Are Generative Classifiers More Reliable for Medical Imaging? Insights from Adversarial and Non-Adversarial Perturbations

Rosalie Millner, Emilio Picard, Thomas Gravier

Abstract

With the increasing volume of medical images generated by various radiological imaging techniques, the use of AI assistance can significantly enhance clinical applications. However, even a minimal error, such as a one-pixel discrepancy in an image ([4]), or a change in brightness or contrast, can lead to incorrect predictions in medical image analysis [5]. Such errors could result in misclassifications, which might lead to wrong clinical decisions. This vulnerability can be seen as adversarial/non-adversarial attacks on deep learning models. In this report, we explore the extent to which a deep generative classifier and a deep discriminative classifier are robust to perturbations, using a widely used MRI image multiclass dataset. Furthermore, experiments explore how altering medical images influences classification accuracy and the robustness of different generative architectures [3]. Results demonstrate that discriminative models for medical images are susceptible to these attacks, whereas GNN can be much more robust and thus lead to less class prediction errors.

1 Introduction

In this report, we present the work we carried out based on the article “Are Generative Classifiers More Robust to Adversarial Attacks?" by Yingzhen Li, John Bradshaw and Yash Sharma, 2019 [3].

The paper explores the robustness of deep neural classifiers against adversarial attacks, and more specifically, the extent to which generative classifiers are more robust than discriminative classifiers. In recent years, it has been shown that the output of neural networks can easily be altered by adding relatively small perturbations to the input vector, which raises questions about the robustness of neural networks and their vulnerability to different types of attacks. The authors introduce generative classifiers, which learn the joint distribution of inputs and labels and generate data samples based on this distribution. In contrast, traditional discriminative classifiers model the conditional probability of the label given the input, focusing directly on classifying the data. Are also presented various detection methods to identify adversarial inputs. Experimental results show that deep generative classifiers demonstrate greater robustness compared to their discriminative counterparts.

After studying the topic, we chose to focus our work towards testing the robustness of generative classifiers in the context of medical imagery. With AI being increasingly used in medical applications, we increasingly rely on medical imaging for illness detection or diagnosis, and thus classifier outputs are becoming more and more critical. A misclassified medical image due to an attack could lead to incorrect treatment, potentially endangering patients’ health. Therefore, it is of growing importance to make sure that the models employed are reliable and safe.

At first, we decided to specifically explore the very limited scenario where only one — or only a few — pixel(s) can be modified. From an interpretative perspective, such perturbations could simulate non-intentional errors, resulting from imprecise equipment or human mistakes,

that coincidentally affect vulnerable regions of the image. However, after experimenting, the results were not as satisfying as expected and we thus decided to explore a wider range of attacks, many of which are mentioned in [3]. In addition, we then decided to explore how our models performed against attacks that are what we call “non-adversarial", corresponding to perturbations like changes in brightness or contrast, which better aligns with the kind of unintentional errors we aimed to simulate in the beginning of our research.
The goal was overall to evaluate the robustness of classifiers trained on medical data when being faced with different types of perturbations. We aim to raise concerns about the reliability of classifiers trained for medical tasks, and to point out their flaws and possible improvements, as well as to confirm whether or not our experiments show that, indeed, generative classifiers are more robust to attacks than discriminative classifiers.

For the matter, we have implemented both a discriminative and a generative classifier from scratch, and have coded various types of adversarial and non-adversarial attacks, to compare their robustness on a medical dataset. Are presented in this report the theoretical and the practical considerations of our work. We have worked together throughout most of the project, so we were able to brainstorm and do our research collectively. Emilio spent more time on the implementation of the discriminative classifier, Thomas on the generative one and Rosalie on the various attacks.

2 Theoretical framework

Let $p_D(x, y)$ be the data distribution, where $x \in \mathbb{R}^D$ represents features and y the label among C classes. According to the manifold hypothesis, we suppose the data lies on a low-dimensional manifold, simplifying the learning task.

A generative model explicitly models the joint distribution $p(x, y) = p(x|y)p(y)$.
Using Bayes’ rule, we have:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}, \quad \text{with} \quad p(x) = \sum_{c\ \in C} p(x|y_c)p(y_c).$$

The logits s_c for each class $c \in C$ are defined as:

$$s_c = \log p(x|y_c) + \log p(y_c), \quad \text{and} \quad p(y|x) = \text{softmax}_c(s_c).$$

To model $p(x, y)$, a latent variable z is introduced to simplify and capture the complex relationships between x and y. Two main factorizations are commonly used:

$$p(x, z, y) = p(z)p(y|z)p(x|z), \quad \text{or} \quad p(x, z, y) = p(z|x)p(y|z)p_D(x).$$

According to [3], the first case corresponds to the GBZ model and the second to the DBX model, both depicted in Figure 1. In the first case, z generates the class y through $p(y|z)$, followed by the data x through $p(x|z)$.
For this GBZ generative model, we aim to maximize the marginal likelihood $p_\theta(x)$, which is intractable due to the integration over z:

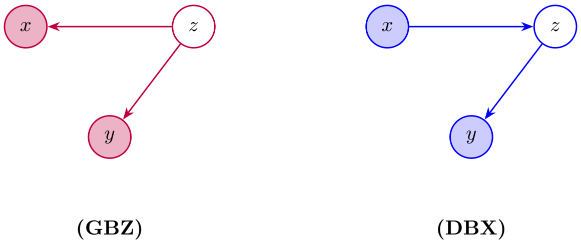

FIGURE 1. *Two graphical models, the ones that we implemented for experiments. Both discriminative and generative models here are supposed to be better than the others from [3].*

$$p_\theta(x) = \int p_\theta(x|z)p(z)\,dz.$$

To address this issue, we employ the Evidence Lower Bound (ELBO) introduced by [2] to optimize data reconstruction while regularizing the latent space using KL divergence. A classification component is added, making the total loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{class}},$$

where the losses include binary cross-entropy for reconstruction, KL divergence for regularization, and cross-entropy for classification. For more details, refer to Appendix A.

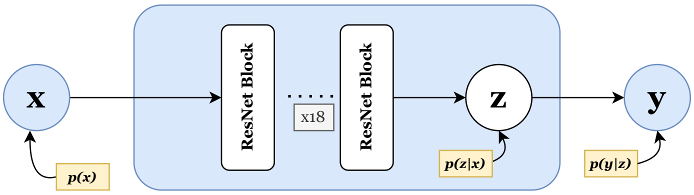

FIGURE 2. *A representation of the discriminant model that we used (DBX).*

3 Attacks

In this section, we present the various attacks we have experimented with. All the adversarial attacks we introduce here, are also used in the reference paper [3].

3.1 Adversarial Attacks

To deceive the model into making incorrect predictions, we group the adversarial attacks into 2 distinct categories:

- White-box attacks: This occurs when the adversary has full access to the model's information, including its parameters, its architecture, its gradients, and training datasets. On one side, we have the gradient-based attacks, that use the model's gradient to exploit vulnerabilities. Such attacks include the Fast Gradient Sign Method (FGSM) and the Projected Gradient Descent (PGD). Another type is optimization-based attacks that solve optimization problems, like the Carlini & Wagner (C&W) method (that we have not implemented here).

- Black-box attacks: In this case, the attacker does not have any internal information about the model, and relies for example on inferring the model's behavior through input-output queries. This category includes attacks that do not depend on the model's gradients or parameters such as the One-Pixel attack.

The **Fast Gradient Sign Method** (FGSM) is a popular type of adversarial attack that modifies the input image by adding a small perturbation in the direction of the gradient of the loss function with respect to the input. Our perturbed image is given by:

$$x_{\text{adv}} = x \; + \; \varepsilon \cdot \text{sign}\left(\nabla_x J\left(\theta, x, y\right)\right)$$

where x is the input image, y the true label, ε is the perturbation magnitude, θ is the parameters of the network and J is the loss function.
The FGSM is a powerful and simple attack that is easy to implement and does not require optimization techniques. It is also computationally efficient, as it only requires one gradient computation per image.

The **Projected Gradient Descent** (PGD) is a more iterative and robust version of the FGSM attack. After initializing the perturbation randomly, the adversarial image at step $(t + 1)$ is similar to the FGSM procedure:

$$x_{\text{adv}}^{(t+1)} = x_{\text{adv}}^{(t)} + \alpha \cdot \text{sign}\left(\nabla_{x_{\text{adv}}^{(t)}} J(\theta, \mathbf{x}_{\text{adv}}^{(t)}, y)\right)$$

After each update we ensure the adversarial image remains within the allowed perturbation space, by projecting it in it:

$$x_{\text{adv}}^{(t+1)} = \text{Proj}_{\text{B}_\epsilon}(x_{\text{adv}}^{(t+1)})$$

where $\text{B}_\epsilon = B(x_{\text{adv}}^{(t+1)}, \epsilon)$ is the open ball with radius ϵ. We repeat this for a certain number N of iterations.
PGD is thus a stronger and more sophisticated adversarial attack compared to FGSM, having multiple gradient steps to iteratively refine perturbations, while maintaining the perturbation within a specified range.

An attack in a very limited scenario is the **One-Pixel** attack, introduced by [4]. The idea is pretty straightforward: it consists of altering a single pixel from the original image, the one considered the most susceptible of confusing the classification. The attack can also be extended to altering a certain number L of pixels in the image to perturb, in order to have an attack of bigger magnitude ($L = 1$ in the one-pixel framework). For more details on the approach, see Appendix 4. Unlike the other adversarial attacks presented above, this attack is very localized. The paper [6] demonstrates how even a single pixel alteration can impact the network, as the perturbed pixel(s) propagate through the successive convolutional layers.

3.2 Non-adversarial attacks

We now explore attacks that we qualify as being non-adversarial, in the sense that they can simulate natural perturbations or non-intentional errors. More specifically, we consider changes in **brightness** of the original image as well as changes in **contrast** — in order to evaluate how these perturbations affect the classification predictions. These "attacks" do not depend on the model's internal information, and contrarily to all other attacks defined above, there is no optimization procedure. Each of these changes are simply parameterized by magnitude factor ε.

4 Experiments

Our implementation is available at `https://github.com/rosaliemillner/PGM_project`. The robustness evaluation of both generative and discriminative models against adversarial and non-adversarial attacks was conducted using the Medical MNIST dataset. It contains approximately 50,000 medical images categorized into six distinct classes: AbdomenCT, BreastMRI, ChestCT, CXR, Hand, and HeadCT.

To evaluate robustness, we employed the GBZ generative model (the architecture is described in Appendix E (4)) and a ResNet18 model as the discriminative counterpart. The ResNet18 model belongs to the DBX family of discriminative architectures. Complete hyperparameter configurations can be found in Appendix G (4) or in our accompanying GitHub repository. We selected the DBX discriminative and GBZ generative models due to their demonstrated superiority in their respective classes, as shown in paper [3]. Specifically, GBZ outperforms other generative models, while DBX outperforms alternative discriminative architectures.

For training, we followed the following protocol: Both models were trained for 15 epochs on an NVIDIA RTX 4050 Laptop GPU. The discriminative model was initialized with weights pretrained on *ImageNetV1*, and fine-tuned for 15 epochs. During fine-tuning, the first and final layers were set to have learnable parameters, while the intermediate layers remained frozen. The training process for 15 epochs required approximately 9 minutes. Figure 3 demonstrates a very good and coherent training process for the GBZ generative model.

To assess robustness, we evaluated the models on both clean and perturbed test inputs using accuracy score. The test set comprises $5,896$ images (10% of the dataset), and we used a 5-fold cross-validation split to ensure statistical reliability. The reported results are averaged across these splits. Gradient-based attacks such as the Fast Gradient Sign Method (FGSM) [1] were tested under white-box conditions, with various values of ε (ranging from 0.01 to 0.1, following [3]). Accuracy curves corresponding to these tests are presented in Figure 7 in Appendix **??**. The results are shown in Table **??** below. The results marked in bold indicate the best values between the two models.

We further evaluated robustness using adversarial Projected Gradient Descent (PGD) attack with $\varepsilon = 0.1$. The computational time for these evaluations was approximately 1 hour for each model.

Furthermore, we evaluated the robustness of our model when faced with the one-pixel attack, inspired by its relevance to simulating non-intentional medical anomalies. The results, summarized in Table **??**, demonstrate that this attack is largely ineffective against our models on this particular dataset. This surprised us at first, as it contradicts [5] for which the method was efficient at fooling a discriminative classifier. However, their experiments were based on a

Attack	ResNet18	GBZ
Clean	**0.999**	0.998
FGSM-$eps = 0.1$ (white)	0.68	**0.78**
FGSM-$eps = 0.03$ (white)	0.82	**0.91**
PGD-$eps = 0.1$ (white)	0.61	**0.88**
OnePixel (black)	0.995	**0.997**

TABLE 1. Accuracy table of both generative and discriminative models against **adversarial** attacks.

dataset where the classes were far more similar to one another (such as healthy VS sick lung). Thus, our observation can surely be attributed to the inherent separability of our dataset's classes, which are highly distinct. For a detailed analysis of the underlying latent space and its implications for adversarial robustness against small perturbations, refer to Figure 4 in Appendix **??**. Consequently, both models tested exhibit strong performance in resisting the One-pixel attacks.

We then tested the robustness of our two models against our non-adversarial perturbations: changes in brightness and in contrast. Results are shown in Table **??**.

Attack	ResNet18	GBZ
Clean	**0.999**	0.998
Brightness-$eps = 1.3$	0.83	0.83
Contrast-$eps = 4$	0.73	**0.88**

TABLE 2. Accuracy of both generative and discriminative models against **non-adversarial** attacks.

While the ResNet18 achieved higher test accuracy on clean data, its performance degraded significantly more than the GBZ model when subjected to adversarial and non-adversarial attacks. This result aligns with our hypothesis that generative models like GBZ, designed to be inherently robust, outperform their discriminative counterparts in adversarial scenarios. Furthermore, the ResNet18 architecture includes a bottleneck mechanism, which is theoretically known to enhance robustness against attacks such as FGSM. The GBZ model still demonstrates superior robustness, highlighting the resilience of generative models over discriminative ones. These observations support the conclusion that generative models offer a more robust solution in scenarios where adversarial attacks are prevalent, even when compared to theoretically strong discriminative architectures like ResNet18.

Conclusion

In this work, we compared the robustness of generative and discriminative classifiers against both adversarial attacks and non-adversarial perturbations, focusing on a medical context where accurate classification is crucial. Our experiments support the conclusion that generative classifiers exhibit greater robustness to the range of perturbations we implemented. It would be interesting and valuable to experiment with datasets where the classes are less separated, as this could make the attacks (for instance the One-Pixel attack) more effective and provide a deeper and more rigorous evaluation of our models' robustness.

References

[1] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.

[2] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. 2014.

[3] Yingzhen Li, John Bradshaw, and Yash Sharma. Are generative classifiers more robust to adversarial attacks?, 2019.

[4] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, October 2019.

[5] Min-Jen Tsai, Ping-Yi Lin, and Ming-En Lee. Adversarial attacks on medical image classification. *Cancers*, 15:4228, 08 2023.

[6] Danilo Vasconcellos Vargas and Jiawei Su. Understanding the one-pixel attack: Propagation maps and locality analysis. 2019.

Appendix

A: Details for theoretical VAE training

The Evidence Lower Bound (ELBO) employed corresponds to:

$$\log p_\theta(x) \geq \mathcal{L}(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{\mathrm{KL}}(q_\phi(z|x) \| p(z)),$$

where $q_\phi(z|x)$ is a Gaussian distribution parameterized by $\mu_\phi(x)$ and $\sigma_\phi(x)$, with reparameterization given by:

$$z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I).$$

By incorporating a classification component, the total loss becomes:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{class}},$$

where:

- The **reconstruction loss** is given by the binary cross-entropy between the original data x and its reconstruction x_{recon}:
$$\mathcal{L}_{\text{recon}} = \text{BCE}(x_{\text{recon}}, x).$$
- The **Kullback-Leibler divergence** regularizes the latent space by aligning $q_\phi(z|x)$ with the prior distribution $p(z)$:
$$\mathcal{L}_{\text{KL}} = -\frac{1}{2} \sum_{i=1}^{d} \left(1 + \log \sigma_i^2 - \mu_i^2 - \sigma_i^2\right),$$
where d is the dimension of the latent space.
- The **classification loss** is defined as the cross-entropy between the predictions y_{pred} and the labels y:
$$\mathcal{L}_{\text{class}} = \text{CrossEntropy}(y_{\text{pred}}, y).$$

In practice, the training process for the model can be summarized as follows:

- Encode x to obtain the parameters $\mu_\phi(x)$ and $\sigma_\phi(x)$;
- Sample z using the reparameterization trick;
- Reconstruct x through the decoder $p_\theta(x|z)$;
- Predict y using a classifier based on z;
- Minimize the total loss:
$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{class}}.$$

Thus, this approach ensures a balance between reconstruction quality, latent space regularization, and classification performance.

B: One-Pixel Procedure

To determine what the optimal pixel is in the One-Pixel attack procedure, the goal is to solve:

$$\min_{\epsilon(x)} f_{\text{adv}}(x + \epsilon(x)) \quad \text{where } \epsilon(x) \text{ is subject to } \quad \|\epsilon(x)\|_0 \leq L$$

with x the original image, $\epsilon(x)$ the pixel modification to apply to x, and we set f_{adv} as being the confidence of the model to correctly predict the class of x, which is considered to be the class with highest probability at input x.

Finding the optimal solution to this problem in the black-box setting relies on **differential evolution**, an algorithm inspired by the natural breeding process. It begins with a randomly initialized population of candidate pixel perturbations. Across generations of pixel candidates, new candidates are generated by combining existing ones based on their performance. Over time, this process refines the population to identify the most effective perturbations. For more details on how the differential evolution approach works, see the original paper [4].

C Evolution of Train and Validation loss and accuracy of the GBZ model during training

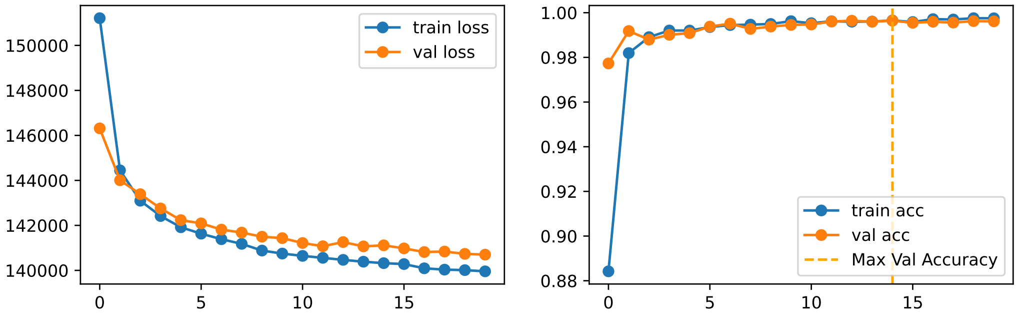

FIGURE 3. *Evolution of Train and Validation loss and accuracy of the GBZ model during training.*

D: Projection of the representations of the classes in the latent space, via PCA

Figure 4 illustrates a Principal Component Analysis (PCA) applied to the latent variable z (dimension 100) of the generative GBZ model, corresponding to each test image x, with the dimensionality reduced to two principal components for visualization. This analysis was performed to investigate the inherent separability of the data and to better understand the limited impact of the one-pixel attack on classification performance. The resulting visualization reveals that the classes are distinctly partitioned into six well-defined clusters, even after dimensionality reduction. These observations suggest that the original high-dimensional latent space provides a similarly robust separation among the six classes in the dataset, thereby offering insights into the resilience of the classifier against minor adversarial perturbations.

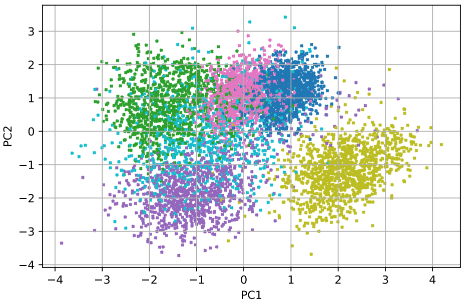

FIGURE 4. *Latent dataset vizualisation with GBZ using PCA reduction.*

E: Description of the GBZ's model architecture

The GBZ model that we decided to implement consists of three main components: an Encoder, a Decoder, and a Classifier, all interacting via a latent variable z. The Encoder uses two convolutional layers followed by fully connected layers to extract features from input images, producing the mean (μ) and log-variance ($\log \sigma$) of the latent distribution. The latent variable z is sampled from this distribution via a reparameterization trick, ensuring differentiability. The Decoder reconstructs the input image from z using a series of transposed convolutional layers, enabling the generation of images. Simultaneously, the Classifier predicts class labels based on z through two fully connected layers. The GBZ model leverages this shared latent representation to perform classification, encouraging a compact and informative latent space that captures the essential features of the data. It is trained using the reconstruction error of the image, the Kullback-Leibler divergence and the classification loss . The use of the sigmoid activation in the decoder ensures pixel values remain in the range $[0, 1]$, while the classifier provides logits for class probabilities.

F: Examples of input images and perturbations

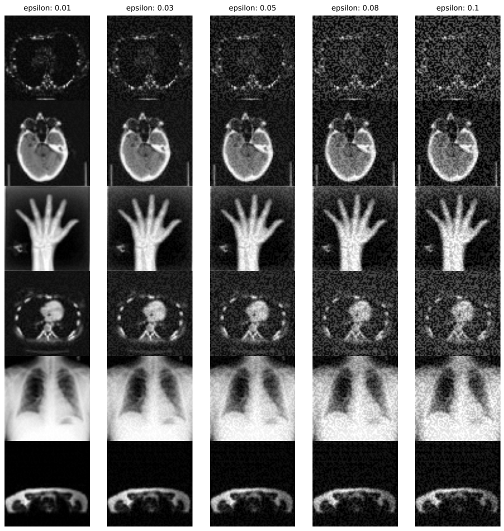

FIGURE 5. *Examples of* ***FGSM*** *attack on GBZ images for different magnitudes ε.*

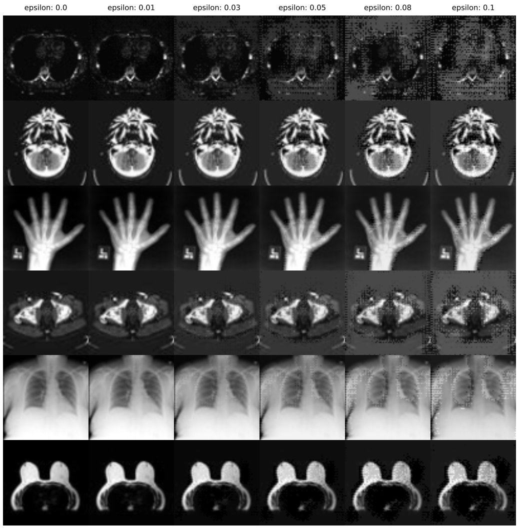

FIGURE 6. *Examples of* ***PGD*** *attack on GBZ images for different magnitudes ε.*

G: Hyperparameters Table

TABLE 3. Hyperparameters for the GBZ Model and Training Configuration

Hyperparameter	**Value**
Model-specific Parameters (GBZ)	
Latent Dimension (`latent_dim`)	100
Kernel Size (Conv2D)	4
Image Size	$1 \times 64 \times 54$ (grayscale)
Train-Test Split	80% Train-Val, 20% Test
Training Configuration (All Models)	
Number of Epochs (`num_epochs`)	20
Learning Rate (`lr`)	3×10^{-3}
Batch Size (`batch_size`)	64
Optimizer	Adam (`lr`=3×10^{-3}, `weight_decay`=1×10^{-3})
Scheduler (`StepLR`)	Step Size = 10, Gamma = 0.1

H: Evolution of accuracy for different values of epsilon in the FGSM attack

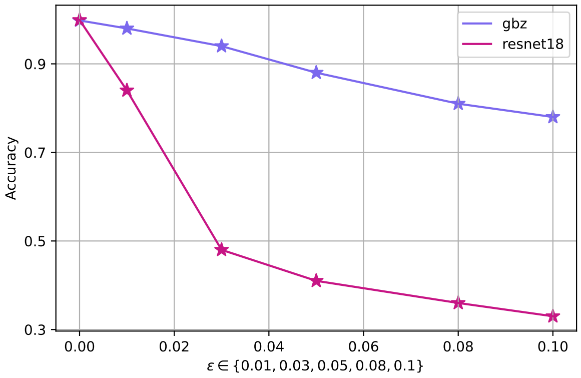

FIGURE 7. *Evolution of accuracy for different values of epsilon in the FGSM attack.*