Securing Deep Learning against Adversarial Attacks for Connected and Automated Vehicles

Final Report

by

Pierluigi Pisu, Ph.D., Clemson University
Gurcan Comert, Ph.D., Benedict College
Negash Begashaw, Ph.D., Benedict College
Chunheng Zhao, Ph.D. candidate, Clemson University

Contact information

Pierluigi Pisu, Ph.D.
4 Research Drive, Greenville, SC 29607
Clemson University
Phone: (864) 283-7227; *E-mail:* pisup@clemson.edu

September 2023

Center for Connected Multimodal Mobility (C^2M^2)

200 Lowry Hall, Clemson University
Clemson, SC 29634

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by the Center for Connected Multimodal Mobility (C^2M^2) (Tier 1 University Transportation Center) Grant, which is headquartered at Clemson University, Clemson, South Carolina, USA, from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.

Non-exclusive rights are retained by the U.S. DOT.

ACKNOWLEDGMENT

The authors would like to acknowledge the Center for Connected Multimodal Mobility (C^2M^2), which is a Tier 1 University Transportation Center, for supporting this research.

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Securing Deep Learning against Adversarial Attacks for Connected and Automated Vehicles		**5. Report Date** March, 2023 **6. Performing Organization Code**
7. Author(s) Pierluigi Pisu, Ph.D.; ORCID: 0000-0003-4266-1336 Gurcan Comert, Ph.D.; ORCID: 0000-0002-2373-5013 Chunheng Zhao, Ph.D. candidate; ORCID: 0000-0002-3121-4779 Negash Begashaw, Ph.D.; ORCID: 0000-0002-4192-3069		**8. Performing Organization Report No.**
9. Performing Organization Name and Address Department of Automotive Engineering Clemson University 4 Research Drive, Greenville, SC 29681		**10. Work Unit No.** **11. Contract or Grant No.**
12. Sponsoring Agency Name and Address Center for Connected Multimodal Mobility (C^2M^2) USDOT Tier 1 University Transportation Center Clemson University 200 Lowry Hall, Clemson Clemson, SC 29634		**13. Type of Report and Period Covered** Final Report (August 2021 – September 2023) **14. Sponsoring Agency Code**

15. Supplementary Notes

16. Abstract

Intelligent mobile robots, including autonomous agents, highly rely on the correctness of surrounding environment perception. Recently, Deep Learning-based perception models have been shown to be vulnerable to adversarial attacks through one kind of well-designed input called adversarial examples. Existing defenses include mainly adversarial training and adversarial detecting; however, they fail to solve the intrinsic issue of current deep learning models, which is the weak adversarial robustness, which partly lies in the opaque nature of the black box models. This project developed a deep ensemble network for image classification based on the fusion of discriminative features and generative models. Specifically, a causal adversarial graph is built into a generative model to model the distribution of adversarial perturbations. To improve the accuracy of generative classifiers, pre-trained object features and original images are fused together. We show that the ensemble network is robust against adversarial examples even without adversarial training (i.e., trained only with clean data), yet needs shorter training time and lower computation cost. In addition, we leverage counterfactual explanations to evaluate the model causality of the ensemble network.

17. Keywords	18. Distribution Statement
Connected and Autonomous Vehicles; Adversarial Examples; Security; Deep Learning; Classification.	No restrictions.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	19	NA

Table of Contents

LIST OF TABLES

LIST OF FIGURES

EXECUTIVE SUMMARY

Recent developments on connected and automated vehicles (CAV) show that many companies, such as Tesla, Lyft, and Waymo, are substantially investing in the development of perception modules based on deep learning algorithms. However, deep learning algorithms are susceptible to adversarial attacks aimed at modifying the input of the neural network to induce a misclassification, which may compromise vehicle decision-making and, therefore, functional safety.

The overall vision of this project is the development of a robust deep learning model which can be used for CAV resilient to adversarial attacks and, therefore, capable of satisfying more stringent system safety and performance requirements. More specifically, we leverage deep Bayes classifier and generative models to model the distributions for both clean data and adversarial data. We apply discriminative features as input to maintain the classification accuracy while introducing robustness. The main objective of this project is therefore to address the challenge by exploiting discriminative features and generative modeling to achieve higher resilience to particular type of cyber-attack known as adversarial attacks. To achieve such a goal, it is required to build intrinsic robustness into the deep learning model and not use any specific type of adversarial examples in the training, so that the robust model can be resistant to unseen adversarial examples during the testing. The training dataset should not be enlarged by simply injecting adversarial examples, which can avoid huge computation costs while deploying. The main activities for this project are summarized as follows:

- Create a causal graph that incorporates certain latent variables to build relations between different causes for the formation of adversarial inputs.
- Leverage Bayes' rule and the causal graph to build a generative classifier. The classifier needs to fuse discriminative features from pre-trained discriminative classifiers as inputs, which can provide reasonable classification accuracy.
- Use counterfactual metrics to evaluate the model causality.

In total, this project aims to develop a robust deep learning model for autonomous driving perception systems that will be able to give proper perception results even in corrupted conditions, when malicious sensor data (i.e., adversarial examples) is injected. Given the foreseeable future in which autonomous driving technology is expected to enter the market, the proposed research addresses the problem of improving the resilience of autonomous vehicles to the possibility of cyber-attacks aimed at impairing or affecting the perception results by fooling deep learning models with adversarial examples.

The main results of this project include:

- A bottom-up discriminative-generative ensemble model for image classification is developed, which leverages both generative and discriminative models with built-in adversarial causal relationships. A causal graph with latent variables is created to build Bayes-based generative classifier. The inputs consist of both original inputs and discriminative features.
- The proposed ensemble model not only shows better classification accuracy against adversarial examples but also shows better model causality when using adversarial examples as counterfactual metrics, compared with baseline models.

CHAPTER 1

Introduction

In recent years, deep learning (DL) has made significant progress in many fields like robotics, autonomous driving, and human-machine interaction [1], [2], [3] for the use of environment recognition or perception. For conventional machine learning (ML) algorithms, it is challenging to extract well-represented features due to limitations, such as the curse of dimensionality. DL can deal with massive high-dimensional data and solve the problem of feature representation through deep neural networks (DNN) by building multiple simple features (i.e., neurons) to learn a sophisticated concept. DNN, especially convolutional neural network (CNN) based models, are widely used for the task of sensing the surrounding environment. However, recent studies find that DNN is vulnerable to adversarial attacks through well-designed input samples [4], [5], [6]. Designing an input in a specific way to get the wrong classification result from the model is called an adversarial attack, and these kinds of modified and misclassified inputs are called adversarial examples [4]. An adversarial example is usually modified slightly so humans don't misclassify it. It's challenging for humans to recognize adversarial examples, thus, the adversarial attack is a non-trivial threat to many DNN-based applications. DNN-based environment perception, as the first stage in the robot navigation pipeline, should be accurate and robust enough against external perturbations and noises. A misclassification of robots' or vehicles' surrounding environment can compromise their decision-making, traveling trajectories, and, therefore, functional safety and even passenger safety in autonomous vehicles. To secure DL in safety-crucial applications, DL models must be robust and built to be inherently resistant to adversarial attacks. More specifically, the model robustness in this project is defined as the ability to maintain good accuracy against adversarial perturbations.

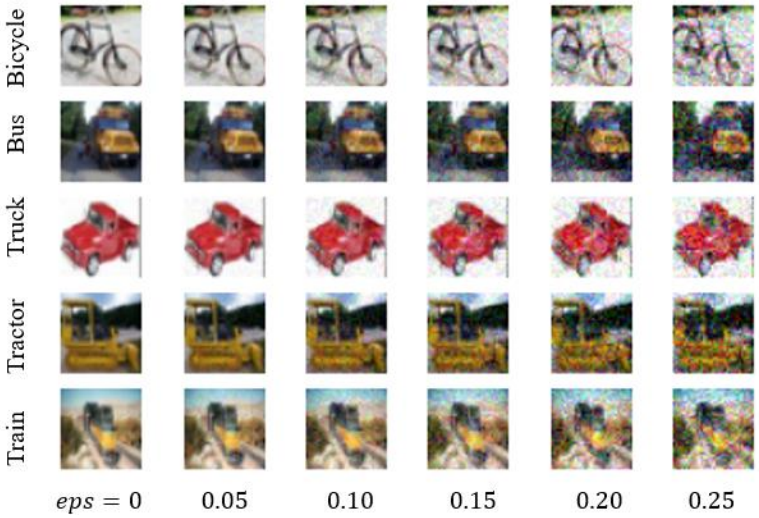

Figure 1: Projected Gradient Decent (PGD) attack examples on CIFAR-100 dataset. From top to bottom are 5 classes, including bicycle, bus, pickup truck, tractor, and train. ϵ (eps) controls the perturbation size.

In this project, we develop a generalized deep neural network architecture for image classification, an ensemble network consisting of discriminative features and generative models. Discriminative classifiers achieve higher accuracy on large-scale image classification datasets (e.g., ImageNet), while generative classifiers can be more robust to adversarial examples [7].

Our research finds that a combination of the two models can leverage their respective advantages to build a robust classifier with good accuracy. A causal graph-guided deep latent model is adapted in the generative classifier, which can model the distribution of adversarial perturbations. More specifically, adversarial perturbation is modeled as one of the latent variables, and the classification probability is estimated using Bayes' rule in the generative classifier.

In this project, the main contributions are summarized as

- We introduce a bottom-up discriminative-generative ensemble model for image classification, which leverages both generative and discriminative models with built-in adversarial causal relationships.
- We show that the ensemble network can be resistant to different types of adversarial attacks with different strengths, and the attack success rates decrease significantly compared with the baseline model. The accuracy of the clean dataset is not affected much.
- Using adversarial examples as counterfactual metrics shows that the proposed ensemble model has better model causality than the baseline models.

The remainder of this report is organized as follows. Chapter 2 provides a literature review of the adversarial attacks and defenses in autonomous vehicles. Chapter 3 discussed the proposed neural network robustification approach with generative models and causal graphs. Chapter 4 presents the experiments on CIFAR-10 and CIFAR-100 dataset. Lastly, Chapter 5 provides concluding remarks and future works.

CHAPTER 2

Literature Review

2.1 Adversarial Attacks on Images

There are several different types of algorithms for generating adversarial images. Szegedy et al. first used an L-BFGS method to solve the generation of adversarial image examples [4]. However, L-BFGS attack is time-consuming due to the use of an expensive linear search method. Goodfellow et al. proposed a fast approach called fast gradient sign method (FGSM) to generate adversarial examples [5]. However, FGSM is designed primarily to be fast instead of producing very close adversarial examples. Moosavidezfooli et al. proposed an efficient method called Deepfool that produces closer adversarial examples than the L-BFGS approach [8]. Carlini and Wagner also proposed a much more effective attack compared with FGSM attack [9]. The proposed attack is successful on defensive distilled neural networks. In addition, projected gradient descent (PGD) attack is another strong iterative adversarial attack which is a multi-step variant of FGSM [10]. Yuan et al. [11] suggested an adaptive adversarial attack that may provide a 3-6-fold speedup compared to contemporary iterative methods. Jia et al. [12] examined adversarial attacks against object detection and moving object tracking systems by attacking just three frames on autonomous cars' onboard sensors.

It has been discovered that popular scene segmentation algorithms based on deep neural networks (DNN) are vulnerable to adversarial attacks. Specifically, [13] demonstrates an iterative projected gradient-based attack approach that may mislead multiple DNN-based segmentation models with a much greater attacking success rate and significantly fewer adversarial perturbations. [14] create a stereo-regularizer to train the model on the implicit connection between images and define the loss function's local smoothness.

2.2 Defenses for Adversarial Attacks on Images

Adversarial detecting is a typical reactive approach to detect adversarial examples. Roth et al. proposed a statistical test for detecting adversarial examples [15]. Statistics leverage log-odds and exploit certain anomalies that adversarial attacks introduce. Metzen et al. empirically showed that adversarial examples could be detected surprisingly well using a detector subnetwork attached to the main classification network [16].

On the other hand, adversarial training is one of the few proactive techniques which can defeat strong attacks through regularizing deep models by encouraging the neural network to classify both clean examples and perturbed ones correctly [5]. The idea is to include adversarial examples in the training stage to make the network more robust. With adversarial training, the error rate fell to 17.9% from 89.4% on adversarial examples based on FGSM on MNIST [5]. Kurakin et al. studied how to increase robustness to adversarial examples of large models (Inception v3) trained on a large dataset (ImageNet) [17]. However, it has also been shown that injecting adversarial examples into the training set would decrease the accuracy of the clean dataset, and adversarial training is expensive in training time due to the construction of sophisticated adversarial examples. Shafahi et al. presented a "free" adversarial training algorithm which can eliminate the overhead cost of generating adversarial examples by recycling the gradient information computed when updating model parameters [18].

[19] conducted a detailed analysis of the comparative robustness of RGB images and LiDAR channel-based deep fusion systems. Some of the findings are that the sensor fusion models are more resistant to adversarial single-channel perturbation attacks than single-channel models

before and after adversarial training, highlighting the importance of fusion in enhancing robustness. Nonetheless, adversarial training with perturbations to the whole input often overfits the attack and fares worse than fusion models before adversarial training.

Apart from these techniques, recently, explainable artificial intelligence (XAI) also draws a lot of interest in the research community, and it also provides an emerging area which is defeating adversarial attacks using model explainability. It has been shown that better adversarial robustness can be achieved by building a more explainable model [27]. Ross et al. proposed a new training method with input gradient regularization to improve adversarial robustness [28]. In addition to these feature relevance techniques in XAI, some visualization explanation techniques can also be used as countermeasures against adversarial attacks. Saliency map, as a widely used technique to visualize features, including Smooth-Grad, Grad-CAM, Grad-CAM++, etc., has been adopted to either detect adversarial examples as post-hoc explanations or improve adversarial robustness as regularization terms [29-32].

The limitations of the existing countermeasures can be summarized as follows: first, current defense mechanisms focusing on detecting adversarial examples do not consider the computational burden and have not been tested in real-time scenarios [33]. Therefore, they may not be applicable in autonomous driving scenarios. Even those who have been already tested in autonomous driving models have either a low success rate or a high false-positive rate [34]. Second, current robustifying approaches like adversarial training don't build model robustness intrinsically and cannot solve the basis problem of deep learning models which is lacking interpretability. However, even XAI-based approaches with regularization may harm the model performance on clean dataset (i.e., similar to adversarial training). Therefore, in this project, we aim to robustify a neural network model by introducing a new architecture to build a more adversarially robust and more interpretable neural network model, without compromising the performance on clean data.

CHAPTER 3

Research Approach

Deep discriminative classifiers achieve great success in many classification tasks by modeling decision boundaries between different classes. It learns what features in the input contribute most to distinguishing between the various classes. In terms of defenses against adversarial examples for deep discriminative classifiers, adversarial training as one of the strongest defense mechanisms has shown improvements in the robustness of deep discriminative models by involving adversarial examples in the model training [5]. However, as many different types of adversarial examples are shown in Chapter 2, it is unrealistic to involve all types of adversarial examples in the training phase due to the high training time cost. In addition, the adversarially trained model is only robust against the types of adversarial examples used in training, which means the transferability of adversarial training cannot be guaranteed. Therefore, the goal of the proposed approach is to avoid using any specific types of adversarial examples in training; instead, to model the formation of universal adversarial perturbations by using the generative model with latent variables.

Different from discriminative classifiers, generative classifiers try to model the actual distribution of each class, which means it models how one specific class generates the input data. Therefore, a generative model could be more robust to adversarial examples as it knows what adversarial inputs look like. On the other hand, the discriminative model is vulnerable to adversarial examples because adversarial examples are designed against discriminative models to set outliers for decision boundaries to confuse the classifier. This gives the underlying theoretical support that generative classifiers could possibly be used to defeat adversarial examples, as it's more challenging to shift feature distribution than create outliers. Zhang et al. improve the generative classifier robustness via modeling the adversarial perturbation from a causal view, but the results on MNIST and CIFAR-binary require test time fine-tuning [20]. Besides, [20] couldn't obtain results on the full image dataset as they report VAE-based generative classifiers are less satisfactory for classifying clean CIFAR-10 images (< 50% clean test accuracy).

In our research, a bottom-up discriminative-generative ensemble model is developed to combine the discriminative features with a generative classifier. Discriminative features are used to ensure high classification accuracy, while the generative model is used to model the distribution of adversarial inputs to provide adversarial robustness. We further expand the work in [20] and make our technique succeed on full CIFAR-10 and CIFAR-100 datasets to show that the proposed model can be used towards more complex image classification tasks, which can contribute potentially to the autonomous vehicle and robot's perception system.

3.1 Overall Ensemble Model

The overall ensemble model architecture is shown in Figure 2. The architecture contains a bottom-level pre-trained discriminative feature extraction network and a top-level generative classification network. As we intend to generalize the proposed ensemble model for various discriminative classifiers or relative models, the bottom-level discriminative network can be any pre-trained CNN model (e.g., VGG, ResNet, etc.) depending on the actual classification task or users' preference. The pre-trained layers will be kept frozen during the model training process.

As shown in Figure 2, the top-level generative classifier takes both features extracted from the bottom-level pre-trained CNN and the original image as inputs. We consider fusing both the

original image and features because of the characteristic of a generative model to regenerate its inputs. The generative classifier should be trained to learn distributions not only from features but also from original data, which contributes to modeling adversarial images, not just adversarial features. The feature extractor (i.e., blue in Figure 2) is pre-trained for a generalized solution, and the design of the generative classifier (i.e., green in Figure 2) will be presented in 3.2 and 3.3.

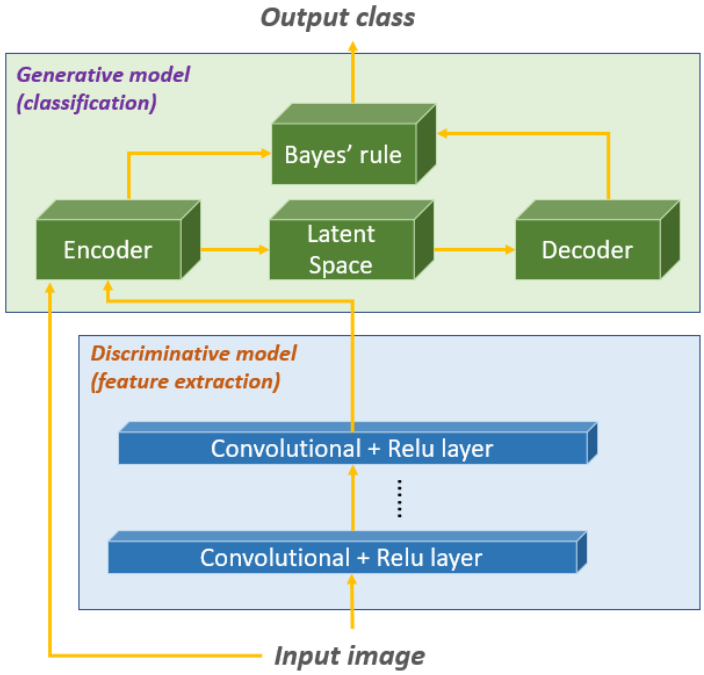

Figure 2: Bottom-up discriminative generative architecture. The ensemble model consists of two parts which are a standard features extractor and a generative classifier.

3.2 Causal Graph with Latent Variables

To build the top-level generative classifier with latent variables, a causal graph should be created first to model the relationship between inputs, outputs, and latent variables, as shown in Figure 3. By leveraging causal reasoning, DNNs can be trained to learn the causal relations rather than just statistical relations between inputs and outputs to avoid overfitting and improve robustness. In this project, we define the inputs to the generative classifier as X_1 and X_2, as there are two types of inputs. X_1 represents the original image while X_2 represents the features extracted from the pre-trained CNN. Given an input image, multiple factors or causes impact the formation of the image data. Among these factors, Y is the predicted label containing the class of the object, M is a set of variables that can be changed or modified artificially (i.e., in our case, the adversarial perturbations can be applied or injected directly and artificially), and Z represents all the other factors that cannot be changed like camera positions and object materials which would affect the reflection of light.

Following this causal graph, we consider adversarial perturbations M as a specific type of noise on the input X_1 and X_2, which could lead to misclassifications of neural networks. In order to fool a DNN, the adversarial perturbations M are generated given the target labels (i.e., misclassified as an assigned target label), or true labels (i.e., misclassified as any labels except true label), the input data (X_1, X_2) and network details θ (i.e., in case of white-box attacks). On the other hand, as adversarial perturbations are generated to fool the network with incorrect labels, label Y is affected by adversarial perturbations M, input data (X_1, X_2) and network details θ. Then for

the input data generation, adversarial inputs consist of adversarial perturbations M and original inputs (X_1, X_2) which are affected by labels Y and other factors Z. To simplify the problem, we don't consider the case that other factors can change true labels.

The causal model representing the forming mechanism of input data can then be formulated as follows:

$$X_1^{adv},\ X_2^{adv} = P_1(M, X_1, Y, Z),\ P_2(M, X_2, Y, Z) \quad (1)$$

The generative model should be able to learn the causal relationship from the input data during the training phase and make the correct classifications based on its reasoning from these factors during the inference phase.

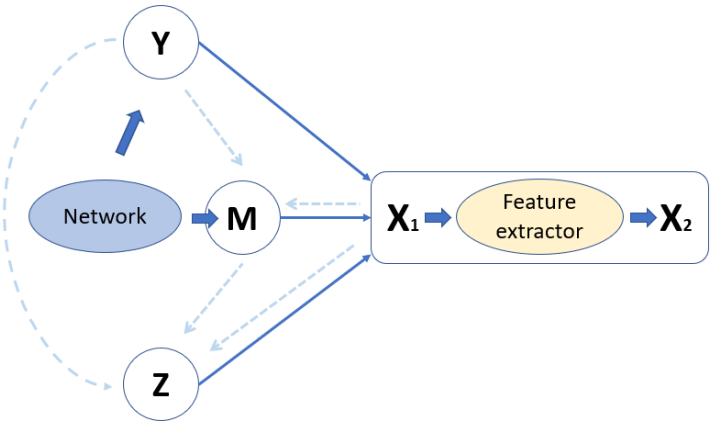

Figure 3: Bottom-up discriminative generative architecture. The ensemble model consists of two parts which are a standard features extractor and a generative classifier.

3.3 VAE-based Generative Classifier

After identifying the causal relations between inputs, outputs, and latent variables, we could leverage (1) and Bayes' rule to build the generative classifier. Given Y is the target label and X is the input, Bayes' rule can be applied to estimate the probability of y given x by $p(y|x) = p(y)p(x|y)/p(x)$. In this project, the generative classifier predicts the label y of an input x as:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} = softmax_{c=1}^{C}[log\, p\,(x, y_c)] \quad (2)$$

where C is the total number of classes and likelihood function $log\, p\,(x, y_c)$ is maximized during the training. During the prediction, the log-likelihood for each $y = c$ is computed for the distribution, then applied with softmax for the final prediction. After including latent variable m and z, (2) can be reformulated as:

$$p(y|x) = softmax_{c=1}^{C}\left[log \int p(x, y_c, z, m)\ dm\, dz\right] \quad (3)$$

In this project, as inputs X contains both x_1 and x_2, the probability $p(x, y_c, z, m)$ can then be reformulated with latent variables by

$$p(x_1, x_2, y, z, m) = p(x_1, x_2|y, z, m)p(y, z, m) \quad (4)$$

From the generative modeling process in Figure 3 (i.e., solid lines), we can then represent $p(y, z, m)$ as:

$$p(y, z, m) = p(m)p(z)p(y) \tag{5}$$

After substituting $p(x, y_c, z, m)$ in (3) with (4) and (5), the prediction probability (3) can be reformulated as:

$$p(y|x) = softmax_{c=1}^{C}\left[\int p(x_1, x_2|y, z, m)p(m)p(z)p(y)\ dm\ dz\right] \tag{6}$$

For the intractability of the marginal log-likelihood due to the intractable true posterior $p(z|\cdot)$ and $p(m|\cdot)$ for latent variables with conditional distributions, an approximate distribution [21] $q(z, m; \lambda)$ could be used to approximate the true posterior with variational parameters λ. Then the model training of maximizing log-likelihood function in (6) is equivalent to minimizing the divergence between the variational distribution and true distribution. However, this divergence is almost impossible to minimize to zero because the variational distribution is usually not sufficient enough to catch the complexity of the true posterior due to insufficient parameters. To solve the issue, Evidence Lower Bound (ELBO) can be adapted here, which is a lower bound on the log marginal probability of the data. [22] showed that minimizing the divergence is equivalent to maximizing ELBO. ELBO of the log-likelihood in (6) can be derived using variational posterior $q(z, m; \lambda)$ and Jensen's inequality as follows:

$$\begin{aligned}
log\, p\,(x, y) &= log \int p(x_1, x_2|y, z, m)p(m)p(z)p(y)\ dm\, dz \\
&= log \int p(x_1, x_2|y, z, m)p(m)p(z)p(y)\frac{q(z, m; \lambda)}{q(z, m; \lambda)}\ dm\, dz \\
&= log\, E_{q(z,m;\lambda)}\left[\frac{p(x_1, x_2|y, z, m)p(m)p(z)p(y)}{q(z, m; \lambda)}\right] \\
&\geq E_{q(z,m;\lambda)}\left[log\frac{p(x_1, x_2|y, z, m)p(m)p(z)p(y)}{q(z, m; \lambda)}\right]
\end{aligned}$$

Now we can design the inference network (i.e., variational posterior) according to (7) and the causal graph (Figure 3) as follows:

$$q(z, m; \lambda) = q_\delta(z, m|x_1, x_2, y) = q_{\delta_1}(z|x_1, x_2, y, m)q_{\delta_2}(m|x_1, x_2, y) \tag{8}$$

Here the variational parameters are $\delta = \{\delta_1, \delta_2\}$, where δ_1 is parameter for encoder network $q_{\delta_1}(z|x_1, x_2, y, m)$, and δ_2 is parameter for encoder network $q_{\delta_2}(m|x_1, x_2, y)$. Similar variational parameters are defined for the decoder network:

$$p_\theta(x_1, x_2, y, z, m) = p_{\theta 1}(x_1, x_2|y, z, m)p(m)p(z)p(y) \tag{9}$$

where the variational parameters are $\theta = \{\theta_1\}$ and θ_1 is parameter for decoder network $p_{\theta_1}(x_1, x_2|y, z, m)$.

Figure 4 shows the Variational auto-encoder (VAE)-based architecture for both the decoder and encoder network containing those separate neural nets. As both original images and object features are used in the causal graph, a CNN inside encoder processes the original image x_1 to align the data type with object features for better fusion results, as shown in the encoder network. And a deconvolutional neural network inside the decoder is used to process the object features to reconstruct the original image x_1 as shown in the decoder network. The encoder network is used to compute $q_\delta(z, m|x_1, x_2, y)$, where the parameters of CNN is included in q_{δ_2}. The decoder network is used to compute $p_\theta(x_1, x_2, y, z, m)$, where the parameters of de-CNN is included in p_{θ_1}.

After combining (7), (8) and (9), the training of p_θ and q_δ network on a dataset S with N samples can be done by maximizing the lower-bound function:

$$E_S = \sum_{n=1}^{N} E_{q_\delta}\left[log\frac{p(z)p(y_c)p_{\theta 1}(x_1, x_2|y, z, m)}{q_{\delta 1}(z|x_1, x_2, y, m)}\right] \tag{10}$$

In order to avoid the enlarged dataset and time-consuming issue in adversarial training, only clean data is used here for the model training, which means m is set to 0 during the training time. The prior distribution of $p(z)$ is set with $\mu = 0$ and $\sigma = 0$. The prior distribution of $p(y)$ is set according to the total classes in the dataset (e.g., 0.1 for CIFAR-10 and 0.01 for CIFAR-100). During the model inference period, m_t is not set to 0 but instead sampled from $q_{\delta_2}(m|x_1, x_2, y_c)$ and z_t is sampled from $q_{\delta_1}(z|x_1, x_2, y_c, m_t)$. The prediction probability can be obtained by:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \simeq softmax_{c=1}^{C}\left[log\sum_{k=1}^{K}\frac{p(z)p(y_c)p_{\theta 1}(x_1, x_2|y_c, z_t, m_t)}{q_{\delta 1}(z_t|x_1, x_2, y_c, m_t)}\right] \tag{11}$$

Where C is the number of classes, and K is the number of samples.

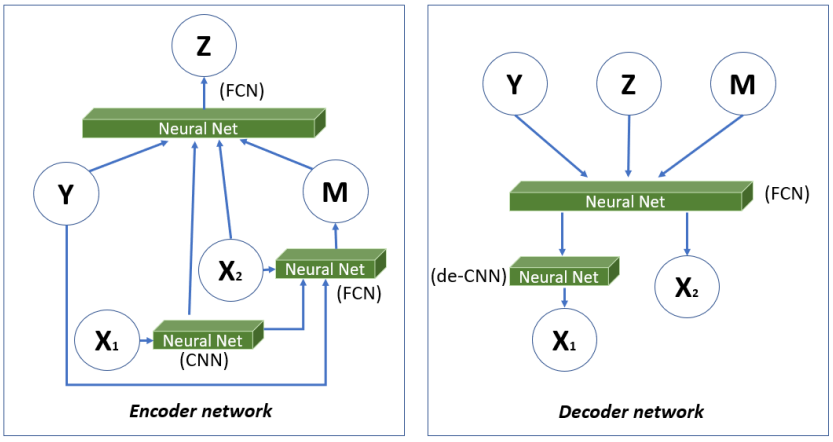

Figure 4: Generative model. Each individual neural net in the encoder and decoder estimates the independent probabilities for q and p, respectively.

The overall classifier training process can be regarded as an adjusted transfer learning process as we leverage the transfer learning concept in our ensemble model. The transfer learning technique keeps the feature extraction layers (i.e., convolutional neural networks (CNN)) but

modifies classification layers (i.e., fully connected neural networks (FCNN)) according to the actual task. Unlike traditional transfer learning, which freezes convolutional layers to keep the extracted features and then adds new fully connected layers for classification, our proposed ensemble model freezes the convolutional layers for features, but we replace the FCNN-based discriminative classifier with the above-developed Bayes-based generative classifier.

CHAPTER 4

Experiment Setup and Results

4.1 Setup

Datasets: CIFAR-10 and CIFAR-100 [23] are used in this work for preliminary results as these two datasets are widely used in terms of image classification, and they are lightweight to validate the model before testing on more sophisticated datasets. CIFAR-10 contains 10 classes with 6000 images each, divided into 5000 training and 1000 testing images per class. CIFAR-100 contains 100 classes with 600 images each, divided into 500 training and 100 testing images per class. Each image in CIFAR-10 and CIFAR-100 is 32x32x3.

Pre-trained Models: A state-of-the-art discriminative image classifier VGG-16 [24] and a generative classifier GBZCONV9 [7] are considered in this work. VGG-16 is a standard CNN with 16 convolutional and dense layers, while GBZ-CONV9 is a deep generative classifier with convolutional features as input. Those two models were selected because VGG-16 is widely used in a variety of different image classification applications and is one of the discriminative classifiers with the best accuracy. GBZ-CONV9, on the other hand, has the best performance in terms of generative classifiers against adversarial examples, which is proposed in the only paper in the field of robust generative image classifier on full CIFAR-10 dataset. Although there are several different generative models with different architectures and parameters in [7], we selected GBZ-CONV9 here as it achieves the best accuracy among all the models.

VAE Architecture: The FCN estimating $q_{\delta_1}(z|x_1, x_2, y, m)$ and $q_{\delta_2}(m|x_1, x_2, y)$ both consist of 2 hidden fullyconnected layers, each with 500 neurons and ReLU activation. The CNN to process X_1 has 3 hidden convolutional layers with filter size 5x5 and [64, 128, 256] channels. The FCN estimating $p_{\theta_1}(x_1, x_2|y, z, m)$ consists of 2 hidden fully-connected layers, each with 500 neurons and ReLU activation. The deconvolutional CNN to reconstruct X_1 has 3 hidden convolutional layers with filter size 5x5 and [128, 64, 3] channels. Training optimizer is set to Adam with learning rate 5e−5, and batch size is set to 100. The total iterations are 300.

Attacks: Two types of adversarial attacks are considered in this project, which are fast gradient sign method (FGSM) [5] and projected gradient descent (PGD) [10]. FGSM is fast enough for several real-time applications, while PGD is an iterative attack and one of the strongest. FGSM attack can be formulated as:

$$adv_x = x + \epsilon * sign\big(\nabla_x J(\theta, x, y)\big) \tag{12}$$

where adv_x is the generated adversarial image, x is the original image, ϵ is the multiplier to ensure the perturbations are small, and $J(\theta, x, y)$ is a loss function concerning the neural network parameters θ, input x and output labels y.

PGD attack is an iterative variant of FGSM attack and can be formulated as:

$$adv_x^{t+1} = \Pi_{x+S}\left(adv_x^t + \epsilon * sign\big(\nabla_x J(\theta, x, y)\big)\right) \tag{13}$$

where Π_{x+S} projects perturbations into set S which can be l_2 or l_{inf}, etc.

We evaluated ϵ (i.e., eps) from 0 to 0.2 (eps = 0 indicates no attacks) as perturbations with eps larger than 0.2 would be too large and obvious (see Figure 1).

Causality evaluation: We leverage counterfactual explanations to evaluate neural network model causality according to [25]. Proximity and speed are selected as two counterfactual

properties. In this work, we regard adversarial examples as counterfactual and evaluate the minimal perturbation size as well as minimal attack iterations (i.e., number of gradient updates) on the proposed ensemble model and baseline models.

In addition, Remove and Retrain (ROAR) and Keep and Retrain (KAR) [26] are two attribution method evaluation metrics used in this work to measure the interpretability of deep neural networks. The idea is to measure the accuracy change if some features were occluded based on the ordering assigned by the attribution method (gradient-based saliency map). For ROAR, the most important image pixels are replaced with a constant value. For KAR, instead, the least important pixels are replaced with a constant value. Then the network is retrained on the modified dataset, and the change in testing accuracy is recorded.

4.2 Comparison of Accuracy

As shown in Figure 5(a) and 5(d), VGG16-CNN is the least robust network on CIFAR-10, and even a small eps can reduce the accuracy to 20% (i.e., eps 0.05 for FGSM and 0.01 for PGD), which makes the classifier misclassify significantly. GBZ-CONV9 can increase the accuracy but is still not good enough (lower than 80% for nearly all the cases). The proposed discriminative-generative network (DGN) can significantly improve accuracy compared with the baseline VGG16-CNN network and GBZ-CONV9. The accuracy is around 90% for FGSM attacks, which makes the FGSM attack ineffective. For PGD attacks, it can achieve 80% accuracy for eps 0.05 and 50% for eps 0.1, which is a significant improvement. The difference in performance for FGSM and PGD is due to the strength of the attack, as FGSM is a quick one-step attack while PGD is an iterative and stronger attack. We can also see that the ensemble model doesn't make the clean dataset accuracy (i.e., eps = 0) decrease and even increase it a little. We assume this is because the ensemble DGN has better causality and interpretability (shown in Section IV-C), which could improve the overall deep neural network performance.

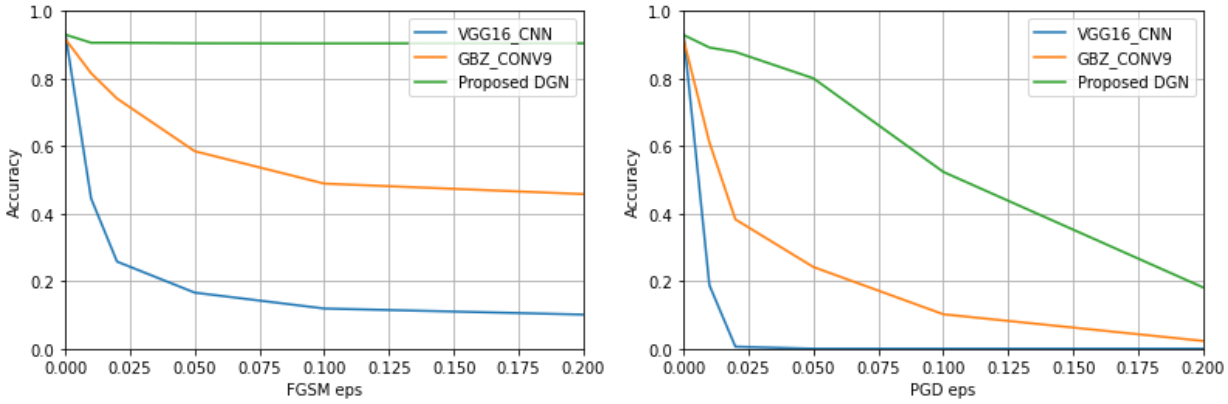

Figure 5: Classification accuracy for the adversarial examples generated by FGSM and PGD on CIFAR-10 dataset. eps control the perturbation size. The proposed ensemble could improve the accuracy against adversarial examples and even on the clean dataset.

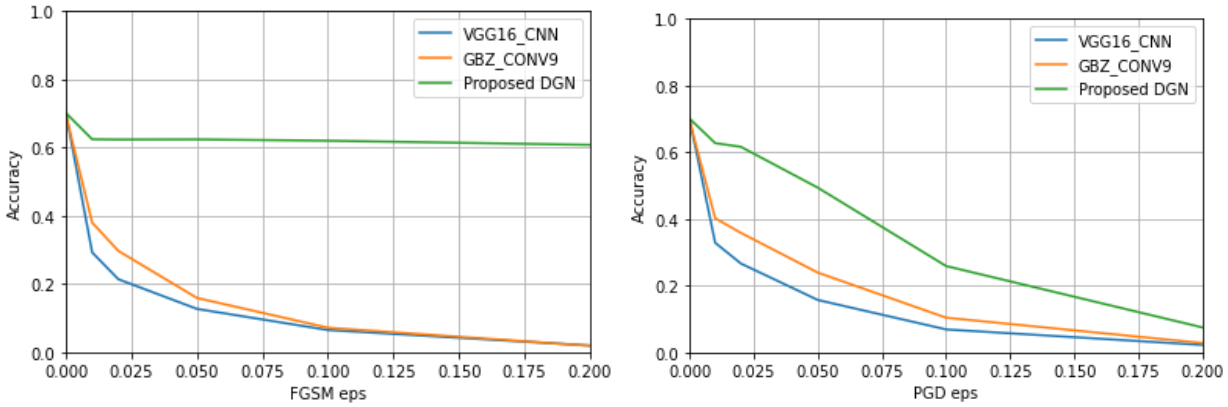

Figure 6: Classification accuracy for the adversarial examples generated by FGSM and PGD on CIFAR-100 dataset. eps control the perturbation size. The proposed ensemble could improve the accuracy against adversarial examples.

As shown in Figure 6(a) and 6(d), for CIFAR-100, VGG16-CNN is still the least robust network. And the proposed DGN ensemble can still outperform the other two baseline models even on this much larger dataset. We observe that PGD attacks against VGG16 are less effective with small eps on CIFAR-100 than CIFAR-10, and we assume this is due to the enlarged size and increased classes of the dataset.

It has been shown that it takes 3–30 times longer to form a robust network with adversarial training than forming a nonrobust equivalent [18]. In this project, although the training time is not recorded, we suppose it's quicker to train this model as the dataset is not enlarged by adversarial examples.

4.3 Evaluation of Causality

As in Section IV-A, minimal perturbation size is calculated by measuring the minimal perturbation needed to be added to the original images. A more robust network tends to need stronger adversarial attacks with larger perturbations to reduce the classification accuracy successfully. As shown in Figure 5(b) and 5(e), for FGSM attack, the VGG16- CNN is the easiest network to attack, which needs the slightest perturbation, and the proposed DGN and the GBZCONV9 require larger perturbation for the attack to be effective. For PGD attacks, there are no apparent differences in the average perturbation size between the models. We suppose this is because the perturbation for iterative attacks can accumulate after several steps, and the differences on different networks can be minor if a fixed number of iterations are used (T = 30 iterations in this case). A higher eps leads to a larger perturbation size as the eps represents the attack step size (i.e., final perturbation is the product of original perturbation and attack step size). Regarding the results on CIFAR-100 dataset, as shown in Figure 6(b) and 6(e), the same trends show for FGSM attack, but no apparent differences can be reported between the proposed DGN and GBZ-CONV9 for small eps. Differences among the three models are even more minor for PGD attacks.

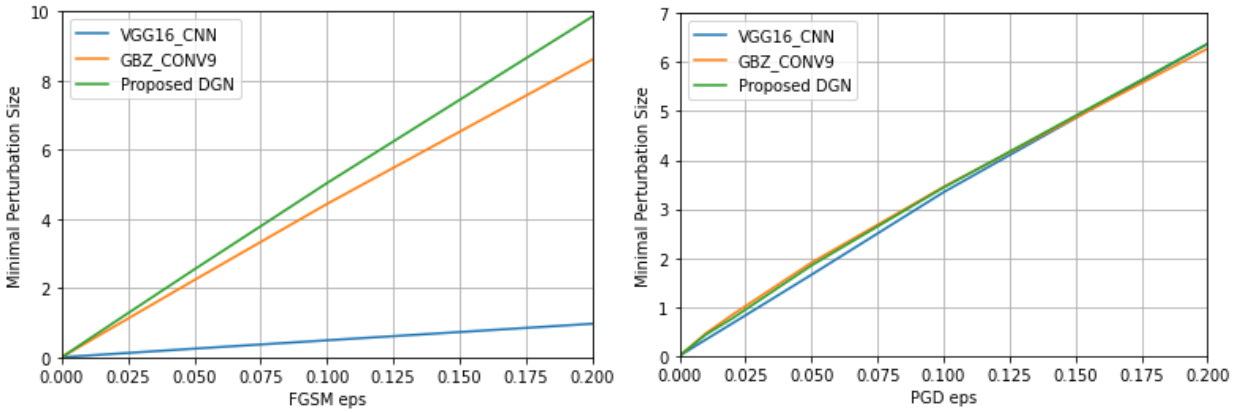

Figure 7: Average minimal perturbation size for the adversarial examples generated by FGSM and PGD on CIFAR-10 dataset. A larger average minimal perturbation size indicates a more robust model. The proposed DGN ensemble shows better results on FGSM attacks while limited performance on PGD attacks.

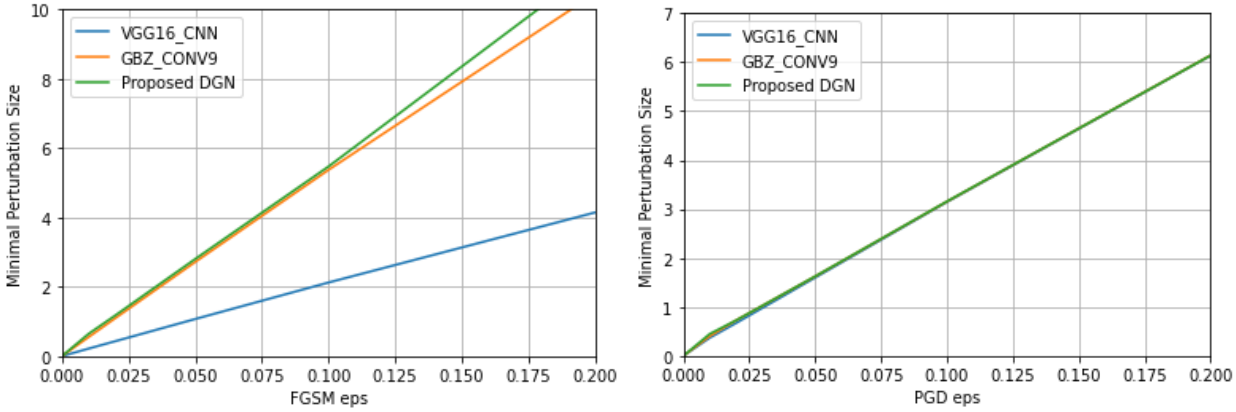

Figure 8: Average minimal perturbation size for the adversarial examples generated by FGSM and PGD on CIFAR-100 dataset. The ensemble shows a larger average minimal perturbation size on FGSM attacks while no improvement on PGD attacks compared with VGG16.

Minimal iterations are the least number of iterations needed while running the attack-generating algorithm. More iterations usually mean larger perturbation; thus, the network can be identified as more robust, and only stronger attacks can successfully attack it. As shown in Figure 5(c) and 6(c), for FGSM attacks, the VGG16-CNN only requires one iteration, which means even a one-step FGSM attack is enough to compromise it. GBZ-CONV9 performs a little better that the max minimal iterations are 5 on CIFAR-10 dataset but can also be attacked by a one-step FGSM attack on CIFAR-100 dataset. For the proposed DGN, the minimal iterations are always 30 on both CIFAR-10 and CIFAR-100 datasets, the maximum iteration number we set in the experiment. We stop the iteration at 30 even if the accuracy is not reduced to the desired values (40% in this experiment). At this point, the DGN requires more than 30 iterations for the FGSM attack to be effective. For PGD attacks, as shown in Figure 5(f) and 6(f), the proposed DGN needs more iterations for small eps values than VGG16-CNN and GBZCON9. However, for large eps values (e.g., more than 0.1), the proposed DGN has limited performance.

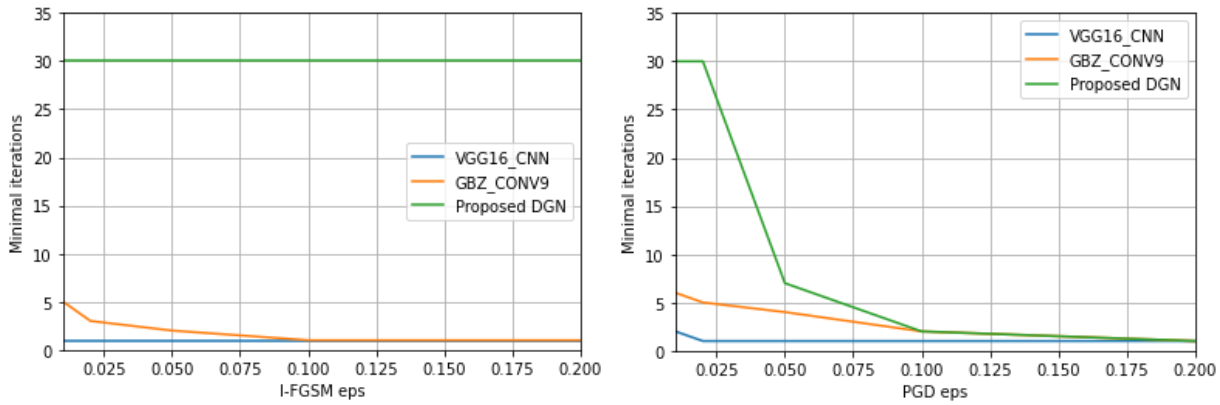

Figure 9: Minimal iterations needed for the adversarial examples generated by FGSM and PGD on CIFAR-10 dataset. Larger minimal iterations indicate a more robust model. The proposed DGN ensemble significantly outperforms baselines on FGSM and PGD attacks with small eps.

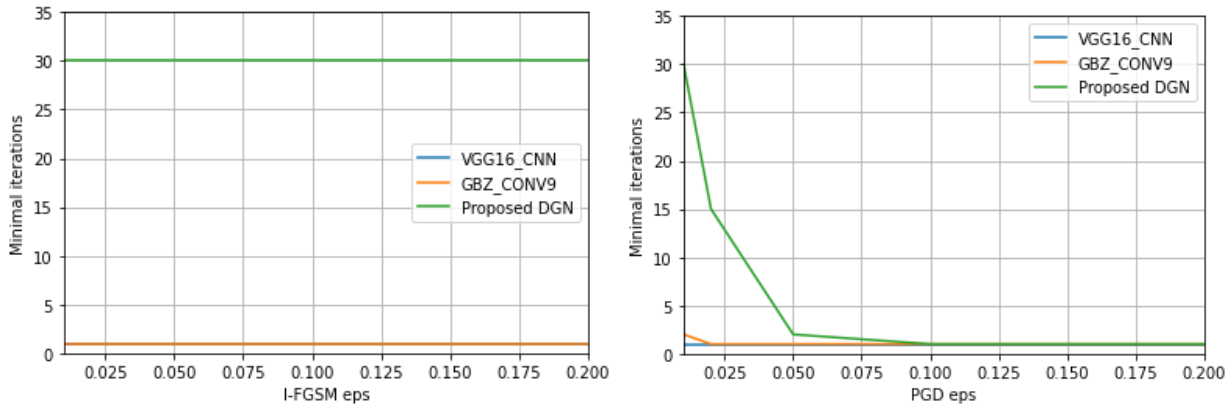

Figure 10: Minimal iterations needed for the adversarial examples generated by FGSM and PGD on CIFAR-100 dataset. Larger minimal iterations indicate a more robust model. The proposed DGN ensemble significantly outperforms baselines on FGSM attacks regardless of eps and PGD attacks with small eps.

Table I shows the results on ROAR and KAR. The higher the ROAR, the better the interpretability; the lower the KAR, the better the interpretability. A high ROAR means removing those important features can reduce the accuracy a lot, and a low KAR means removing those unimportant features cannot affect the accuracy a lot, which both indicates the network can be interpreted by the attribution method properly. As shown in Table I, the proposed DGN has the highest ROAR and the least KAR, which shows the ensemble model can be better interpreted with the same technique, indicating better interpretability.

Table 1: ROAR/KAR on CIFAR-10. Large ROAR and small KAR indicate better model interpretability, which is both achieved by the proposed ensemble model.

Network	ROAR	KAR
VGG16-CNN [24]	0.006	0.012
GBZ-CONV9 [7]	0.0005	0.0063
Proposed DGN	0.0214	0.0013

CHAPTER 5

Conclusions

This project proposes a deep ensemble model for image classification with the fusion of discriminative features and generative models. A causal graph is designed while constructing the deep Bayes classifier to model the adversarial perturbations. As the deep Bayes classifier can't achieve the state-of-the-art accuracy of discriminative classifiers, we fuse the object features extracted from pre-trained CNNs with original images as final inputs. Benefiting from this structure, the proposed method is generic and can be applied to various discriminative classifiers. The generative model can be used as an auxiliary network to be built on top of any pre-trained CNNs. Experimental results show that the proposed ensemble model achieves reduced accuracy loss against adversarial examples and gains better overall model causality and interpretability. By integrating this model into the autonomous driving perception modules, autonomous vehicles could be more robust against adversarial attacks aiming at causing misclassifications. Therefore, the technique can improve vehicle security and safety by building a resilient perception module. Future research directions would be applying the proposed robust model to object detectors and commercialized autonomous driving stacks to validate the results on more sophisticated autonomous driving models. Testing the model against physical adversarial attacks could also be further explored.

REFERENCES

[1] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixao, F. Mutz, et al., "Self-driving cars: A survey," Expert Systems with Applications, p. 113816, 2020.

[2] H. A. Pierson and M. S. Gashler, "Deep learning in robotics: a review of recent research," Advanced Robotics, vol. 31, no. 16, pp. 821 – 835, 2017.

[3] R. A. Khalil, E. Jones, M. I. Babar, T. Jan, M. H. Zafar, and T. Alhussain, "Speech emotion recognition using deep learning techniques: A review," IEEE Access, vol. 7, pp. 117 327 – 117 345, 2019.

[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," arXiv preprint arXiv:1312.6199, 2013.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv preprint arXiv:1412.6572, 2014.

[6] N. Papernot, P. McDaniel, A. Sinha, and M. Wellman, "Towards the science of security and privacy in machine learning," arXiv preprint arXiv:1611.03814, 2016.

[7] Y. Li, J. Bradshaw, and Y. Sharma, "Are generative classifiers more robust to adversarial attacks?" in International Conference on Machine Learning. PMLR, 2019, pp. 3804 – 3814.

[8] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2574 – 2582.

[9] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in 2017 ieee symposium on security and privacy (sp). IEEE, 2017, pp. 39 – 57.

[10] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," arXiv preprint arXiv:1706.06083, 2017.

[11] X. Yuan, P. He, X. Lit, and D. Wu, "Adaptive adversarial attack on scene text recognition," in IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2020, pp. 358 – 363.

[12] Y. J. Jia, Y. Lu, J. Shen, Q. A. Chen, H. Chen, Z. Zhong, and T. W. Wei, "Fooling detection alone is not enough: Adversarial attack against multiple object tracking," in International Conference on Learning Representations (ICLR' 20), 2020.

[13] X. Xu, J. Zhang, Y. Li, Y. Wang, Y. Yang, and H. T. Shen, "Adversarial attack against urban scene segmentation for autonomous vehicles,"IEEE Transactions on Industrial Informatics, vol. 17, no. 6, pp. 4117 – 4126, 2020.

[14] Q. Sun, A. A. Rao, X. Yao, B. Yu, and S. Hu, "Counteracting adversarial attacks in autonomous driving," in Proceedings of the 39th International Conference on Computer-Aided Design, 2020, pp. 1 – 7.

[15] K. Roth, Y. Kilcher, and T. Hofmann, "The odds are odd: A statistical test for detecting adversarial examples," in International Conference on Machine Learning. PMLR, 2019, pp. 5498 – 5507.

[16] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," arXiv preprint arXiv:1702.04267, 2017.

[17] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," arXiv preprint arXiv:1611.01236, 2016.

[18] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" arXiv preprint arXiv:1904.12843, 2019.

[19] S. Wang, T. Wu, A. Chakrabarti, and Y. Vorobeychik, “Adversarial robustness of deep sensor fusion models,” in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 2387 – 2396.
[20] C. Zhang, K. Zhang, and Y. Li, “A causal view on robustness of neural networks,” arXiv preprint arXiv:2005.01095, 2020.
[21] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” arXiv preprint arXiv:1312.6114, 2013.
[22] C. Zhang, J. B¨utepage, H. Kjellstr¨om, and S. Mandt, “Advances in variational inference,” IEEE transactions on pattern analysis and machine intelligence, vol. 41, no. 8, pp. 2008 – 2026, 2018.
[23] A. Krizhevsky, G. Hinton, et al., “Learning multiple layers of features from tiny images,” 2009.
[24] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.
[25] R. Moraffah, M. Karami, R. Guo, A. Raglin, and H. Liu, “Causal interpretability for machine learning-problems, methods and evaluation,” ACM SIGKDD Explorations Newsletter, vol. 22, no. 1, pp. 18 – 33, 2020.
[26] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim, “A benchmark for interpretability methods in deep neural networks,” Advances in neural information processing systems, vol. 32, 2019.
[27] N. Liu, M. Du, R. Guo, H. Liu, and X. Hu. Adversarial attacks and defenses: An interpretation perspective. arXiv preprint arXiv:2004.11488, 2020.
[28] A. Ross and F. Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018.
[29] A. Noack, I. Ahern, D. Dou, and B. Li. Training deep neural networks for interpretability and adversarial robustness. arXiv preprint arXiv:1912.03430, 2019.
[30] P. Mangla, V. Singh, and V. N. Balasubramanian. On saliency maps and adversarial robustness. arXiv preprint arXiv:2006.07828, 2020.
[31] C. Etmann, S. Lunz, P. Maass, and C.-B. Schonlieb. On the connection between adversarial robustness and saliency map interpretability. arXiv preprint arXiv:1905.04172, 2019.
[32] D. Ye, C. Chen, C. Liu, H. Wang, and S. Jiang. Detection defense against adversarial attacks with saliency map. arXiv preprint arXiv:2009.02738, 2020.
[33] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman. (2017). "PixelDefend: Leveraging generative models to understand and defend against adversarial examples." [Online]. Available: https://arxiv.org/abs/1710.10766
[34] Y. Deng, X. Zheng, T. Zhang, C. Chen, G. Lou, and M. Kim. "An Analysis of Adversarial Attacks and Defenses on Autonomous Driving Models, 2020," arXiv preprint arXiv:2002.02175.