Generating Adversarial Examples through Latent Space Exploration of Generative Adversarial Networks

Luana Clare
University of Coimbra, CISUC, DEI
Coimbra, Portugal
luanasantos@student.dei.uc.pt

João Correia
University of Coimbra, CISUC, DEI
Coimbra, Portugal
jncor@dei.uc.pt

ABSTRACT

Artificial Neural Networks are vulnerable to adversarial examples, malicious inputs that aim to subvert neural networks' outputs. Generative Adversarial Networks (GANs) are generative models capable of generating data that follows the training data distribution. We explore the hypothesis of using the latent space of the trained GAN to find adversarial examples. We test the adversarial examples on external classifiers trained on the same training data. Thus, we propose a framework for Generating adversariaL exAmpleS through latent Space Exploration (GLASSE). A Genetic Algorithm evolves latent vectors as individuals and uses a trained GAN to generate examples to maximise a target activation value of the discriminator network. After the evolutionary process, an external classifier trained on the same dataset evaluates whether it is adversarial. The results indicate that we can optimise the objective and find adversarial examples. We tested the generated examples with models from the adversarial learning literature, showing that 82% on average of the generated examples resulted in successful attacks. We show a t-SNE analysis of the examples, showcasing that generated adversarial examples are blended in the cluster of each belonging class and visually similar to the training dataset examples, showcasing the viability of the proposed approach.

CCS CONCEPTS

• **Computing methodologies** → **Genetic algorithms**; **Generative and developmental approaches**.

KEYWORDS

Adversarial Examples, Generative Adversarial Networks, Evolutionary Computation, Latent Space Exploration

ACM Reference Format:
Luana Clare and João Correia. 2023. Generating Adversarial Examples through Latent Space Exploration of Generative Adversarial Networks. In *Genetic and Evolutionary Computation Conference Companion (GECCO '23 Companion), July 15–19, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3583133.3596392

GECCO '23 Companion, July 15–19, 2023, Lisbon, Portugal

ACM ISBN 979-8-4007-0120-7/23/07.
https://doi.org/10.1145/3583133.3596392

1 INTRODUCTION

Adversarial examples are created to attack the machine learning model, causing it to misclassify with high confidence. Typically these examples are created from the modification of existing data, from the testing or training, with only small perturbations and mostly imperceptible. The interest in adversarial examples rises from the fact that even state-of-the-art deep neural networks (DNNs) are vulnerable to such manipulations [16]. The topic of adversarial attacks and defences is widespread and can help us understand the model's assurance, security and generalization. Studying the attacks to understand better the vulnerability and weakness of DNNs is essential to develop a defensive strategy for more robust models. For this purpose, many algorithms have been proposed to create adversarial examples, such as the fast gradient sign method [11], the backward pass differentiable approximation [1], the AdvGAN method [18] and the GreedyFool method [8].

Generative Adversarial Networks (GANs) can generate synthetic but convincing images following a given training dataset distribution. GANs are composed of two models: a generator and a discriminator. During training, the interplay between the generator and discriminator allows the generator to learn how to generate realistic examples that the discriminator classifies as belonging to the training dataset [10]. With a GAN trained with a dataset of multiple classes, it is hard to precisely control the appearance (e.g. class) of the examples being generated. In contrast, a Conditional GAN offers the opportunity to condition the generator to produce examples from a given class [14].

After the model's training, the generator maps a latent space to synthetic examples that follow the distribution of the training dataset. In simple terms, the latent space can represent compressed data where similar data points are closer together in the latent space. Thus, for each image to be generated, the generator of a GAN receives as input a vector from the latent space. As its training progresses, the generator learns to produce images that follow the distribution of the original dataset, encoding the patterns needed to produce these images in the latent space. Each image created by the GANs is originated from a vector, in which a value is selected for each latent space dimension. By varying the vector values, we generate different images.

Thus, in this paper, we search the latent space for examples generated by a GAN that are adversarial when classified by an external system, a classifier trained for the same problem and data. Its primary focus is to explore the generator's latent space with a genetic algorithm (GA), guiding a population of latent vectors towards the generation of adversarial examples. This work takes inspiration from the latent space exploration via Evolutionary means previously used in Fernandes et al. [9] to guide the generation of

images in batches to maximize the diversity of the batch, which motivated us to explore this matter further. In the GA from GLASSE, the evolution of the latent vectors is based on the discriminator loss of a trained GAN. The goal of targeting the discriminator loss is to get feedback on how close the images are to the distribution of the original dataset. The generator of a GAN takes the latent vectors as input and produces images that are submitted to an independent classifier, which evaluates the images. The vectors that lead to the desired discriminator loss and misclassifications by the classifier are potential adversaries. GLASSE shows how it is possible to explore the latent space based only the feedback of a discriminator of a GAN, and find latent vectors that result in the generation of adversarial examples. Moreover, it is possible to see how generated images follow the original dataset distribution through an analysis with t-SNE maps [7].

In this paper, we work towards an evolutionary approach that uses latent space exploration to create a black-box adversarial attack that only requires a trained GAN and an independent classifier. The contributions are the following: (i) proposal and prototyping of a framework for the generation of adversarial examples via latent space exploration using Evolutionary Computation; (ii) experiments using the framework and subsequent analysis to provide a proof of concept; (iii) validation of adversarial examples by successful attacks on an undistilled network and a defensively distilled network; (iv) map visualization of proximity between generated adversarial examples and original dataset.

The remainder of this paper is divided as follows. In Section 2, we cover different approaches to generate adversarial examples related to this work. In Section 3, we describe our approach with the GLASSE method. We present our experimental setup in Section 4. Our results are presented in Section 5, and our final conclusions and future work are in Section 6.

2 RELATED WORK

In this section, we review related work relevant to the GLASSE framework, namely regarding GANs, latent space exploration and related adversarial attacks.

GANs [10] are powerful generative models capable of producing highly realistic images, videos or voice outputs. It is a state-of-art model from which many models are derived. There is much research on GANs to make them more reliable and capable of producing better and more diverse images. However, most of this research focuses on the models and their training. Nevertheless, the latent space, which is the generator's input, defines the output produced and hides information that can be used to produce better examples. Usually, the vector input is taken randomly from the latent space of a GAN. However, recent work has shown that it is possible to guide the generation towards an objective by traversing the latent space with pre-determined criteria [9]. The ELSEGANs approach successfully used a genetic algorithm and a Multi-dimensional Archive of Phenotypic Elites to search the latent space for vectors that would generate diverse images by a Deep Convolutional GAN. In this paper, we aim to use a genetic algorithm to search for vectors in the latent space that result in the generation of adversarial examples by a Conditional GAN.

Moreover, the evolutionary computation was used in Roy et al. [15] to evolve fingerprints at the feature level towards a synthetic dictionary of MasterPrint, which are partial fingerprints that match with other fingerprints and compromise the security of fingerprint-based authentication systems by impersonating one or more users. They proposed three techniques to achieve their goal: Covariance Matrix Adaptation Evolution Strategy (CMA-ES), Differential Evolution (DE) and Particle Swarm Optimization (PSO). In Bontrager et al. [2], evolutionary computation is used to evolve latent vectors to generate image-level MasterPrints known as DeepMasterPrints. The method is called Latent Variable Evolution and consists of training a GAN with a set of real fingerprint images with a CMA-ES to search for latent vectors, use them as input to the generator of the GAN, and maximize the number of impersonations as assessed by a fingerprint recognizer. In contrast, GLASSE uses a GA with an objective function that does not use an external model to guide evolution; it is guided only based on the feedback of the GAN's discriminator. The work was expanded in Charity et al. [4] with the introduction of Diversity and Novelty MasterPrints. The diversity algorithm aims to generate partial fingerprints that can impersonate users that were not impersonated previously, i.e., whenever there is a new search, it changes the objective metric. It excludes users that can already be impersonated. In turn, the novelty algorithm aims to search for fingerprints far from each other.

Again, evolutionary computation and latent space exploration was used in Machín et al. [13] through an evolutionary algorithm to generate synthesized images of human faces that were similar to a target human face. The work was expanded in Machín et al. [12], where a method also based in an evolutionary algorithm aimed to generate synthesized images of human faces that contain the main features of two target faces.

Adversarial examples are inputs manipulated with the purpose of confusing machine learning models. The discriminator of a GAN is vulnerable to adversarial attacks. Many white-box settings have been proposed to generate adversarial examples, such as the fast gradient sign method (FGSM) [11] and the GreedyFool method [8]. While both attacks rely on gradient information to generate a perturbation to the original input, we propose a genetic algorithm to guide the generation of adversarial inputs in our approach. Genetic algorithms were also used in Chen et al. [5] and in Wu et al. [17] to find adversarial examples by searching for perturbation to be applied to existing examples. With GLASSE, we use a Genetic Algorithm to explore the latent space of GAN and search for adversarial examples.

Conditional GANs were previously used to generate adversarial examples in Xiao et al. [18]. The AdvGAN method can be used in both a semi-white and a black box scenario. In the semi-white box attack, there is only necessary to have access to the target model during the training of the generator and the discriminator. During training, the generator is responsible for creating a perturbation to the original instance that will increase the loss of the target model. On the other hand, the discriminator encourages the perturbed data to appear similar to the original data. A soft hinge loss on the L_2 norm is used to bound the magnitude of the perturbation. There is no access to the target model for the black box scenario. Therefore, before the training of the GAN, a distilled model is dynamically trained for the target model with query information. Although we

also explore using Conditional GANs to generate realistic adversarial examples, the vectors given as input to our generator are not randomly drawn from the latent space but instead selected by our genetic algorithm. Differently from the AdvGAN approach, which trains the generator to produce adversarial examples, the models in the GLASSE framework are all traditionally trained. Moreover, while the AdvGAN approach actively produces adversarial images targeting a classification model, the GLASSE method explores the latent space based only on the discriminator loss and finds adversarial examples without explicitly producing them for an external targeted model.

3 THE APPROACH

We first start with the problem definition and explain how the GLASSE framework operates to tackle the problem of generating adversarial images via latent space exploration of trained conditional GAN.

3.1 Problem Definition

Let $Z \subseteq \mathrm{R}^n$ be the latent space, defined according to a Random distribution. Firstly, we initialize and register a random set of i vectors z_i with n features each. They will serve as individuals to the initial population.

The generator G takes an individual z as input and generates an image $G(z)$. The algorithm aims to select and evolve the individuals to find adversarial examples. In order to do that, the algorithm evaluates each z with an evaluation function f that assigns them a fitness value. The function f is the main factor that will guide the evolution of the population into the desired solutions. The adversarial examples must trick the neural network into misclassifying them. Our best individuals will be the ones that fool the discriminator D, making it uncertain if it is a real or a fake image. The discriminator loss we aspire to achieve for this scenario is represented by $target$.

$$f(z) = 1 - |\mathcal{L}_D(G(z)) - target| \quad (1)$$

3.2 GLASSE Method

We aim to achieve our objectives by proposing a new framework named Generating Adversarial Examples through Latent Space Exploration (GLASSE). The architecture used in the GLASSE framework is shown in Figure 1.

We use a GA as the Evolutionary approach, a GAN and a classifier model C. The generator from the GAN is responsible for producing our examples, and the discriminator for helping the evaluation of individuals in the GA and serves as our target for the attack. The classifier is an independent observer who serves as a supervisor, as in Correia et al. [6], which is used to identify if the generated example is adversarial by confirming the label and activation of the generated example with a predefined target.

To calculate the fitness value of an individual, f submits the individual as an input to G. The generated image $G(z)$ is passed to D, whose decision y is returned to f. Finally, f can calculate the loss of D. The fitness value is given by equation 1, where z is the individual (latent vector), $G(z)$ is the generated image, and $L_D(G(z))$ is the discriminator loss for $G(z)$. The algorithm evaluates the whole population of individuals.

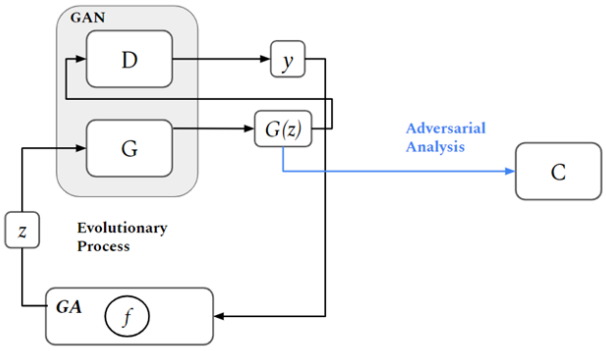

Figure 1: Overview of the GLASSE Architecture.

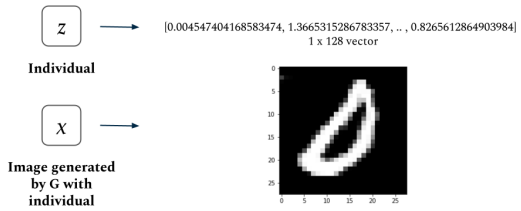

Figure 2: Example of individual with 128 float attributes and its image produced by the generator of a Conditional GAN trained with the MNIST dataset and conditioned by the number 0.

Figure 2 shows an example of an individual generated with a Conditional GAN trained with the MNIST dataset. This is an individual with 128 attributes, i.e., an array of floats of shape $1x128$.

For the following generation, the GA keeps an elite from the current one. An elite size gives the number of individuals kept from the current generation. In addition, it uses a tournament to select the parents and applies various techniques to generate the rest of the next generation.

Once the Evolutionary Process is done, the image $G(z)$ is generated from the vector z and submitted to C to analyse the adversity of the fake images. This is done for every individual in the population.

The GLASSE algorithm is presented in Algorithm 1, and was implemented using the Distributed Evolutionary Algorithms in Python (DEAP) library version 1.3.3, implemented in Python 3.9. The code for the algorithm is available in the GLASSE repository [1]. It is executed with the aim of maximising the individual fitness value in order to achieve the targeted discriminator loss. In the algorithms, the hyperparameters *npop*, *n*, *generations*, *ELITE_SIZE*, *CXPB*, *MUTPB*, *mu*, *sigma*, *indpb*, and *number*, are, respectively, the number of individuals in a population (population size), the number of features of an individual, the number of generations, the absolute size of the elite, the crossover rate, the mutation rate, the gaussian mutation mean, the gaussian mutation standard deviation, the mutation rate per gene, and the number to be generated by the

[1] https://github.com/luanaclare/GLASSE

Algorithm 1: GLASSE Algorithm

Input: Generator G and discriminator D from trained GAN, trained classifier model C, evaluate function f.
Output: Individual with highest fitness value z, image $G(z)$ generated by G with z, classification of $G(z)$ given by C

```
/* Evolutionary Process                                    */
```
Initialize *npop* individuals with *n* features, where each feature is drawn from a normal distribution, to create a *population*
```
/* Evaluation of initial population                        */
```
fitnesses = f(*population*)
population = sort(*population*, fitnesses, reverse=True)
for $i \leftarrow 1$ **to** *generations* **do**
  offspring = select(*population*, *npop* − *ELITE_SIZE*)
  offspring = variationOperators(*population*, *CXPB*, *MUTPB*, *mu*, *sigma*, *indpb*)
  fitnesses = f(offspring)
  population = sort(*population*, fitnesses, reverse=True)
  population = *population*[:*ELITE_SIZE*] + offspring
  population = sort(*population*, fitnesses, reverse=True)
```
  /* Adversarial Analysis                                  */
```
  images = G(*population*, *number*)
  classifications = C(images)
end
z = best individual from *population*
$G(z)$ = image generated with best individual

Table 1: Parameters for the training of the Classifier and the Conditional GAN

Parameter	Classifier	Conditional GAN
batch size	64	64
number of classes	10	10
number of channels	-	1
input shape	(28, 28, 1)	discriminator (28, 28, 11) generator (138,)
latent dimension	-	128
number of epochs	5	100
learning rate	0.001	0.0003
loss function	Categorical crossentropy	Binary cross-entropy
optimizer	RMSprop	Adam
epsilon	1e-8	-

generator. The corresponding values are explored in Section 4 and Table 2.

4 EXPERIMENTAL SETUP

In our experiments, we explore the MNIST dataset. Here, we define the setup for training the models used in the GLASSE architecture: the Classifier and the Conditional GAN. We also set the genetic algorithm parameters that led to our most successful experiment, which will be further explored in Section 5.

Table 2: Parameters for GA in the Experiment

Parameter	Setting
number of features	128
number to be generated	0
population size	100
number of generations	40
elite size	1
desired discriminator loss (target)	0.5
type of crossover	two-point crossover
type of mutation	gaussian
crossover rate	1
mutation rate	1
mutation rate per gene	0.1
gaussian mutation mean	0
gaussian mutation STD	3
tournament size	3

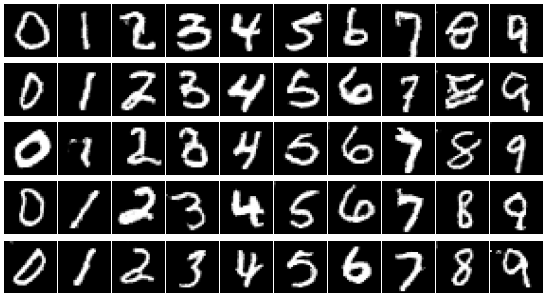

Figure 3: Images produced with Conditional GAN.

The classifier architecture was built according to the Tensorflow MNIST Classifier tutorial[2]. It receives an image of shape $28x28x1$ as an input and the last layer returns the probability for the image to be from each class. Each class represents a digit between 0 and 9, creating an output shape of 1x10. We trained our classifier C with the parameters in Table 1 and a total of 60,000 images from the MNIST dataset were used for the training. We obtained an accuracy of 99.25% when testing it with 10,000 images of the test set.

The approach requires a discriminator and a generator. For this reason, a Conditional GAN must be previously trained, and its weights loaded to a discriminator D and a generator G. Once the models are functional, they can be used in the Evolutionary Process. They were defined according to the Tensorflow Conditional GAN tutorial[3], and trained with the entire MNIST dataset, with a total of 70,000 images. Although it can be conditioned to generate only one class, the GAN is trained with examples from all classes. Figure 3 shows images generated by our trained GAN.

While a parameter study would be relevant for the tuning of the GA hyperparameters, it is considered out of scope of the paper, as the focus is primarily on the exploration of the latent space towards the generation of adversarial examples. Therefore, a compromise between performance and time was done, and the parameter study left to be explored in future work. The combination of parameter

[2]https://www.kaggle.com/code/amyjang/tensorflow-mnist-cnn-tutorial/notebook
[3]https://keras.io/examples/generative/conditional_gan/

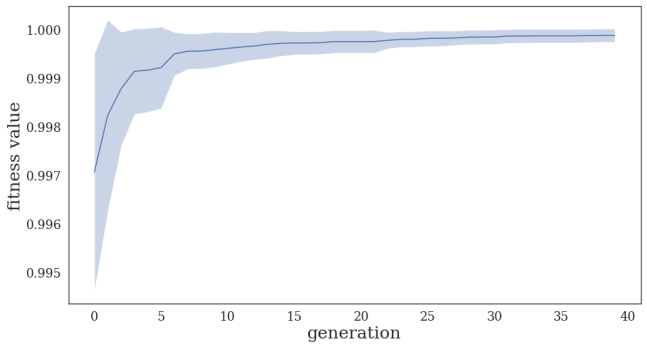

Figure 4: Fitness values of the best individual per generation. The results are averages of 30 different runs for class 0.

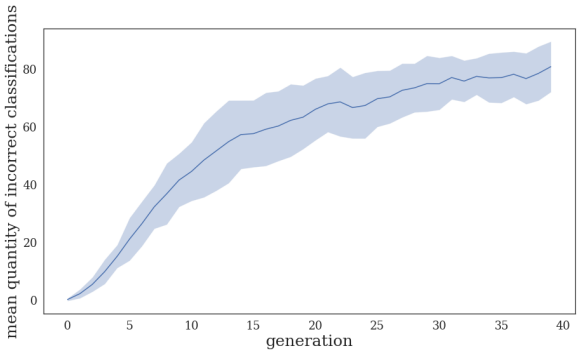

Figure 5: Number of wrongly classified examples per generation. Each generation has 100 individuals. The results are averages of 30 different runs for class 0.

values used in this paper is presented in Table 2. The target discriminator loss was set as 0.5 because the goal is to obtain images that confuse the model and are at their limit between real or fake images. The crossover and mutation rates were set as high values to aim for high diversity, exploration and exploitation. The results will be analysed in Section 5.

5 EXPERIMENTAL RESULTS

In this section, we explain and analyse experiments done with the GLASSE framework. We conducted 30 evolutionary runs, varying the random generator seed, with the parameters set in Table 2 to search the latent space for adversaries of class 0. The experiments were conducted in a PC with an AMD Ryzen 55600X 6-Core 3.7GHz with one Nvidia GPU 3080 TI, in which the Evolutionary Process showed an average runtime of 12.45 minutes. Then, an experiment was conducted for the other nine MNIST classes. Again, the parameters were set as in Table 2, with the exception of the number to be generated, which was set according to the class. The results obtained from the experiments and the analysis here discussed are available in the GLASSE repository [4].

5.1 Evolution of solutions

The genetic algorithm of the GLASSE framework aims to maximise the fitness value. Therefore the highest fitness belongs to the best individual. In Figure 4, it is possible to see that the Evolutionary Process is successful - the best individual's fitness (highest fitness) evolves as the generations progress. As it is possible to see, fitness stabilises close to number one in early generations, meaning that the desired discriminator loss is an easy target and quickly achieved by the best individual.

The goal of the experiments with the GLASSE framework and the genetic algorithm is to find individuals that can lead to the generation of adversarial examples. In this scope, we consider an example potentially adversarial if its discriminator loss is in the interval $[target - 0.01, target + 0.01]$, and the classifier C (Figure 1) mislabels it with an activation higher than 0.5. The range of 0.01 was empirically trialled and is a parameter that can be adjusted, and this was one value of the multiple possibilities. Nevertheless, the motivation is based on the typical loss of stable GAN loss values. The

[4]https://github.com/luanaclare/GLASSE

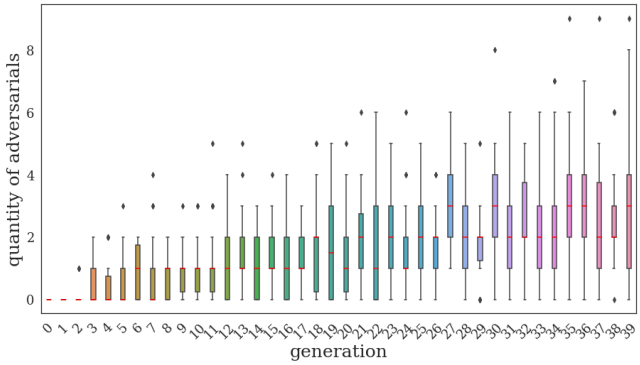

Figure 6: Box-plot of the quantity of potential adversarial examples per generation in 30 runs for the class belonging to digit 0. Each generation has 100 individuals.

aim of establishing this particular interval is to obtain individuals with discriminator loss close to the targeted value. On that account, there are two objectives that a potential adversarial example must oblige to. We consider the set of examples that achieve these two objectives our adversarial pool. However, as discussed before, the evaluation done in the genetic algorithm aims to maximise the fitness loss, i.e. to approximate an individual's discriminator loss to the value of the targeted discriminator loss. Thus, the evolution of individuals only takes into consideration the discriminator loss - the activations of the adversarial analysis classifier are not present in the fitness function, therefore, do not play a part in the evolution and only participate in the triage of individuals for the adversarial pool after the evolution.

In the first set of experiments, the generator of the Conditional GAN is conditioned only to produce examples of the class of digit 0; we will shorten the designation hereon to class 0. Therefore all examples evaluated here are classified as class 0. To find our potential adversarial examples, we first searched the individuals that when submitted to the Adversarial Analysis by C resulted in wrong classifications with activations higher than 0.5. Figure 5 shows the mean quantity of misclassifications per generation. Although

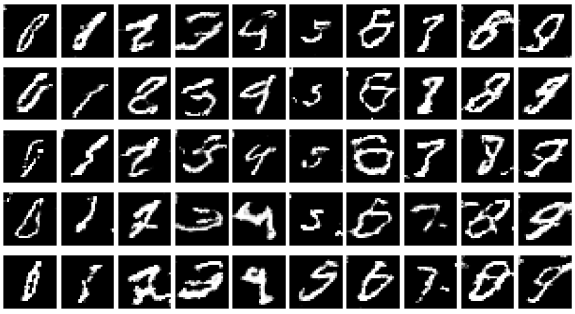

Figure 7: Examples of actual adversarial images generated with GLASSE.

there are many misclassifications, not all of them are adversarial - some could be noise images. It was necessary to analyse their discriminator loss to narrow down the misclassified examples to potentially adversarial ones - those with loss in the interval around the target. However, we found that most individuals in a population do not present discriminator losses in the desired interval. Therefore, the quantity of potential adversaries is clearly lower than that of misclassifications. The box plot in Figure 6 shows the number of potential adversaries per generation across thirty runs. In the first three generations, there are no potential adversaries despite outliers. However, a higher amount of the box-plot begins to appear in generation 3. Before generation 18, there was still a low median quantity of potential adversarial examples, with outlier runs in generations 11 and 13 achieving good results. Though, there is an increase in the median quantity of potential adversaries as the generations progress, with the highest values reached between generations 27 and 39. However, there are one hundred individuals in a population, and, compared to the whole, few potential adversaries were found.

The actual adversarial images are the ones we searched for with the GLASSE framework. They not only possess a discriminator loss in the desired interval and are misclassified by the classifier as not belonging to the conditioning class with an activation higher than 0.5, i.e belonging to the adversarial pool, but also look like examples of the conditioning class to the human eye. Figure 7 shows selected adversarial examples from the experiments conducted with the GLASSE framework for one of the ten classes at a time. It is possible to see that some are similar to the examples from the Conditional GAN in Figure 3.

5.2 Evaluation of generated adversarial examples

In order to evaluate the potential adversaries, the pool of potential adversarial examples was submitted to two models from Carlini et al. [3] work: model A (undistilled network) and model B (defensively distilled network). To clarify, in this process, the submitted examples are classified by the models and if a given example is classified as belonging to another class than the designated ground truth label, it is rendered as a successful attack. The potential adversarial examples from a run correspond to all the individuals from the Evolutionary Process of GLASSE conditioned to the ten classes

Table 3: Overview of successful attacks to models A and B with the possible adversaries from all classes produced with GLASSE. Columns are: Evolutionary Run (Run), possible adversaries (P. Adv), adversaries for model A (Adv A), adversaries for model B (Adv B), and adversaries for model C (Adv C). Best run values for model A and model B is marked in bold.

Run	P. Adv	Adv A	Adv B
1	548	**480 (87.59%)**	489 (89.23%)
2	538	436 (81.04%)	437 (81.23%)
3	661	528 (79.88%)	542 (82.00%)
4	616	492 (79.87%)	441 (71.59%)
5	597	513 (85.93%)	518 (86.77%)
6	714	541 (75.77%)	614 (85.99%)
7	665	525 (78.95%)	444 (66.77%)
8	623	526 (84.43%)	523 (83.95%)
9	623	515 (82.66%)	523 (83.95%)
10	727	586 (80.61%)	**658 (90.51%)**
11	563	468 (83.13%)	502 (89.17%)
12	588	489 (83.16%)	492 (83.67%)
13	585	482 (82.39%)	478 (81.71%)
14	583	487 (83.53%)	509 (87.31%)
15	604	506 (83.77%)	540 (89.40%)
Mean	615.67	504.93 (82.01%)	514 (83.49%)

that satisfy the conditions of discriminator loss and classifier misclassification from the independent adversarial analysis classifier previously stated. Therefore, a subset of individuals was extracted from a set of all generated individuals (10 classes x 40 generations x 100 individuals per generation). The number of potential adversarial examples and the number of examples adversarial to model A and to model B are shown in Table 3. Based on these results on the adversarial attacks, the attack on model B is slightly more likely to be successful than on model A. It is interesting to note that model B was trained with a defensive mechanism and GLASSE is able to generate examples that on average are more successful in attacking this model. Nonetheless, all fifteen runs presented high percentages of success with examples from an Evolutionary Process guided only by the loss of the discriminator of the Conditional GAN.

We also analysed the generated examples using a t-distributed Stochastic Neighbour Embedding (t-SNE) approach for visualisation of the examples in the MNIST distribution of examples in a two-dimensional grid, based on the work of Costa et al. [7]. The t-SNE technique is able to provide a lower-dimension representation of data distribution, used to create a two-dimensional grid that spatially distributes the input images, revealing the distribution of examples according to their inner features. Figure 8 shows a visualisation map of discretized points of the t-SNE manifold where the points are represented by images of the dataset blended with the ones generated from GLASSE - original examples with 0.3 opacity and adversarial examples are with 1.0 opacity making this effect along the manifold space. In Figure 9, the scatter plot showcases all points without discretization - where zones with higher pixel intensity are zones with more examples per point. We can see that the adversarial images are within the cluster of each digit, enforcing that we are able to generate adversarial examples that

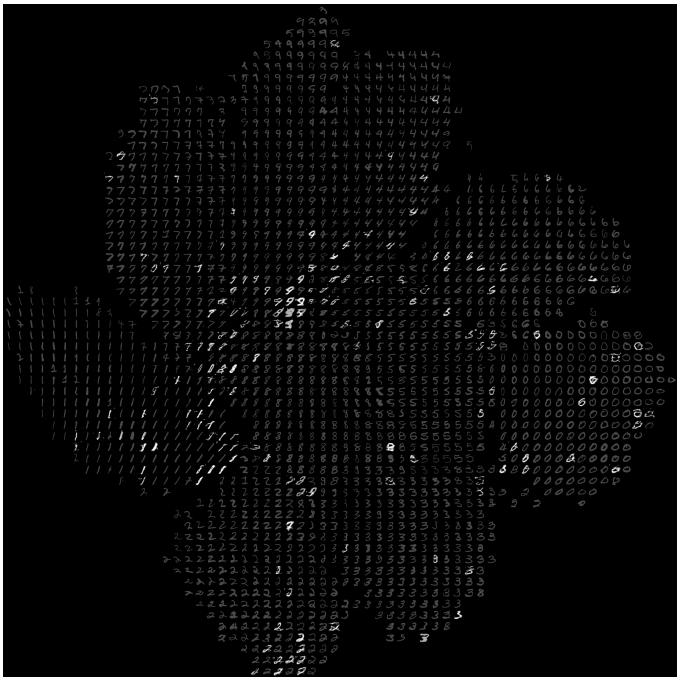

Figure 8: Blend of the MNIST dataset (0.3 opacity) and all potential adversarial examples from one run.

are similar to the digits that we aim to generate. Note that in some points, we can see in the adversarial examples parts of the original images showcasing how we are generating examples that are small variations of the original dataset but are rendered as adversarial.

Among the adversarial pool, there can be images that may not be considered pure adversarial, as the classifier and Conditional GAN models can ultimately not be completely reliable. Suppose the classifier fails in correctly predicting a label. In that case, we may have images in the pool that are only misclassifications and not proper adversarial examples - as would be with the case of noise images. This aspect of generating non-adversarial examples but errors and exploits is not new [6, 18]. On the other hand, we are conditioned by the successful training of the Conditional GAN. An error from the generator can result in the generation of numbers which do not belong to the conditioning class but are still labelled as doing so. Nevertheless, the visualization of the generated images among the MNIST dataset in Figure 8 and in Figure 9 show that there are no new clusters created to accommodate the generated images, and the generated images have a high intersection with the original dataset. Therefore, the images resulting from the failures of the models have a low occurrence in the adversarial pool.

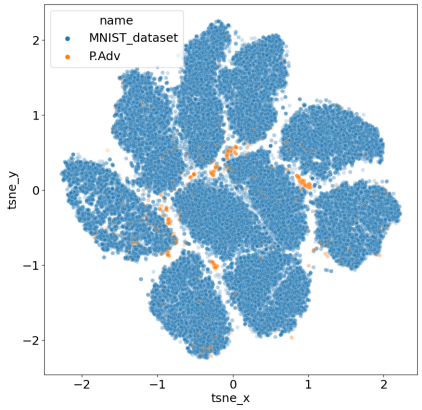

Figure 9: Scatterplot of all points from MNIST dataset and of all potential adversarial examples (P. Adv) from one run.

6 CONCLUSIONS

Adversarial examples are manipulated inputs created with the purpose of confusing a machine learning model. After a GAN is trained, the latent space of its generator maps information regarding the training dataset and the GAN. Thus, we focus on using an evolutionary approach to explore the latent space of a generator and guide the generation of adversarial examples with the discriminator loss. We build upon Fernandes et al. [9] work by switching the original goal of generating diverse images to generating adversarial images. Moreover, we explore a Conditional GAN to generate our images. In Wu et al. [18], GANs were already used to generate adversarial examples in black-box and semi-white-box settings.

In this work, we combine evolutionary computation and GANs by using a genetic algorithm to explore the latent space of a GAN to produce adversarial examples just by using the feedback of the discriminator of the GAN. Additionally, we use an independent classifier to evaluate and judge if the generated image is adversarial.

As previously mentioned, the experiments to validate the GLASSE framework revolved around the genetic algorithm to find suitable

latent vectors to generate potential adversarial examples. The genetic algorithm uses the discriminator loss of a generated latent vector towards a predetermined target loss. Moreover, the algorithm requires a classifier and a Conditional GAN previously trained with the MNIST dataset.

The experimentation with GLASSE yielded different populations of vectors through the generations of the algorithm aimed to be potential adversarial examples. Each vector is passed to the generator from the GAN to produce the images that are then submitted to an Adversarial Analysis with the independent classifier. For the conducted experiments, we consider an individual as an adversarial example if its discriminator loss is in a predetermined loss interval range and if the adversarial analysis classifier predicts it as having a different label than the one intended to be generated, with an output activation higher than 0.5.

From the experiments conducted with the GLASSE framework and the MNIST dataset, we found that the best individual in a population quickly reached our desired discriminator loss of 0.5. The experiments with class 0 evaluated by the classifier doing the adversarial analysis show how the mean number of misclassifications by the classifier increases per generation. Although there are many misclassifications, the number of adversarial examples is restricted by their discriminator loss. Still, the GLASSE framework was able to generate examples that accomplished our two objectives for all classes, and the potential adversaries resulted in successful attacks on two networks used in Carlini et al. [3]. Moreover, an exploration using a t-SNE visualization map and scatter plot of the adversarial pool and original dataset shows the proximity between the generated and original examples, highlighting the small perturbations that result in adversarial examples. From the adversarial pool we observed that not all instances were adversarial. Among the pool, there were images of noise and of numbers other than the one set to be generated by the Conditional GAN. The former is a consequence of the limitations of the classifier, and the latter reveals a failure from the generator. However, those are low occurrences as it can be seen with the t-SNE visualization map and scatter plot.

Future work will expand on the obtained results by testing the GLASSE framework with different datasets, exploring alternative fitness functions with more information to generate suitable adversarial examples, and studying evolutionary approaches to explore the latent space more efficiently to reduce the number of examples generated and be more consistent. We are also pursuing a comparison of the GLASSE performance to successfully generate attacks and efficiency with other methods of generating adversarial images. A broader avenue is also open based on these results, the latent space exploration of a generative model can be used to unveil problems in the training of GANs, on the generalisation power of classifiers and robustness of both the GAN and target classifiers trained with the same dataset.

ACKNOWLEDGEMENTS

This work is partially founded by Project "Agenda Mobilizadora Sines Nexus". ref. No. 7113, supported by the Recovery and Resilience Plan (PRR) and by the European Funds Next Generation EU, following Notice No. 02/C05-i01/2022, Component 5 - Capitalization and Business Innovation - Mobilizing Agendas for Business Innovation and by the FCT - Foundation for Science and Technology, I.P./MCTES through national funds (PIDDAC), within the scope of CISUC R&D Unit - UIDB/00326/2020 or project code UIDP/00326/2020.

REFERENCES

[1] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. https://doi.org/10.48550/ARXIV.1802.00420

[2] Philip Bontrager, Aditi Roy, Julian Togelius, Nasir Memon, and Arun Ross. 2018. DeepMasterPrints: Generating masterprints for dictionary attacks via latent variable evolution. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems, BTAS 2018 ("2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems, BTAS 2018)*. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/BTAS.2018.8698539

[3] Nicholas Carlini and David A. Wagner. 2016. Towards Evaluating the Robustness of Neural Networks. *CoRR* abs/1608.04644 (2016). arXiv:1608.04644 http://arxiv.org/abs/1608.04644

[4] M Charity, Nasir Memon, Zehua Jiang, Abhi Sen, and Julian Togelius. 2022. Diversity and Novelty MasterPrints: Generating Multiple DeepMasterPrints for Increased User Coverage. https://doi.org/10.48550/ARXIV.2209.04909

[5] Jinyin Chen, Mengmeng Su, Shijing Shen, Hui Xiong, and Haibin Zheng. 2019. POBA-GA: Perturbation optimized black-box adversarial attacks via genetic algorithm. *Computers & Security* 85 (Aug. 2019), 89–106. https://doi.org/10.1016/j.cose.2019.04.014

[6] João Correia, Penousal Machado, and Juan Romero. 2012. Improving haar cascade classifiers through the synthesis of new training examples. 1479–1480. https://doi.org/10.1145/2330784.2331001

[7] Victor Costa, Nuno Lourenço, João Correia, and Penousal Machado. 2021. Demonstrating the Evolution of GANs Through t-SNE. In *Applications of Evolutionary Computation*, Pedro A. Castillo and Juan Luis Jiménez Laredo (Eds.). Springer International Publishing, Cham, 618–633.

[8] Xiaoyi Dong, Dongdong Chen, Jianmin Bao, Chuan Qin, Lu Yuan, Weiming Zhang, Nenghai Yu, and Dong Chen. 2020. GreedyFool: Distortion-Aware Sparse Adversarial Attack. https://doi.org/10.48550/ARXIV.2010.13773

[9] Paulo Fernandes, João Correia, and Penousal Machado. 2020. Evolutionary Latent Space Exploration of Generative Adversarial Networks. In *Applications of Evolutionary Computation: 23rd European Conference, EvoApplications 2020, Held as Part of EvoStar 2020, Seville, Spain, April 15–17, 2020, Proceedings* (Seville, Spain). Springer-Verlag, Berlin, Heidelberg, 595–609. https://doi.org/10.1007/978-3-030-43722-0_38

[10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. https://doi.org/10.48550/ARXIV.1406.2661

[11] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and Harnessing Adversarial Examples. https://doi.org/10.48550/ARXIV.1412.6572

[12] Benjamín Machín, Sergio Nesmachnow, and Jamal Toutouh. 2022. Multi-Target Evolutionary Latent Space Search of a Generative Adversarial Network for Human Face Generation. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (Boston, Massachusetts) *(GECCO '22)*. Association for Computing Machinery, New York, NY, USA, 1878–1886. https://doi.org/10.1145/3520304.3533992

[13] Benjamin Machín, Sergio Nesmachnow, and Jamal Toutouh. 2021. Evolutionary latent space search for driving human portrait generation. 1–6. https://doi.org/10.1109/LA-CCI48322.2021.9769851

[14] Mehdi Mirza and Simon Osindero. 2014. Conditional Generative Adversarial Nets. https://doi.org/10.48550/ARXIV.1411.1784

[15] Aditi Roy, Nasir Memon, Julian Togelius, and Arun Ross. 2018. Evolutionary Methods for Generating Synthetic MasterPrint Templates: Dictionary Attack in Fingerprint Recognition. In *2018 International Conference on Biometrics (ICB)*. 39–46. https://doi.org/10.1109/ICB2018.2018.00017

[16] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. https://doi.org/10.48550/ARXIV.1312.6199

[17] Chenwang Wu, Wenjian Luo, Nan Zhou, Peilan Xu, and Tao Zhu. 2021. Genetic Algorithm with Multiple Fitness Functions for Generating Adversarial Examples. In *2021 IEEE Congress on Evolutionary Computation (CEC)*. 1792–1799. https://doi.org/10.1109/CEC45853.2021.9504790

[18] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. 2018. Generating Adversarial Examples with Adversarial Networks. https://doi.org/10.48550/ARXIV.1801.02610