector tends to increase. For instance, the round trained on real images resulted in a relatively lower number of images (621) that managed to deceive the detector, compared to the round trained on fake images which saw a significantly higher number (7,927) of deceptive images. This finding suggests that the generator’s ability to create images that effectively fool the detector improves with training on deceptive examples.

The lack of a clear pattern in the distribution of deceptive images implies that the detector’s black-box model may be operating in an unexplainable manner. It is possible that the detector is detecting subtle features or patterns that are beyond human perception or the conventional FID.
Figure 24: Round 1-4, images that fooled the detector. These results show us the type of images that fooled the detector and potentially point out features that distinguish fake from real images.
score. This observation highlights the complex nature of the detector’s decision-making process and suggests the presence of additional factors that contribute to its detection accuracy.

After carefully examining the individual performances of the generator and detector, we can now assess the overall ADVANCE system based on the results obtained. Figure 25 provides a comprehensive visualization of the generator’s FID score and the detector’s accuracy, represented by the blue line below. The trends observed in this figure indicate that as the generator undergoes further training on optimal data, the performance of the detector simultaneously declines. This correlation suggests that the generator’s ability to produce realistic deepfake images hampers the detector’s effectiveness in accurately identifying them.

However, an interesting observation arises from this reciprocal relationship. The newly generated data by the generator, although successful in fooling the detector, can be utilized to improve the robustness of the detector itself through iterative training. As depicted in Round 5 of Figure 25, once the detector is retrained using the generated data, it exhibits a remarkable increase in accuracy, approaching nearly 100%. This finding highlights the potential of leveraging the adversarial nature of the generator-detector relationship to enhance the detector’s performance and adapt it to evolving deepfake techniques.

Figure 25: Round 6, more samples of images that fooled the detector.

After completing the training of the detector in Round 5, the system enters a new iteration. The generator now has the opportunity to train on images that still manage to deceive the retrained detector. It is worth noting that the set of images that fool the detector, shown in Fig. 25, appear to exhibit more visual patterns compared to previous rounds. However, due to the black-box nature of the experiment, it is challenging to draw definitive conclusions about this observation. Further investigation is necessary to delve deeper into the underlying mechanisms and dynamics at play.
within the generator-detector interplay.

In summary, the ADVANCE system demonstrates the capability of the generator to generate realistic images that deceive the detector, while the detector showcases the potential for self-improvement through iterative retraining. The curated set of deceptive images serves as evidence of the generator’s ability to compromise the detector’s performance. Nonetheless, the absence of a clear and consistent pattern in the distribution of deceptive images underscores the complexity and unexplainable nature of the detector’s decision-making process. These experimental findings provide valuable insights into the dynamic and adversarial relationship between the generator and the detector, contributing to the ongoing development and evaluation of deepfake detection systems. Further research is necessary to explore strategies for improving the overall performance and resilience of such systems in the face of evolving deepfake techniques.
Chapter 4

CONCLUSION AND FUTURE WORK

In conclusion, this thesis has addressed the pressing concern surrounding adversarial learning methods and their impact on the authenticity of information in the digital landscape. While advancements have been made in detecting adversarial generated data, the reliance on non-iterative feed-forward designs has led to an ongoing arms race between machine learning and detection systems. To tackle this challenge, the main research objective of this thesis was to develop a continuous and collaborative pipeline, known as ADVANCE (Adversarial Collaborative Learning for Detection and Verification of Artificially Created Examples), that fosters an iterative process between a generative system and a detection system to simulate an arms race. Throughout this thesis, a comprehensive examination of the field of adversarial learning within cyber security has been conducted, encompassing the methods of generating and detecting fake data. Moreover, a computational analysis of the ADVANCE pipeline has been undertaken, leading to significant findings and insights. The results indicate that ADVANCE demonstrates stability and effectiveness in enhancing the detection of adversarial generated data.

The experiments conducted in this study aim to investigate the effectiveness of the ADVANCE system. The findings from the experiments revealed several significant outcomes, shedding light on the capabilities and limitations of the ADVANCE system. The first notable finding was related to the performance of the generator. The evaluation of the generator’s performance using the Fréchet Inception Distance (FID) score demonstrated a gradual decrease over the training rounds. This decrease indicated that the generator was able to produce images that maintained a sense of realism while effectively fooling the detector. Although the visual analysis of the generated images revealed some variations in quality and realism, the overall trend demonstrated the generator’s ability to deceive the detector effectively.

The second significant outcome pertained to the performance of the detector. As the generator improved its ability to produce deceptive images, the detector’s accuracy exhibited a corresponding
decrease. This finding aligned with the research objective of creating an arms race between the generator and the detector. The decrease in the detector’s accuracy indicated that the generator was successful in compromising the performance of the detection system, highlighting the adversarial nature of the relationship between the two components.

Moreover, the experiments showcased the iterative nature of the ADVANCE system. After retraining the detector with the newly generated images, its accuracy improved significantly, achieving an almost 100% detection rate. This result demonstrated the system’s ability to leverage the generator’s output to enhance the detection capabilities, thus creating a more robust system. The iterative nature of the pipeline highlighted the dynamic interplay between the generator and the detector, where each component adapted and improved in response to the other, emphasizing the complexity of the adversarial relationship.

This research has made several significant contributions to the field of study. Firstly, by developing and evaluating the ADVANCE system, this research has provided a comprehensive and practical framework for addressing the authenticity of information in the presence of adversarial learning methods. The iterative and collaborative pipeline of the ADVANCE system represents a novel approach in the field, enabling the continuous improvement of both the generator and the detector.

Furthermore, this research has built upon and extended existing knowledge in multiple ways. Firstly, by incorporating an intelligent reasoning algorithm into the ADVANCE system, the study has demonstrated the potential of combining advanced algorithms with generative and detection systems. This integration not only enhances the performance of the system but also contributes to the growing body of research on intelligent reasoning in the context of adversarial learning.

Moreover, the research has addressed the gap in the literature regarding the dynamics between the generator and the detector in an adversarial setting. By examining their interplay and presenting experimental findings, this study has shed light on the complex and dynamic nature of their relationship. This understanding is crucial for developing more robust detection systems and improving our ability to combat the challenges posed by adversarial generated data.

The findings of this study align with previous research explored in the literature review in several aspects. The utilization of GANs as a generative model and the iterative training process to enhance the generator’s performance are consistent with existing studies. The effectiveness of GANs in maintaining realism while fooling the detector was also observed, validating the findings of prior research in adversarial learning. As well as the recognition of the arms race between generative systems and detection methods. The understanding that advancements in one domain lead to countermeasures in the other is widely acknowledged in the literature and is consistent with the
findings of this study.

Furthermore, the study’s focus on the dynamic nature of the adversarial relationship between the generator and the detector, and the concept of an arms race, aligns with the evolving understanding of adversarial learning in the literature. This highlights the importance of continuously improving both the generative and detection systems to stay ahead in this ongoing battle.

However, some deviations from previous studies were also identified. The incorporation of an intelligent reasoning algorithm as part of the ADVANCE system is a unique contribution of this research. While previous studies primarily focused on the generator-detection relationship, the integration of intelligent reasoning adds a new dimension to the adversarial learning framework, enabling intelligent decision-making and adaptation throughout the iterative process.

The methodology and experimental design employed in this study have both strengths and weaknesses that warrant evaluation. By critically reflecting on the appropriateness and effectiveness of the chosen methods, as well as addressing challenges and limitations encountered during the experiments, we can gain insights for potential improvements and alternative approaches in future studies.

One of the strengths of the methodology is the integration of two state-of-the-art detection and generation methods which allowed for the continuous improvement of both the generator and the detector. This approach acknowledges the dynamic nature of adversarial learning and provides a practical application for addressing the arms race between generative systems and detection methods. The use of an intelligent reasoning algorithm further enhanced the system’s performance, as it facilitated intelligent decision-making and adaptation throughout the iterative process. Each of these contributions allow for a flexible and generalizable system for adversarial learning methods.

Another strength of this methodology is that the system is versatile and can be used to minimize or maximize the detector’s performance. This is beneficial for creating more robust systems. It allows for designers to exploit their systems iteratively in order to maximize performance.

Moreover, the experimental design incorporated a comprehensive evaluation of the ADVANCE system by considering multiple factors such as FID score, detection accuracy and visual analysis of generated images. This multi-faceted approach ensured a thorough assessment of the system’s performance and provided a more holistic understanding of its effectiveness.

However, there were also weaknesses and limitations associated with the methodology and experimental design. One limitation is the reliance on a single dataset for training and evaluation. Future studies could consider incorporating multiple datasets or conducting experiments on diverse datasets to enhance the robustness and generalizability of the results. Furthermore, the methodology
also relies on large amounts of data in order to train and evaluate, making the experiments more difficult to carry out.

Another weakness is the black-box nature of the detector. Although the detector’s accuracy and performance were evaluated, the lack of transparency regarding its decision-making process poses challenges in understanding its inner workings. This limitation restricts the ability to precisely identify the features or patterns that the detector uses for detection. Developing methods to gain more insights into the detection process would be valuable for improving the interpretability and explainability of the system.

The experiments also encountered challenges related to the evaluation metrics. While FID score and detection accuracy were used as quantitative measures, assessing the qualitative aspects of generated images and their realism remained subjective. The reliance on human judgment introduces inherent biases and limitations. Incorporating additional metrics or exploring alternative evaluation methods, such as perceptual similarity metrics or user studies, could help address these challenges and provide a more comprehensive evaluation framework.

In conclusion, this research has contributed to the field of study by presenting the ADVANCE system, a continuous and collaborative pipeline for addressing the authenticity of information in the presence of adversarial learning. By incorporating an intelligent reasoning algorithm and examining the dynamics between the generator and the detector, this study has extended existing knowledge and provided valuable insights. However, limitations in the understanding of the detector, the focus on image-based adversarial learning and the need for countermeasures against adversarial attacks highlight avenues for future research. By addressing these gaps and limitations, future studies can further enhance the effectiveness and applicability of adversarial detection and verification techniques, ensuring the integrity and trustworthiness of digital content in an evolving landscape of adversarial learning.

Moving forward, the findings presented in this thesis provide a solid foundation for further research and development in the field of adversarial learning and detection systems. The ADVANCE pipeline holds promise for future advancements and enhancements, fostering a more secure and trustworthy digital environment. Continued exploration and refinement of the pipeline will contribute to the ongoing arms race between generative and detection systems, ultimately strengthening the defense against adversarial attacks and ensuring the integrity of digital content.

Potential avenues for future research based on the findings and limitations of this study include integrating multimodal analysis that considers multiple modalities, such as audio, text and video, to detect inconsistencies and discrepancies across different media types. Due to its generalizable nature,
it would be interesting to test this method on different real-word applications. Another avenue for future work is developing efficient and scalable algorithms that can detect deepfake content in real-time, enabling timely intervention and mitigation of potential harm. Due to the limitations of time and resources for this experiment, I believe a more efficient and scalable system is possible. Finally, given the black-box nature of detection systems, future research can focus on developing methods to better interpret and explain the decision-making process of these systems. This would provide valuable insights into the features and patterns that contribute to the detection of deepfake content, enhancing our understanding of the underlying mechanisms.

This research makes significant contributions to the field by advancing our understanding of adversarial learning and deepfake detection. The ADVANCE framework provides valuable insights into the development of more robust detection systems and offers practical implications for addressing the challenges posed by deepfake content. As deepfake technology continues to evolve, this work emphasizes the urgency of ongoing research, collaboration, and proactive measures to safeguard against the potential misuse of AI-generated synthetic media. It is our collective responsibility to ensure the integrity and trustworthiness of digital information in an increasingly complex and interconnected world and continue on with this research.


[35] W. Hu and Y. Tan, “Generating adversarial malware examples for black-box attacks based on


models to understand and defend against adversarial examples,” 2018.

[38] S. Gu and L. Rigazio, “Towards deep neural network architectures robust to adversarial examples,”
2015.


perturbations against deep neural networks,” in 2016 IEEE Symposium on Security and Privacy
(SP), 2016, pp. 582–597.

[41] M. Abbasi and C. Gagné, “Robustness to adversarial examples through an ensemble of specialists,”
2017.

[42] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath,


in Proceedings of the 34th International Conference on Machine Learning, ser. Proceedings of


