Title

Statistical Innovations in Health and Data Security: Lung Cancer Diagnosis, Microbiome Community Detection, and Adversarial Attack Analysis

Author

Wang, Xiawei

Publication Date

2024

Peer reviewed|Thesis/dissertation

Statistical Innovations in Health and Data Security: Lung Cancer Diagnosis, Microbiome Community Detection, and Adversarial Attack Analysis

By

Xiawei Wang

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Biostatistics

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Thomas C. M. Lee

James Sharpnack

Shizhe Chen

Committee in Charge

2024

To all the experiences.

Contents

Abstract

This dissertation aims to investigate three distinct problems. Firstly, it aims to enhance lung cancer diagnosis and survival predictions through the implementation of deep learning techniques and CT imaging. Secondly, it delves into understanding the differences in distortion patterns present in adversarial images generated by various attack methods. Lastly, it explores the application of the Minimum Description Length (MDL) principle for optimal threshold determination in microbiome community detection.

Supervised by Professor Thomas Lee and Professor James Sharpnack, Chapter 2 proposes the utilization of convolutional neural networks to model the intricate relationship between the risk of lung cancer and the morphology of the lungs depicted in CT images. Introducing a mini-batched loss extending the Cox proportional hazards model, this approach accommodates the non-convexity induced by neural networks, enabling training on large datasets. The combination of mini-batched loss and binary cross-entropy facilitates the prediction of both lung cancer occurrence and the risk of mortality. Results from simulations and real data experiments highlight the potential of this method to advance lung cancer diagnosis and treatment.

Supervised by Professor Thomas Lee, Chapter 3 discusses the application of the MDL principle in microbiome data analysis, particularly focusing on community detection methods. Addressing the challenge of subjective threshold selection in correlation-based techniques, MDL is employed to identify the optimal community structure by minimizing the subjectivity in choosing a cut-off for correlation strength. The chapter provides a detailed derivation of the MDL principle, discusses its consistency in threshold selection, and validates its effectiveness through simulations. A real data experiment involving microbiome data from the Great Lakes offers practical insights into the application of MDL in a real-world context.

Supervised by Professor Thomas Lee, Professor Yao Li, and Professor Cho-Jui Hsieh, Chapter 4 explores the vulnerability of deep neural networks to adversarial examples. Focusing on three common attack families – gradient-based, score-based, and decision-based – the research aims to recognize distinct types of adversarial examples. By identifying the information possessed by attackers, effective defense strategies can be developed. The study demonstrates that adversarial

images from different attack families can be successfully identified with a simple model. Experiments on CIFAR10 and Tiny ImageNet reveal differences in distortion patterns between various attack types for both L_2 and L_∞ norms.

Acknowledgments

The completion of this dissertation marks the culmination of a journey filled with challenges, growth, and invaluable support from numerous individuals, without whom this endeavor would not have been possible.

I am immensely grateful to my advisors, Professors Thomas Lee and James Sharpnack, whose unwavering guidance, encouragement, and support have been indispensable throughout my doctoral studies. They have not only served as outstanding mentors, offering invaluable insights and constructive feedback at every step of the research process, but have also inspired and motivated me to delve deeper into my work, developing a robust foundation in my field of study. In addition to their professional guidance, I am thankful for Thomas and James's patience, kindness, and understanding during the inevitable challenges that arise in the research process. Their support and encouragement have been a strength, especially during difficult times.

I extend my heartfelt gratitude to my collaborators, Professors Yao Li and Cho-Jui Hsieh, for their invaluable contributions to this dissertation. Collaborating with Yao and Cho-Jui on my second project has been an enriching experience. Without their profound knowledge and keen intuition in adversarial attacks, it would have been much more difficult for me to navigate through this area. Additionally, I have gained tremendous insight from Yao, who serves as an exemplary role model for me as a graduate student in statistics.

I am genuinely grateful to all the professors for serving on my qualifying exam and dissertation committees: Professors Shizhe Chen, Chris Drake, Lihong Qi, Krishnakumar Balasubramanian, and Miriam Nuno. I am very grateful for their kind assistance, encouragement and valuable comments. Furthermore, I want to express my sincere appreciation to Professor Zhengjun Zhang from the University of Wisconsin-Madison for his belief in my potential and unwavering encouragement to pursue a Ph.D., a path I had not previously considered. His wisdom and support have been invaluable, and I have greatly benefited from our enlightening conversations.

I also want to extend my gratitude to my physical therapists, Ahmed Altashi and Williams Singh, for their indispensable assistance throughout my Ph.D. journey.

Additionally, I am deeply grateful to the incredible peers I had the pleasure of meeting in Madison and Davis, including Shouwei Hui, Ye He, Qianhui Wan, Shuting Liao, Yuxuan Zhang, Jingwei

Xiong, Xiaohan Hu, Jiaxiang Li, Tesi Xiao, Xi Chen, Zhenyang Zhang, and Haolin Chen. Thank you for the wonderful times we shared.

Moreover, I want to express my special thanks to Chengyang Wang for all the experiences we have shared together. Also I want to thank my three beloved kitties, Chouzhu, Pipi, and Liuliu, for their constant companionship and trust. Their unspoken love, reflected in their green eyes and gentle purring at night, has been a source of comfort and healing. Thank you all for being there for me unconditionally.

Finally, I am deeply grateful to my beloved family—my parents, Jing and Yili, and my younger brother, Ningyi and my grandparents Huiqin and Zhongyu. Their boundless love and belief in me have been a guiding light in my life. The consistent encouragement and support have been the bedrock upon which I've earned these remarkable achievements.

CHAPTER 1

Overview

This dissertation comprises three distinct studies, each contributing to the advancement of medical diagnosis, deep learning robustness, and microbiome data analysis.

1.1. Enhancing Lung Cancer Diagnosis and Survival Prediction

Lung cancer is a major cause of cancer-related deaths, and early diagnosis and treatment are crucial for improving patients' survival outcomes. Our research introduces a novel approach employing convolutional neural networks (CNNs) to model the intricate relationship between lung cancer risk and lung morphology, as revealed in CT images. We apply a mini-batched loss that extends the Cox proportional hazards model to handle the non-convexity induced by neural networks, which also enables the training of large data sets. Additionally, we propose to combine mini-batched loss and binary cross-entropy to predict both lung cancer occurrence and the risk of mortality. Our simulations and experiments on the National Lung Screening Trial dataset demonstrate the effectiveness of our approach in improving lung cancer diagnosis and prognosis, showcasing high AUC and C-index scores.

1.2. Statistically Consistent Microbiome Community Detection

Microbiome data obtained through high-throughput sequencing technologies contains information about the microbial community composition, diversity, and relative abundance within a given environment. In this study, we discuss the application of the Minimum Description Length (MDL) principle in microbiome data analysis, specifically focusing on community detection methods. It addresses the challenge of subjective threshold selection in correlation-based techniques for microbiome community detection. The MDL principle is used to identify the optimal community structure by minimizing the subjective nature of choosing a cut-off to determine correlation strength. We provide a detailed derivation of the MDL principle and discuss its consistency in choosing thresholds for community detection methods. Additionally, the effectiveness of MDL in selecting

optimal thresholds is validated through simulations. A real data experiment involving microbiome data from the Great Lakes is also presented to offer practical insights into the application of the MDL principle in a real-world context.

1.3. Understanding Distortion Patterns of Adversarial Attacks

Deep neural networks have achieved remarkable performance in many areas, including image-related classification tasks. However, various studies have shown that they are vulnerable to adversarial examples – images carefully crafted to fool well-trained deep neural networks by introducing imperceptible perturbations to the original images. To better understand the inherent characteristics of adversarial attacks, we study the features of three common attack families: gradient-based, score-based, and decision-based. Our primary objective is to recognize distinct types of adversarial examples, as identifying the type of information possessed by the attacker can aid in developing effective defense strategies. In this study, we demonstrate that with a simple model, adversarial images from different attack families can be successfully identified. To further investigate the reason behind the observations, we conduct carefully designed experiments to study the perturbation patterns of different attacks. Experimental results on CIFAR10 and Tiny ImageNet validate the differences in distortion patterns between various attack types for both L_2 and L_∞ norm.

CHAPTER 2

Enhancing Lung Cancer Diagnosis and Survival Prediction

2.1. Introduction

Lung cancer is one of the most common causes of cancer-related deaths for both men and women worldwide. Early diagnosis and treatment are crucial for improving patients' survival rates [**Alberg and Samet, 2003**, **Spiro and Silvestri, 2005**]. Survival analysis, a branch of statistics that has been widely used in public health research, provides valuable insights into the impact of different conditions on the survival time of patients; e.g., [**Ishaq et al., 2021**, **Lee et al., 2019**]. In the context of lung cancer, early detection through screening methods helps identify the tumor in its early stage and applying survival analysis to lung cancer patients can aid in early detection and ultimately improve patients' survival outcomes. Meanwhile, in recent years, computer-aided diagnosis has gained significant attention, particularly in medical image data analysis [**Chen et al., 2021**, **Du et al., 2022**, **Li et al., 2018**, **Mielke et al., 2009**]. Deep learning techniques have been increasingly applied to analyze various kinds of medical images due to their effectiveness, for example, [**Hou et al., 2016**, **Gao et al., 2019**, **Wang et al., 2019**, **Ardila et al., 2019**, **Liu et al., 2020**, **Zhang et al., 2020**, **Zhong et al., 2023**].

Despite the promising results obtained by using these techniques, the accessibility of high-quality medical images poses a challenge in applying these techniques. For example, Hou *et al.* [**Hou et al., 2016**] required whole slide tissue images obtained from invasive procedures, Gao *et al.* [**Gao et al., 2019**] required multiple longitudinal CT images captured over time, and Wang *et al.* [**Wang et al., 2019**] required both demographic information and chest CT images.

In addition, most of these studies focused on patients already diagnosed, neglecting those who may be prospective candidates undergoing regular CT screening for early detection. Furthermore, there are few works that have utilized survival analysis, which limits the statistical efficiency of these methods. Considering the significant impact of early detection on patients' survival chances [**Blandin Knight et al., 2017**], there is an urgent need to develop a new approach that can

enhance both the early detection and survival prediction for individuals currently diagnosed and those potentially at risk of lung cancer, while considering the accessibility of the medical image data.

This study aims to utilize deep learning techniques to analyze the potential lung cancer patients' survival hazards only based on their most recent CT images. Inspired by DeepSurv [**Katzman et al., 2018**], which uses demographic information, and DeepConvSurv [**Zhu et al., 2016**], which uses 2D pathological images, we adopt 3D convolutional neural networks (CNNs) to model the non-linear relationship between the risk of lung cancer and the lungs' morphology revealed in CT images. A mini-batched loss involving time-to-event and censoring status is applied for handling the non-convexity caused by the neural networks and allowing the training of large data sets at the same time. In addition, we propose to apply the combination of binary cross-entropy and the mini-batched loss to simultaneously predict whether a potential patient has lung cancer and the risk of dying from it. The promising empirical properties of the proposed method are illustrated by simulation experiments and the application to the National Lung Screening Trial (NLST) dataset [**Team, 2011**].

Our approach has several distinct features: (i) it relates patients' survival with 3D medical image classification; (ii) it considers both existing and potential patients, which helps in the early detection of the disease; and (iii) it requires only one raw CT scan, eliminating the need for additional clinical or longitudinal data or human pathologists' annotation, which makes our approach easy to implement and more accessible than methods that require extensive data collection.

The rest of this chaper is organized as follows: Section 2.2 introduces related works in computer-aided diagnosis and basic knowledge about survival data and the Cox proportional hazards model. Section 2.3 derives the mini-batched loss function of the extended Cox model and introduces the idea of the two-task method and corresponding metrics. Section 2.4 presents the simulation study of the mini-batched loss based on the MNIST dataset and the simulation of the two-task method based on the Nodule-CIFAR dataset. Section 2.5 presents the real data experiment with the two-task method, which includes CT images from potential lung cancer patients.

2.2. Background

2.2.1. Related Work. The Cox proportional hazards model [Cox, 1972] was first proposed to explore the relationship between the survival chance of a patient and a group of explanatory variables through the concept of hazard rate, see (2.1) below. Later, Breslow [Breslow, 1972] and Cox [Cox, 1975] discussed the estimation of model parameters, particularly for the baseline hazard function. Despite it being proposed more than 50 years ago, the Cox model continues to be one of the most widely used models in medical research for investigating patients' survival chances. The use of medical images to aid the diagnosis and treatment of diseases has become increasingly popular. Much research has been conducted on the use of deep learning techniques to analyze medical images as a computer-aided diagnosis. For example, Hou *et al.* [Hou et al., 2016] studied the feature of whole slide tissue image patches with a CNN. Wang *et al.* [Wang et al., 2019] detected lung cancer with CT images and clinical demographics. Ardila *et al.* [Ardila et al., 2019] proposed a CNN-based method to predict lung cancer risk. Gao *et al.* [Gao et al., 2019] performed research in detecting lung cancer with long short-term models. Liu *et al.* [Liu et al., 2020] studied detecting nodules from CT images for lung cancer with adversarial attacks. However, some of these images or data may not be readily available or collected. These methods required whole slide tissue images from an invasive procedure [Hou et al., 2016], or longitudinal medical images captured over time [Gao et al., 2019, Ardila et al., 2019], or demographic information in addition to medical images [Wang et al., 2019]. For more details, refer to [Cao et al., 2020, Singh et al., 2020] for a comprehensive review of deep learning techniques applied to medical images.

While these imaging methods have produced excellent results for the tasks that they were designed for, they did not establish a correlation with patients' survival. Katzman *et al.* [Katzman et al., 2018], for the first time, developed the DeepSurv model to study the non-linear relationship between survival hazards and clinical features. It replaced the linear part $\boldsymbol{\beta}^{\intercal}\boldsymbol{x}$ in the Cox proportional hazards model (2.1) with multi-layer perceptrons $f(\boldsymbol{x})$. However, this model has a limitation in that it can only process clinical information. To address this limitation, DeepConvSurv was then proposed by Zhu *et al.* to predict patients' survival directly from the 2D region of interests (ROI) of pathological images, using CNNs for $f(\boldsymbol{x})$.

In this study, we aim to expand previous research by developing a model that classifies lung cancer occurrence from potential lung cancer patients with only one 3D CT scan and further predicts the patient's relative hazards of dying from lung cancer. Our approach integrates 3D CNNs, binary classification, and the Cox proportional hazards model. By combining these techniques, we aim to establish a direct correlation between potential patients' 3D medical images and patients' survival, which could have significant implications for early lung cancer diagnosis.

2.2.2. Survival Data. Survival analysis typically considers time-to-event data. Let $T^* = \min(T, C)$ be the observed time, where T denotes the event time and C denotes the censored time. Here, T is the time from the beginning of the observation to an event, usually death, disease occurrence, or other experience of interest, which can be unobserved if censoring occurs first. The censored time C is the time after which nothing is observed about the object. In addition to observing T^*, we also have the event indicator: $\delta_i = 1_{\{T_i \leq C_i\}}$ that tells us if the i-th observation T_i is censored or not. In our study, T^* is the observed time from the beginning of the study to either observed death or censoring. If death is observed, $T^* = T$ and $\delta = 1$, if censoring is observed, $T^* = C$ and $\delta = 0$. The objective is to model the event distribution of T,

$$F(t) = P(T \leq t) = \int_0^t f(u)du,$$

where the density function $f(t)$ is

$$f(t) = \lim_{\Delta t \to 0} \frac{P(t < T \leq t + \Delta t)}{\Delta t}.$$

In survival analysis, it is common to alternatively study the survival function $S(t)$, or the hazard function $\lambda(t)$, or the cumulative hazard function $\Lambda(t)$, defined respectively as

$$S(t) = P(T > t) = \int_t^\infty f(u)du,$$

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{P(t < T \leq t + \Delta t | T > t)}{\Delta_t},$$

and

$$\Lambda(t) = \int_0^t \lambda(u)du.$$

Their relationships can be expressed as

$$\lambda(t) = \frac{f(t)}{S(t)},$$

and

$$S(t) = \exp\left(-\Lambda(t)\right),$$

so it's equivalent to studying either of them. In this study, we focus on the density function $f(t)$ and the corresponding likelihood function.

Given a set of right-censored samples $\{T_i^*, \delta_i\}_{i=1}^n$, the likelihood function L is:

$$\begin{aligned} L &= \prod_{i=1}^{n} f(T_i^*)^{\delta_i} S(T_i^*)^{1-\delta_i} \\ &= \prod_{i=1}^{n} \lambda(T_i^*)^{\delta_i} S(T_i^*), \end{aligned}$$

which can be further used for parameter estimation.

2.2.3. Cox Proportional Hazards Model and DeepSurv. The Cox proportional hazards model is one of the most used models for exploring the relationship between the hazards $\lambda(t|\boldsymbol{x})$ and the explanatory covariates $\boldsymbol{x}$. In particular, it assumes proportional hazards and linear contribution of the covariates to the log relative hazards function:

$$\lambda(t|\boldsymbol{x}) = \lambda_0(t)\exp(\boldsymbol{\beta}^\intercal \boldsymbol{x}), \tag{2.1}$$

where t represents time, $\lambda_0(t)$ is the baseline hazard function (an infinite dimensional parameter), $\boldsymbol{x}$ is a set of covariates, and $\boldsymbol{\beta}$ is the corresponding coefficient that measures the effect of the covariates. Cox [**Cox, 1972**, **Cox, 1975**] proposed to use the partial likelihood for estimating $\boldsymbol{\beta}$ with the advantage of circumventing $\lambda_0(t)$. Let $R(t) = \{i : T_i^* > t\}$ be the risk set at time t; i.e., the set of all individuals who are "at risk" for failure at time t. The partial likelihood is the product of the conditional probabilities of the observed individuals being chosen from the risk set to fail:

$$L(\boldsymbol{\beta})_{\text{partial}} = \prod_{i=1}^{n} \left[\frac{\exp(\boldsymbol{\beta}^\intercal \boldsymbol{x_i})}{\sum_{j \in R(T_i^*)} \exp(\boldsymbol{\beta}^\intercal \boldsymbol{x_j})} \right]^{\delta_i},$$

where $R(T_i^*)$ denotes the set of individuals that are "at risk" for failure at time T_i^* in the sample.

The estimate $\hat{\boldsymbol{\beta}}$ for $\boldsymbol{\beta}$ can be obtained by minimizing the averaged negative partial log-likelihood $\mathcal{L}(\boldsymbol{\beta})$, which is convex:

$$\mathcal{L}(\boldsymbol{\beta}) = -\frac{1}{n}\sum_{i=1}^{n}\delta_i\Big[\boldsymbol{\beta}^\intercal \boldsymbol{x_i} - \log\sum_{j\in R(T_i^*)}\exp(\boldsymbol{\beta}^\intercal \boldsymbol{x_j})\Big].$$

The cumulative baseline hazard function can be estimated with the Breslow estimator:

$$\begin{aligned}\hat{\Lambda}_0(t;\boldsymbol{\beta}) &= \sum_{j\notin R(t)}\Delta\hat{\Lambda}_0(T_j^*)\\ &= \sum_{j\notin R(t)}\frac{\delta_j}{\sum\limits_{k\in\mathcal{R}(T_j^*)}\exp(\boldsymbol{\beta}^\intercal \boldsymbol{x_k})}.\end{aligned}$$

The DeepSurv method can be seen as a non-linear version of the Cox model. It replaces the linear log relative hazards term $\boldsymbol{\beta}^\intercal \boldsymbol{x}$ in the Cox model with a non-linear multi-layer perceptron (MLP) $f(\boldsymbol{x};\boldsymbol{\theta})$:

$$\lambda(t|\boldsymbol{x}) = \lambda_0(t)\exp\big(f(\boldsymbol{x};\boldsymbol{\theta})\big),$$

where $f(\boldsymbol{x};\boldsymbol{\theta})$ is a fully-connected MLP parameterized by $\boldsymbol{\theta}$.

2.3. Methodology

2.3.1. Extended Cox Model with Convolution Neural Network. In this study, we modeled patients' hazard function of a certain disease based on 3D medical images. We cannot directly apply the DeepSurv or DeepConvSurv model because MLP or 2D CNN is deficient for 3D image data. Therefore, we extended the DeepSurv model by replacing MLP with a 3D convolution neural network $f(\boldsymbol{x};\boldsymbol{\Theta})$, which predicted the effects of a patient's morphological features $\boldsymbol{x}$ on their hazard rate and parameterized by the weights of the network Θ:

$$\lambda(t|\boldsymbol{x}) = \lambda_0(t)\exp\big(f(\boldsymbol{x};\boldsymbol{\Theta})\big).$$

2.3.2. Loss Function Derivation. Let

$$\Lambda(t) = \Lambda_0(t)\exp\big(f(\boldsymbol{x};\boldsymbol{\Theta})\big)$$

and

$$S(t) = \exp\Big(-\Lambda_0(t)\exp\big(f(\boldsymbol{x};\boldsymbol{\Theta})\big)\Big),$$

so the full likelihood function is

$$L(\Lambda_0, \boldsymbol{\Theta}) = \prod_{i=1}^{n} \Big\{ \Big[\lambda_0(T_i^*) \exp\big(f(\boldsymbol{x_i}; \boldsymbol{\Theta})\big)\Big]^{\delta_i} \times \exp\Big(-\Lambda_0(T_i^*) \exp\big(f(\boldsymbol{x_i}; \boldsymbol{\Theta})\big)\Big)\Big\}.$$

Moreover, the negative log-likelihood becomes

$$\mathcal{L}(\Lambda_0, \boldsymbol{\Theta}) = -\frac{1}{n} \sum_{i=1}^{n} \Big\{ \delta_i \Big[f(\boldsymbol{x_i}; \boldsymbol{\Theta}) + \log \lambda_0(T_i^*)\Big] - \Lambda_0(T_i^*) \exp\big(f(\boldsymbol{x_i}; \boldsymbol{\Theta})\big)\Big\}, \tag{2.2}$$

which depends on both Λ_0 and parameters $\boldsymbol{\Theta}$ in f.

In practice, the prior knowledge of Λ_0 is not available. To overcome this issue, we adopted the non-parametric Breslow estimator, which treated the baseline as a piece-wise constant between event failure times:

$$\begin{aligned} \hat{\Lambda}_0(t; \boldsymbol{\Theta}) &= \sum_{j \notin R(t)} \Delta \hat{\Lambda}_0(T_j^*) \\ &= \sum_{j \notin R(t)} \frac{\delta_j}{\sum\limits_{k \in \mathcal{R}(T_j^*)} \exp\big(f(\boldsymbol{x_k}; \boldsymbol{\Theta})\big)}. \end{aligned}$$

Plugged it into the negative log-likelihood Eq.2.2, we derived the partial likelihood without $\lambda_0(t)$:

$$\mathcal{L}_{\text{fb}}(\boldsymbol{\Theta}) = -\frac{1}{n} \sum_{i} \delta_i \left[f(\boldsymbol{x_i}; \boldsymbol{\Theta}) - \log \sum_{j \in \mathcal{R}(T_i^*)} \exp\big(f(\boldsymbol{x_j}; \boldsymbol{\Theta})\big) \right]. \tag{2.3}$$

We refer to this as the *full-batched loss* in this study. In fact, the procedure of getting partial likelihood of the Cox proportional model can lead us to the equivalent loss function. Given the model $\lambda(t) = \lambda_0(t) \exp\big(f(\boldsymbol{x}; \boldsymbol{\Theta})\big)$, the partial likelihood now becomes

$$L(\boldsymbol{\Theta})_{\text{partial}} = \prod_{i=1}^{n} \left[\frac{\exp\big(f(\boldsymbol{x_i}; \boldsymbol{\Theta})\big)}{\sum\limits_{j \in R(T_i^*)} \exp\big(f(\boldsymbol{x_j}; \boldsymbol{\Theta})\big)} \right]^{\delta_i}, \tag{2.4}$$

The full-batched loss function can be obtained by taking the average of the negative log of the partial likelihood.

Even though the full-batched loss is convex in f, due to the non-convexity of the neural network, the full-batched loss is non-convex. Also, the full-batched loss involves complicated sums over the risk set, which can be as large as the full data set, making it computationally expensive.

To deal with the non-convexity and make it scalable to large datasets, we modified the full-batched loss by first subsampling the data and collecting them to a batch Ω, and then restricting the risk set $R(T_i^*)$ only to contain the subsampled data in the current batch:

$$\tilde{\mathcal{L}}_{\text{mb}}(\Theta) = -\frac{1}{|\Omega|}\sum_{i\in\Omega}\delta_i\left[f(\boldsymbol{x_i};\boldsymbol{\Theta}) - \log\sum_j \exp\left(f(\boldsymbol{x_j};\boldsymbol{\Theta})\right)\right] \tag{2.5}$$

with $j \in \mathcal{R}(T_i^*) \cap \Omega$. We refer to this expression as the *mini-batched loss* in this study. If we set the batch as the full data set, then the mini-batched loss is equivalent to the full-batched loss. The batch size can be as small as 2. By restricting data to a randomly sampled batch, we avoided massive calculations. The mini-batched loss is unlike the minibatch gradient descent with i.i.d. data with respect to the full-batched loss since taking the expectation over random minibatch samples does not give the averaged negative log-likelihood.

As an aside, we can see that the partial likelihood in (2.4) is the likelihood of observing the given order of events, which in this case is the order of individuals' deaths. By evaluating the partial likelihood, we are in effect ignoring any information of the timing of the events beyond just their ordering. This objective and the mini-batch gradient descent described above appear in recommendation system applications where user preferences are expressed via the relative ordering of click-through events. The resulting method is called listwise ranking in the recommendation system literature [**Cao et al., 2007**, **Wu et al., 2018**].

2.3.3. Two-task Method for Disease Diagnosis and Survival Hazard Prediction. Lung cancer is one of the most common cancers. Computed Tomography (CT) images, which include a series of axial image slices that visualize the tissues and nodules within the lung area, can be extremely useful for diagnosis purposes. When given a patient's pulmonary CT images, one objective is to diagnose whether the patient has lung cancer or not, i.e., lung cancer classification. In addition, we hope to predict the severity of cancer by estimating the patient's risk of dying from lung cancer, i.e., survival hazard prediction. Traditionally, to fulfill the two tasks, one option is to train separate models with different losses, respectively: binary cross entropy for lung cancer

classification and mini-batched loss for survival hazard prediction. However, it raises concerns about divergent predictions, which may result in predicting a case without lung cancer but with a high risk of mortality of dying from lung cancer.

The link between lung cancer diagnosis and survival prediction is established through the comprehensive analysis of imaging studies. Extracted information from CT images, such as the presence of lung nodules and detailed characteristics (including size, shape, location, and tumor spread), is not only instrumental in confirming the presence of cancer, but also provides critical details that inform prognosis, guide treatment decisions, and influence survival predictions for individual patients. The higher the probability of having lung cancer inferred from CT images, the more likely it is that the cancer exhibits features associated with an advanced or aggressive nature. These features contribute to an increased risk of mortality, forming the basis for the correlation between the probability of having lung cancer and survival prediction. The integration of imaging data into a holistic approach enhances the precision and personalized nature of lung cancer care.

Recognizing the clinical need to integrate these tasks, we present a novel method capable of simultaneously performing lung cancer classification and survival hazard prediction using the same input – a two-task neural net framework, as illustrated in Fig. 2.1. The output layer, which predicted the log relative hazards $f(\boldsymbol{x};\boldsymbol{\Theta})$, was also used for lung cancer classification with sigmoid activation. This choice is intuitive as the function f represents hazard, implying that a higher hazard is indicative of a higher probability of having lung cancer. Instead of having separate losses, we defined the loss as the sum of binary cross entropy and the batched loss. Let y_i be the indicator of having lung cancer, $\boldsymbol{x_i}$ be the image input to the deep neural network, and $f(\boldsymbol{x_i};\boldsymbol{\Theta})$ be the neural network output for log relative hazards, $P(\boldsymbol{x_i};\boldsymbol{\Theta}) = sigmoid(f\big(\boldsymbol{x_i};\boldsymbol{\Theta}\big))$ is predicted cancer probability:

$$
\begin{aligned}
L(\Theta) = -\frac{1}{|\Omega|}\sum_{i\in\Omega}\Bigg\{ & \delta_i\Big[f(\boldsymbol{x_i};\boldsymbol{\Theta}) - \log\sum_j \exp\big(f(\boldsymbol{x_j};\boldsymbol{\Theta})\big)\Big] \\
& + \Big[y_i \log P(\boldsymbol{x_i};\boldsymbol{\Theta}) + (1-y_i)\log\big(1-P(\boldsymbol{x_i};\boldsymbol{\Theta})\big)\Big]\Bigg\},
\end{aligned} \tag{2.6}
$$

with $j \in \mathcal{R}(T_i^*) \cap \Omega$.

One advantage of this approach is consolidating the goals of cancer classification and survival hazard prediction into a singular model, motivated by the clinical reality that the CT image shows information that is critical for both cancer diagnosis and survival prediction. Training a unified

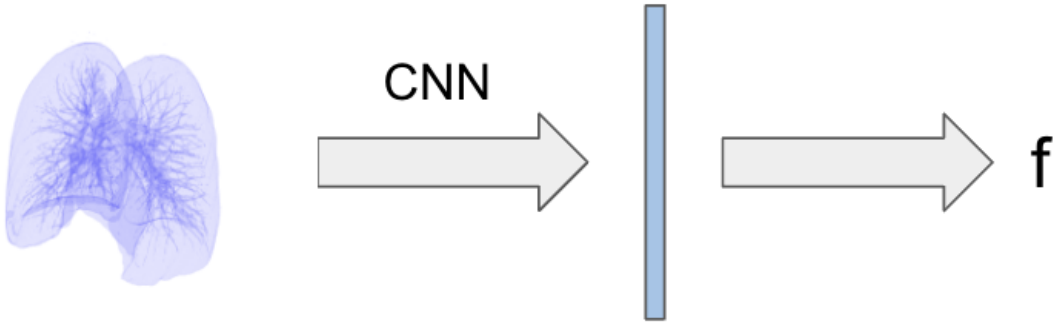


FIGURE 2.1. Two-task convolution neural network illustration.

model concurrently for both objectives with shared neural net parameters promises a more comprehensive understanding and superior predictive performance, while conventional approaches of training separate models with binary cross entropy for cancer classification and mini-batched loss for hazard prediction focus exclusively on one aspect. This two-task method provides a holistic view, bridging the diagnostic and prognostic aspects of lung cancer, and offers a more clinically relevant perspective for personalized patient care decisions. Another advantage lies in the dual losses, which enable more comprehensive supervision of the neural net's fit, thereby preventing overfitting during training.

2.3.4. Evaluation Metrics. For the cancer classification task, we used AUC (area under the ROC curve) to evaluate the model performance. In the hazard prediction task, we employed the concordance index (C-index) for evaluation. C-index, introduced by Harrell *et al.* [**Harrell et al., 1982**], is a goodness of fit measure for models that produce risk scores for censored data. In our context, it estimates the probability that, for any random pair of individuals, the predicted survival times would exhibit the same ordering as their actual survival times. This is equivalent to determining whether, for any random pair of patients, the predicted hazard has the reverse order in comparison to their actual survival times, as patients with higher predicted survival times correspond to lower predicted hazards. The C-index in our context is defined by the following

formula:

$$
\begin{aligned}
C &= \frac{\#\text{ concordant pairs}}{\#\text{ concordant pairs} + \#\text{ disconcordant pairs}} \\
&= P\{\hat{T}_i > \hat{T}_j |\ T_i > T_j, \delta_j = 1\} \\
&\approx P\{\hat{f}_i < \hat{f}_j |\ T_i > T_j, \delta_j = 1\} \qquad (2.7) \\
&= \frac{\sum_{i \neq j} 1\{\hat{f}_i < \hat{f}_j\} 1\{T_i > T_j\} \delta_j}{\sum_{i \neq j} 1\{T_i > T_j\} \delta_j},
\end{aligned}
$$

where approximation (2.7) follows from the argument that a patient with a higher hazard score should have a shorter survival time.

When C-index = 1, it corresponds to the scenario where the order of the predictions is the same as that of the true survival times, while C-index = 0.5 represents a random prediction. Typically, a model with a C-index above 0.7 can be regarded as a good model.

2.4. Simulation Studies

This section reports results from three simulation experiments. Both Simulations A and B focused on the extended Cox model and its prediction of the log relative hazards function f. Simulation A was under the setting where there were event cases only, while Simulation B involved both censored and event cases. Both simulations used the same images from the MNIST dataset and the same generated survival time, but different censoring statuses. We compared the performance of the oracle loss, full-batched loss, and mini-batched loss under the settings of Simulations A and B. Simulation C was designed for the two-task framework, involving both the disease occurrence classification and the survival hazard prediction with the log relative hazards function. We generated a new dataset from the CIFAR-10 dataset, called Nodule-CIFAR. We compared the loss function performance of the combination of binary cross-entropy and full-batched/mini-batched in terms of AUC and C-index.

2.4.1. Simulations A and B.

2.4.1.1. *MNIST Dataset and Time-to-event Data.* We used the MNIST image dataset and generated artificial survival times for digits in our simulations. The MNIST dataset is an image dataset of handwritten digits from 0 to 9; see [**Deng, 2012**]. We selected 2 digits from the MNIST dataset as input images of the neural network with different patterns, w.l.o.g., we selected zeros

TABLE 2.1. Convolution Neural Net Architecture for Simulations A & B

Layer Type	Number of Kernels	Kernel Size	Output Size
Convolution	32	5×5	$28 \times 28 \times 32$
Max Pooling		2×2, stride $= 2$	$14 \times 14 \times 32$
Convolution	64	5×5	$14 \times 14 \times 64$
Max Pooling		2×2, stride $= 2$	$7 \times 7 \times 64$
Flatten			3136
Fully Connected			1024
Fully Connected			128
Fully Connected			1

and ones. We generated the survival time for each digit using an exponential distribution with different constant hazards $\lambda_j = 1 \times \exp(\phi_j)$, $j = 0, 1$, where the baseline hazard $\lambda_0(t)$ was set to 1, and the true log relative hazards was ϕ_j. In Simulation A, all cases were labeled as events. In Simulation B, we randomly labeled half of the individuals who lived beyond the median as censored cases within each digit. The distribution of the test set is shown in Figure 2.2.

2.4.1.2. *Architecture.* Simulations A and B were trained under the same feed-forward convolution neural network, which consisted of a stack of convolution and dense layers. The net structure is listed in Table 2.1.

2.4.1.3. *Results of Simulations A and B.* We introduced the oracle loss (see equation (2.8) and (2.9)). It leverages the prior knowledge of the baseline hazard $\lambda_0(t)$ when compared with the full-batched loss (2.3) and mini-batched loss (2.5), i. In our simulations, w.l.o.g., we set $\lambda_0(t) = 1$ when generating survival time, so that $\Lambda_0(t) = t$. Plugging the baseline hazard into the averaged negative full log-likelihood (2.2) provided us the oracle loss, for which f can be trained:

$$\mathcal{L}_{\text{orc}}(\Theta) = -\frac{1}{n}\sum_{i=1}^{n}\Big[\delta_i f(\boldsymbol{x_i};\boldsymbol{\Theta}) - \exp\big(f(\boldsymbol{x_i};\boldsymbol{\Theta})\big)T_i^*\Big]. \tag{2.8}$$

Due to the non-convexity of neural network f, we used the stochastic gradient descent (SGD) method to minimize the non-convex loss function. Correspondingly, the batched version is provided below.

$$\tilde{\mathcal{L}}_{\text{orc}}(\Theta) = -\frac{1}{|\Omega|}\sum_{i\in\Omega}\Big[\delta_i f(\boldsymbol{x_i};\boldsymbol{\Theta}) - \exp\big(f(\boldsymbol{x_i};\boldsymbol{\Theta})\big)T_i^*\Big], \tag{2.9}$$

where Ω is the selected batch for a training iteration. We will later refer to this as the *oracle loss*.

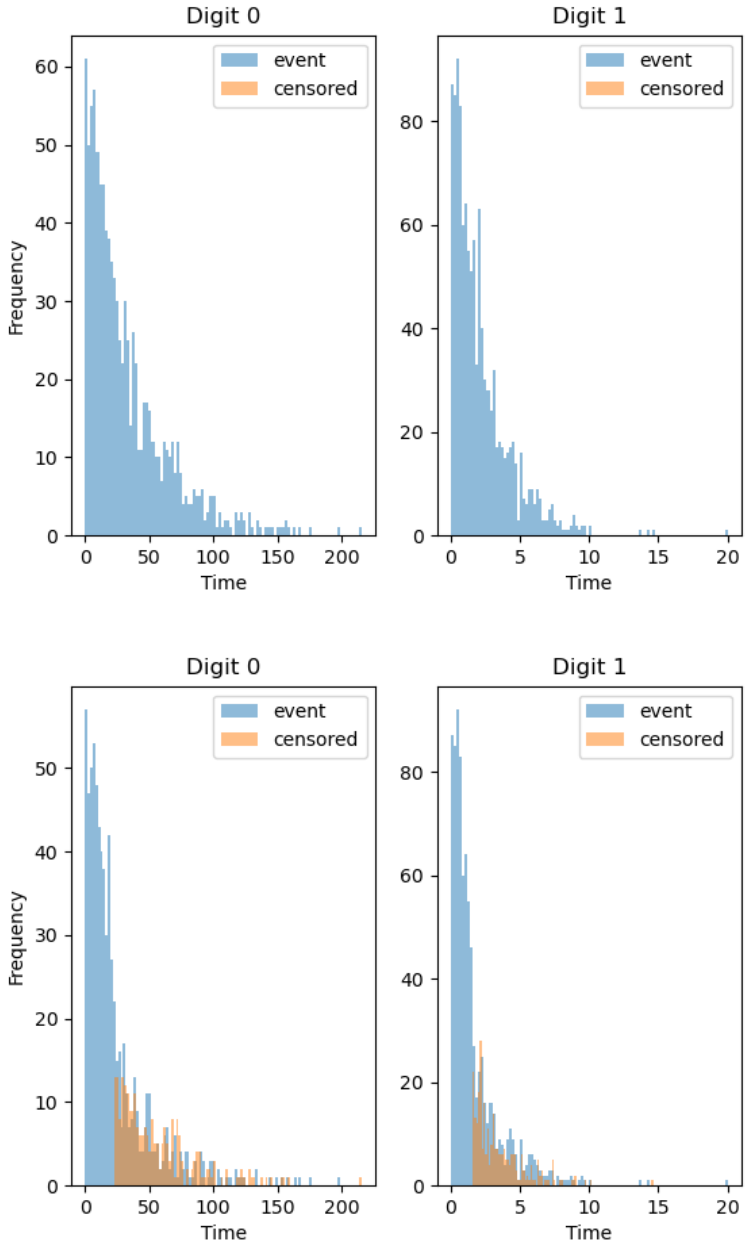


FIGURE 2.2. (a): Survival time distributions for the two digits in Simulation A, without the censoring mechanism; (b): survival time distributions for the two digits in Simulation B, with the censoring mechanism. The censored cases are labeled in orange, which overlaps the upper half of the event cases.

We also calculated the true loss as the baseline for benchmark comparisons. When both the baseline hazard $\lambda_0(t)$ and the log relative hazards ϕ_j were available, we could directly plug them into the averaged negative full log-likelihood (2.2), which gave the true loss.

TABLE 2.2. Simulations A & B: C-indexes under three losses

	Oracle	Full-batched	Mini-batched
A	0.7268	0.7165	0.7189
B w/ censored (C1)	0.7184	0.7146	0.7166
B w/o censored (C2)	0.6845	0.6770	0.6790

Results of Simulations A and B are reported in Figure 2.3 and Table 2.2. In both simulations, the oracle loss settled to the true loss, the oracle loss was less than the batched losses, both batched losses settled to the same value, and the mini-batched loss settled faster than the full-batched loss. This met our expectations since the oracle loss had access to the base rate. In addition, due to the extra information, the C-index trained by the oracle loss is expected to be larger, which was validated in both Simulations A and B, see Table 2.2. In Simulation A, though the C-index curve fluctuated after loss converges, it achieved a high value for both full batched loss and mini-batched loss, showing good rank prediction on the hazards when there is no censoring. In Simulation B, two C-indexes were calculated: C_1 involved both censored and event case, while C_2 involved event cases only. Here, C_1 exceeds 0.7, which means good rank predictions for pairs across censored and event groups and pairs within the event group. Moreover, the faster convergence and small difference between C_{orc} and C_{mb} indicated the feasibility of mini-batched loss for training parameters without prior information of $\lambda_0(t)$.

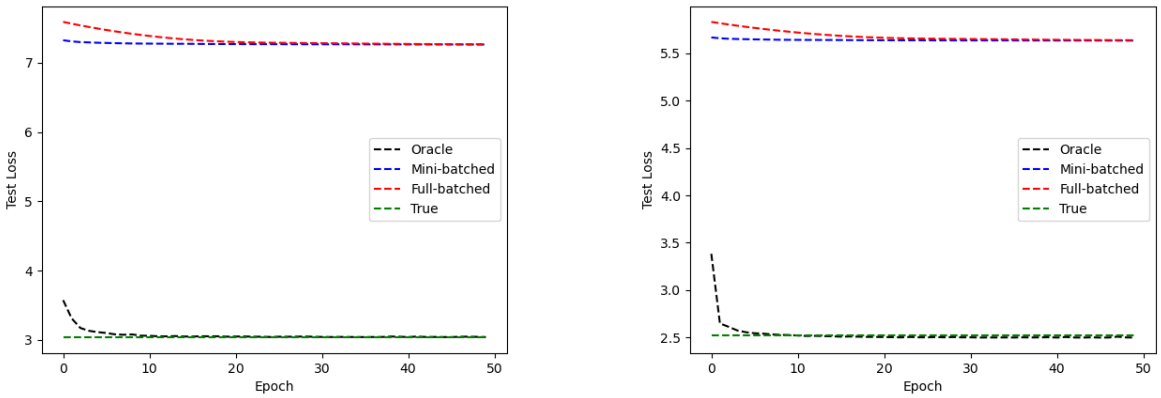


FIGURE 2.3. Different losses by epoch for (a) Simulation A (b) and Simulation B.

2.4.2. Simulation C: Nodule-CIFAR Simulation with Classification and Hazard Prediction.

2.4.2.1. *Nodule-CIFAR Dataset.* We introduced a new dataset, called Nodule-CIFAR, which was generated from the CIFAR-10 dataset [**Krizhevsky et al., 2009**]. Nodule-CIFAR was inspired by Tumor-CIFAR from Gao *et al.* [**Gao et al., 2019**] and simulated benign and malignant nodules on the CIFAR-10 images. In reality, benign nodules typically exhibit smaller sizes with regular round shapes and are non-cancerous, while malignant nodules tend to be larger in size and exhibit irregular shapes. Healthy individuals possess benign nodules, but patients may have both benign and malignant nodules. To simulate this, we introduced black and white dots onto CIFAR-10 images to simulate benign nodules, while dummy nodules were represented as white blobs to simulate malignant nodules.

There were 10,000 samples in the training set and 1,000 samples in the testing set. We randomly assigned images to non-cancerous and cancerous groups with equal probability, so that cancer prevalence was 50% in both training and test sets. Among the cancerous cases, we randomly labeled 50% as censored, and the remaining were labeled as events, the events of failure of dying from cancer. For the non-cancerous cases, they would not die of cancer, so all of them were labeled as censored. Next, we incorporated simulated nodules, either benign or malignant, onto CIFAR-10 images based on their assigned group. The non-cancer images yet featuring benign nodules, displayed numerous small black and white dots distributed across the image to simulate benign nodules. In contrast, the images in the cancer groups had two additional big white patches randomly located in the images, mimicking malignant nodules. Within the cancer group, the censored had relatively smaller white patches compared to the event, because the censored group had not yet reached a deadly stage. The original image categories from the CIFAR-10 dataset were irrelevant in this context; the distinctions between cancer and non-cancer were determined by the presence of simulated white patches. Moreover, within the cancer group, the censoring status was solely associated with the sizes of the simulated white patches. Figure 2.4 is an example of images in the Nodule-CIFAR dataset.

Time-to-event data corresponding to Nodule-CIFAR images were generated based on the largest size of simulated nodules in each image. The recorded time followed an exponential distribution with a parameter of $\lambda = 1 \times \exp(\phi)$, where $\phi \propto size$, the largest size of simulated nodules in each image. This was consistent with our expectation that the larger the nodule size, the larger the hazards, and the smaller the survival time.

FIGURE 2.4. A Nodule-CIFAR example: Non-cancer cases only have small black and white dots scattered over the images, simulating benign nodules. In addition to benign nodules, cancer cases have 2 larger white patches to simulate malignant nodules.

Figure 2.5 shows the distribution of nodule size and survival time for each group. The non-cancer group had smaller nodules on average compared to the cancer group. Within the cancer group, those event cases (eventually died of cancer in simulation) had larger malignant nodules. The time-to-event for the non-cancer group was larger than the cancer group. Within the cancer group, the time-to-event of censored cases was larger than the event cases.

2.4.2.2. *Architecture.* Like Simulations A and B, Simulation C was trained under a feed-forward convolution neural network, which consisted of a stack of convolution and dense layers. The output was used for both disease occurrence classification and hazard prediction evaluation. See Table 2.3 for the structure of the neural network.

2.4.2.3. *Results of Simulation C.* The loss function for the two-task network was the sum of the binary cross entropy and the full-batched/mini-batched loss. To compare the model performance trained with different losses under the same network architecture, see Figure 2.6 for the epoch-wise losses, AUC, and C-index, and Table 2.4 for their stabilized values after the losses converge. As

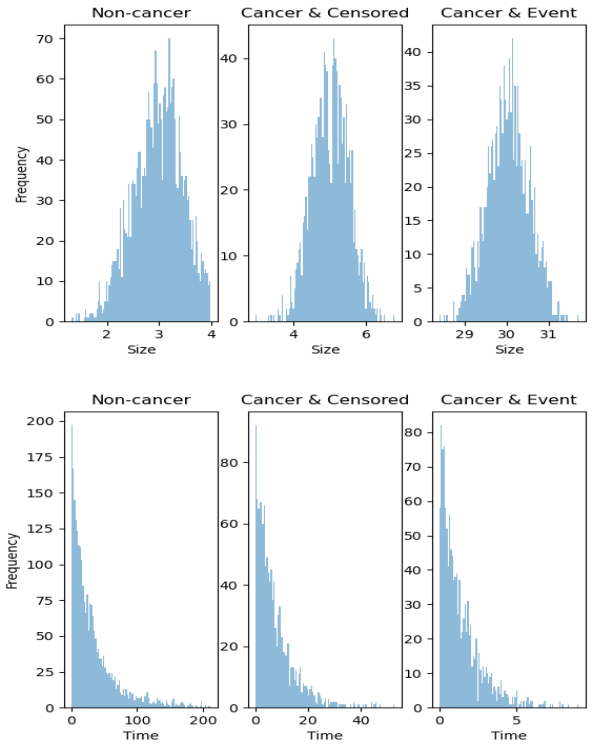


FIGURE 2.5. (a) Nodule size distribution by group. The non-cancer group has smaller nodules on average when compared with the cancer group. Within the cancer group, event cases (those who eventually die of cancer in simulation) have larger malignant nodules. (b) Survival time distribution by group in Nodule-CIFAR. The time-to-event for the non-cancer group is larger than the cancer group. Within the cancer group, the time-to-event of censored is larger than that of the event cases.

TABLE 2.3. Convolution Neural Net Architecture for Simulation C

Layer Type	Number of Kernels	Kernel Size	Output Size
Convolution	32	5×5	$28 \times 28 \times 32$
Max Pooling		2×2, stride $= 2$	$14 \times 14 \times 32$
Convolution	64	5×5	$14 \times 14 \times 64$
Max Pooling		2×2, stride $= 2$	$7 \times 7 \times 64$
Flatten			3136
Fully Connected			100
Fully Connected			10
Fully Connected			1

shown in Figure 2.6a, the one with mini-batched loss (blue) converged much faster than the one with full-batched loss (red); it reached a minimum after a few epochs and stabilized. Figure 2.6b showed both losses outperformed the baseline AUC 50% significantly, which was achieved by predicting all cases as non-cancer, and the model trained with mini-batched loss achieved a slightly higher AUC.

TABLE 2.4. Simulation C: AUC and C-index under two losses

	Full-batched	Mini-batched
AUC	0.770	0.783
C1	0.661	0.677
C2	0.779	0.785

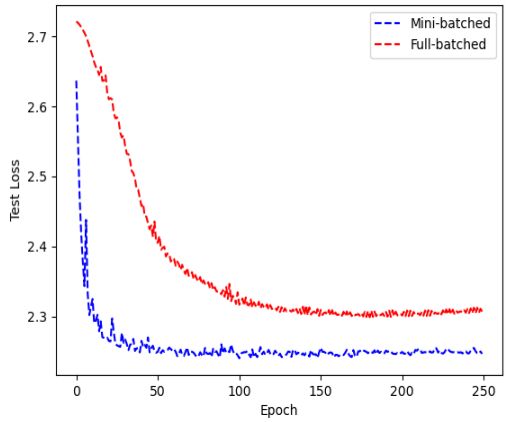
As for the hazard prediction evaluation, we calculated two C-indexes $C1$ and $C2$, where $C1$ was for all cases (cancer and non-cancer, Figure 2.6c) and $C2$ was for the cancer group (Figure 2.6d). Both losses achieved competitive $C1$ and $C2$ values, especially within the cancer group, where $C1$ exceeded 0.75 for both losses. Comparing Figure 2.6c and Figure 2.6d, we noticed the C-index decreased to around 0.65 when it involved the non-cancer group, which was caused by the trade-off between the classification and hazard prediction tasks. Overall, the sum of binary cross entropy and the mini-batched loss performed better in both classification and hazard prediction by achieving higher stabilized AUC and C-index values within fewer epochs.

2.5. Real Data Experiment

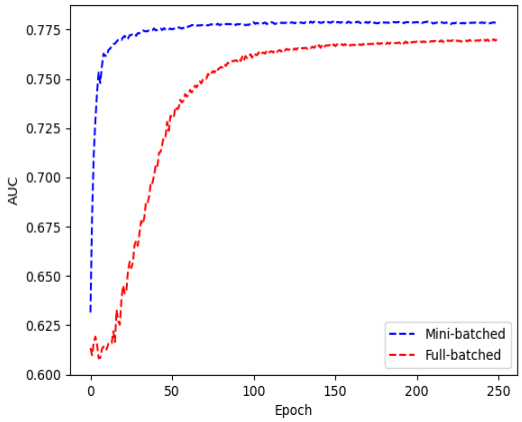
2.5.1. NLST Dataset. The National Lung Screening Trial (NLST) collected medical images and survival information from potential lung cancer patients during 2002-2009, see [**Team, 2011**]. It was a randomized controlled trial to determine whether screening for lung cancer with low-dose helical computed tomography (CT) reduced mortality from lung cancer in high-risk individuals relative to screening with chest radiography (X-ray). Participants were randomly assigned to two study arms in equal proportions. One arm received low-dose helical CT, while the other received single-view chest radiography.

CT images are a set of axial slice images of the human body. They can reveal both normal and abnormal tissues inside the organs. The abnormal tissues of the lungs are called nodules. Nodules usually are spherical but may have other shapes. Each sub-type of nodules has a different cancer probability. Hence, doctors take into consideration all nodules when diagnosing lung diseases with CT images.

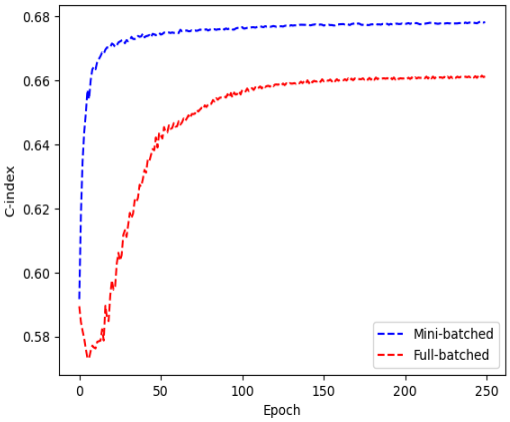
In our experiment, we chose 991 patients who developed cancer during the trial period from a pool of 15,000 patients who received CT treatment. Subsequently, we collected the most recent CT images from these 991 patients confirmed to have lung cancer, among whom 427 passed away due to lung cancer. For the classification task, we similarly gathered the most recent CT images



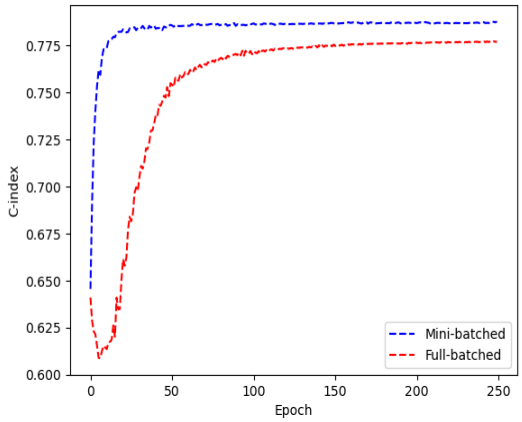
(a) Test Loss by Epoch.



(b) Test AUC by Epoch.



(c) Test C1-index by Epoch.



(d) Test C2-index by Epoch.

FIGURE 2.6. Result of Simulation C. (a): Test loss by epoch; (b): Test AUC by epoch; (c): Test C1-index of all cases by epoch; (d): Test C2-index of the cancer group by epoch. The sum of binary cross-entropy and mini-batched loss performed better in both classification and hazard prediction by achieving higher stabilized AUC, C1, and C2 within fewer epochs.

from an equal number of potential patients who did not have lung cancer. Among the total of 1882 patients, those with confirmed lung cancer cases were assigned a label of $y_i = 1$, while all others were labeled as $y_i = 0$. In addition, those who experienced lung cancer-related mortality were categorized as events of failure (non-censored) with $\delta_i = 1$, whereas the rest were considered censored with $\delta_i = 0$. Each patient's most recent CT examination was utilized as the input image denoted as X. Furthermore, we collected patients' survival time T^* by subtracting their latest exam date from the date they were last known alive.

2.5.2. Preprocessing. In terms of preprocessing the CT images from NLST datasets, we utilized the open-source code [**Zuidhof, 2017**] to segment the lungs from the CT images and applied the nodule detection method described in [**Liao et al., 2019**] to obtain the top 5 suspicious nodule crops as input. For completeness, we provide a brief summary of their method below.

2.5.2.1. *Lung Segmentation.* The CT images are a set of cross-sectional images of the body. Preprocessing for lung segmentation was required before they were ready for the CNN. First, the CT scans should be resampled to $1 \times 1 \times 1 mm^3$ isotropic resolution, then the resampled CT scans were preprocessed with the following main steps:

i. Mask extraction: The first step was to extract the lungs' mask by converting the image to Hounsfield unit (HU) and binarizing the image with the lungs' HU values. HU is a standard quantitative scale for describing radiodensity. Each organ has a specific HU range, and the range remains the same for different people. Here, we used a -320 HU value as the threshold for the lungs. The largest connected component located in the center of the image was extracted as the lungs' mask.

ii. Convex hull computation: The second step was to compute the convex hull of the lungs' mask. Because some nodules might be connected to the outer lung wall and might not be covered by the mask obtained in the previous step, a preferred approach was to obtain the convex hull of the mask. However, it could include other unrelated tissues if one directly computes the convex hull of the mask. To overcome this issue, we first divided the mask into left and right lung masks, then computed their respective convex hulls, and lastly merged them to form the final, whole lungs' convex hull.

iii. Lung segmentation: We obtained a segmentation of the lungs by first multiplying the CT image with the mask and then filling the masked region with tissue luminance.

After completing these three steps, 3D segmented lungs can be extracted. An example is shown in Figure 2.7.

2.5.2.2. *Nodule Detection.* The sizes of the segmented lung images varied for each patient, which went against the requirement for identical image sizes in CNNs to work properly. To resolve this, the segmented images were resampled to the same resolution and fixed slice distance. Although the size of each cropped image might differ due to varying lung sizes among patients, zero padding was used if the image size is less than $224 \times 224 \times 224 \times 1$; otherwise, the central 224-width cubes were

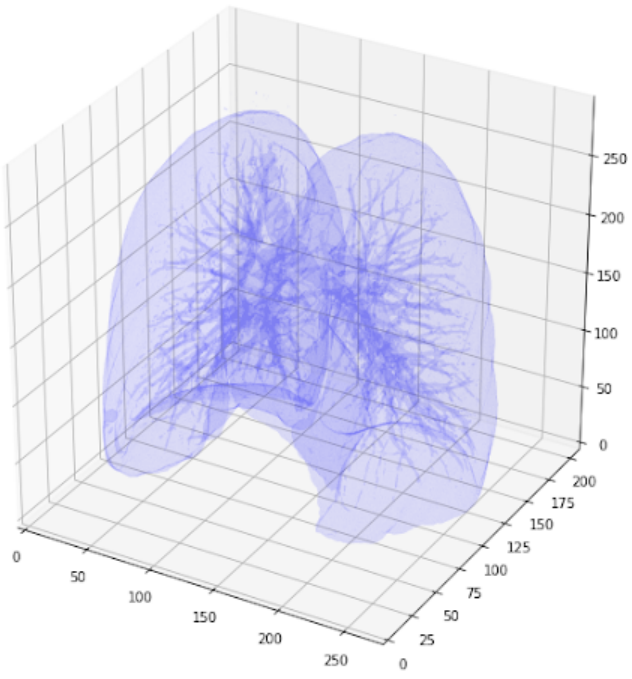


FIGURE 2.7. An example for 3D segmented lungs from CT images.

extracted. An attempt was made to directly input this preprocessed 224-cube into a 3D network for lung cancer classification and hazard prediction. Still, it was computationally time-consuming, and the results were unsatisfactory due to the large size of 3D images and potential memory issues. To address the issue, we followed Liao *et al.* 's nodule detection process [**Liao et al., 2019**]. The nodule detector took in the 3D segmented lung CT image and output predicted nodule proposals with their center coordinates, radius, and confidence. The five most suspicious lung proposals were selected as input X for our network, as Liao *et al.* determined that $k = 5$ was sufficient for recall when different top k proposals with the highest confidence were selected for inference [**Liao et al., 2019**]. For each selected proposal, a $96 \times 96 \times 96 \times 1$ patch centered on the proposed nodule was cropped, resulting in an input size of $5 \times 96 \times 96 \times 96 \times 1$, where one channel represented the number of channels.

2.5.3. Network Structure. The top five regions with the highest nodule confidence were considered for cancer occurrence classification and hazard prediction tasks for each patient. The network had two phases: feature extraction from each lung crop using convolutional layers, and

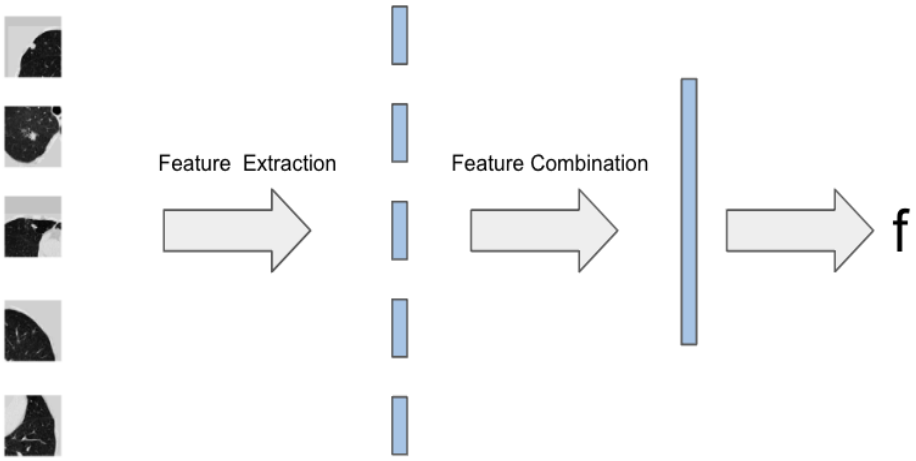


FIGURE 2.8. Network structure with 2 phases: convolution and integration.

feature combination through the integration, as shown in Figure 2.8. The final output f was evaluated with AUC and C-index metrics.

2.5.3.1. *Convolution Phase.* We had three different convolution structures to extract features from the top five nodule crops: Alex3D, VGG163D, and Res-net18. Each took a nodule proposal as input and output a 128-D feature. We also adopted the pre-trained cancer classifier from Liao *et al.* [**Liao et al., 2019**] as a performance benchmark.

2.5.3.2. *3D Alex Net.* Table 2.5 lists layers in Alex 3D. The network was based on the classic 2D Alex Net architecture with modifications specifically tailored for the NLST dataset.

TABLE 2.5. 3D Alex Net architecture for lung CT images

Layer Type	Number of Kernels	Kernel Size	Output Size
Convolution	96	$3 \times 3 \times 3$	$48 \times 48 \times 48 \times 96$
Max Pooling		$3 \times 3 \times 3$	$23 \times 23 \times 23 \times 96$
Convolution	256	$5 \times 5 \times 5$	$23 \times 23 \times 23 \times 256$
Max Pooling		$3 \times 3 \times 3$	$11 \times 11 \times 11 \times 256$
Convolution	384	$3 \times 3 \times 3$	$9 \times 9 \times 9 \times 384$
Convolution	256	$3 \times 3 \times 3$	$9 \times 9 \times 9 \times 256$
Max Pooling		$3 \times 3 \times 3$	$4 \times 4 \times 4 \times 256$
Flatten			16384
Fully Connected			4096
Fully Connected			128

2.5.3.3. *3D VGG16.* Table 2.6 lists the layers in 3D VGG16 developed from 2D VGG16 [**Simonyan and Zisserman, 2014**], with modifications specifically tailored for the NLST dataset.

TABLE 2.6. 3D VGG Net architecture for lung CT images

Layer Type	Number of Kernels	Kernel Size	Output Size
Convolution	64	$3 \times 3 \times 3$	$96 \times 96 \times 96 \times 64$
Convolution	64	$3 \times 3 \times 3$	$96 \times 96 \times 96 \times 64$
Max Pooling		$3 \times 3 \times 3$	$48 \times 48 \times 48 \times 64$
Convolution	128	$3 \times 3 \times 3$	$48 \times 48 \times 48 \times 128$
Convolution	128	$3 \times 3 \times 3$	$48 \times 48 \times 48 \times 128$
Max Pooling		$3 \times 3 \times 3$	$24 \times 24 \times 24 \times 128$
Convolution	256	$3 \times 3 \times 3$	$24 \times 24 \times 24 \times 256$
Convolution	256	$3 \times 3 \times 3$	$24 \times 24 \times 24 \times 256$
Convolution	256	$3 \times 3 \times 3$	$24 \times 24 \times 24 \times 256$
Max Pooling		$3 \times 3 \times 3$	$12 \times 12 \times 12 \times 256$
Convolution	512	$3 \times 3 \times 3$	$12 \times 12 \times 12 \times 512$
Convolution	512	$3 \times 3 \times 3$	$12 \times 12 \times 12 \times 512$
Convolution	512	$3 \times 3 \times 3$	$12 \times 12 \times 12 \times 512$
Max Pooling		$3 \times 3 \times 3$	$6 \times 6 \times 6 \times 512$
Convolution	512	$3 \times 3 \times 3$	$6 \times 6 \times 6 \times 512$
Convolution	512	$3 \times 3 \times 3$	$6 \times 6 \times 6 \times 512$
Convolution	512	$3 \times 3 \times 3$	$6 \times 6 \times 6 \times 512$
Max Pooling		$3 \times 3 \times 3$	$3 \times 3 \times 3 \times 512$
Flatten			13824
Fully Connected			4096
Fully Connected			4096
Fully Connected			128

TABLE 2.7. 3D ResNet-18 architecture for lung CT images

Layer Name	3D Resnet-18	Output Size
Conv1	$7 \times 7 \times 7$,64,stride 2	$48 \times 48 \times 48 \times 64$
Max pooling	$3 \times 3 \times 3$, stride 2	$24 \times 24 \times 24 \times 64$
Res-block1	$\begin{bmatrix} 3 \times 3 \times 3, 64 \\ 3 \times 3 \times 3, 64 \end{bmatrix} \times 2$	$24 \times 24 \times 24 \times 64$
Res-block2	$\begin{bmatrix} 3 \times 3 \times 3, 128 \\ 3 \times 3 \times 3, 128 \end{bmatrix} \times 2$	$12 \times 12 \times 12 \times 128$
Res-block3	$\begin{bmatrix} 3 \times 3 \times 3, 256 \\ 3 \times 3 \times 3, 256 \end{bmatrix} \times 2$	$6 \times 6 \times 6 \times 256$
Res-block4	$\begin{bmatrix} 3 \times 3 \times 3, 512 \\ 3 \times 3 \times 3, 512 \end{bmatrix} \times 2$	$3 \times 3 \times 3 \times 512$
Average-pool		512
Fully Connected		128

2.5.3.4. *3D ResNet-18.* Table 2.7 lists the layers in 3D ResNet-18 developed from a 2D residual network [**He et al., 2016**]. Downsampling was performed by Res-block2_1, Res-block3_1, and Res-block4_1 with a stride of 2.

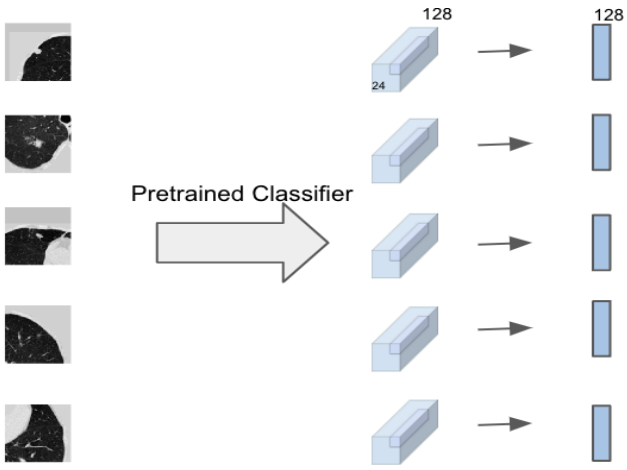


FIGURE 2.9. Using pre-trained classifier to get features from top five suspicious crops

2.5.3.5. *Pretrained Cancer Classifier.* We adopted the pre-trained cancer classifier from Liao *et al.* [**Liao et al., 2019**] as a performance benchmark. Liao *et al.* [**Liao et al., 2019**] proposes a 3D deep neural network based on U-net for cancer probability reference, which has 2 modules: a nodule detection module and a cancer classification module. Because of the limited data size, the classification module (called N-net) integrates the pre-trained detection module as part of the classifier. We followed Liao *et al.* 's process to obtain the features from image patches: For each selected crop, we fed it to the N-net and obtained the last convolutional layer of the nodule classifier, whose size is $24 \times 24 \times 24 \times 128$. The central $2 \times 2 \times 2$ voxels of each proposal feature were extracted and max-pooled, resulting in a 128-D feature, as shown in Figure 2.9.

2.5.3.6. *Integration Phase.* After the convolution phase, the network had five 128D features for each patient. To obtain a single output from these multiple nodule features, three integration methods were explored. The best-performing integration method is shown in Table 2.8, and its graphical representation can be found in Figure 2.10. The features from the top five nodules were individually input into a fully connected layer with 32 hidden units. The maximum value of each feature was considered for the final result after concatenating into a single 5D feature, and a following fully connected layer generated the final output f.

2.5.4. Results. The AUC of the lung cancer occurrence classification and the C-index of the hazard prediction are listed in Table 2.9 for the pair-wise combination of four convolution methods and one integration method. The C-index was calculated based on both cancer and non-cancer

TABLE 2.8. Integration Phase Structure

Layer Type	Output Size
Convolutional Phase Output	128×5
Fully Connected	32×5
Max Pool	1×5
Fully Connected	1

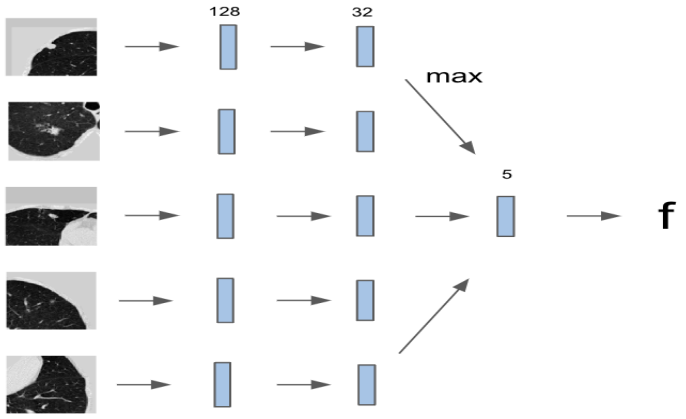


FIGURE 2.10. Graphical representation of feature integration process.

groups, as the non-cancer group in the NLST data set were individuals who had the potential risk of developing cancer. Compared to the pre-trained network of [**Liao et al., 2019**], all three architectures achieved higher AUC and C-index values, indicating better lung cancer classification and survival prediction.

TABLE 2.9. Results of NLST Experiment

	AUC	C-index
3D Alex	0.674	0.601
3D Res18	0.690	0.601
3D VGG16	0.680	0.608
Pretrained	0.550	0.519

2.6. Discussion

The results of our study suggest that the combination of the binary cross-entropy and mini batched loss, obtained by extending the Cox model with CNN, holds the potential to improve the diagnosis and treatment of lung cancer. Our approach demonstrates a high AUC in lung cancer classification and a high C-index in survival prediction, using CT images from the NLST dataset. One strength of

our approach is the use of the mini-batched loss, which effectively handles the non-convexity induced by neural networks and enables the training of large datasets. Additionally, the combination of the mini-batched loss with binary cross-entropy allows for both lung cancer classification and survival hazard prediction. Furthermore, this approach has the potential to be generalized with any type of medical images beyond CT scans. A model can be trained with medical images along with corresponding survival time information to predict the disease occurrence and risk of mortality.

CHAPTER 3

Statistically Consistent Microbiome Community Detection

3.1. Introduction

Microbiome data, generated through high-throughput sequencing technologies, serves as a powerful tool for unraveling the intricacies of microbial communities across diverse environments. Various community detection methods have been proposed, broadly categorized into correlation-based, conditional dependence/graphical models, and network-based methods tailored for trans-kingdom analysis. Among these, correlation-based techniques, which involve grouping samples based on similarities in microbial composition, emerge as the most popular. Nevertheless, they have challenges such as correlation selection in high-dimensional sparse compositional microbiome data and the subjective threshold's impact on predicted outputs.

This chapter introduces the Minimum Description Length (MDL) principle, a concept from information theory and statistics, to address the challenge of subjective threshold. Using the Sparse Correlation Network Investigation for Compositional Data (SCNIC) method as an example, we discuss the consistency of MDL principle in identifying the optimal community structure and we validate it through simulation. Moreover, we perform simulations to validate its effectiveness in selecting optimal threshold across both non-sparse and sparse settings.

The structure of this chapter is as follows: Section 3.2 provides background information on microbiome data and its community detection methods, while Section 3.3 introduces the MDL principle, deriving it under a stochastic block model and discussing its consistency and application in choosing thresholds for community detection methods. Section 3.4 presents simulations to illustrate the consistency and effectiveness of MDL in determining the optimal threshold. Finally, Section 3.5 introduces a real data experiment involving microbiome data from Lake Michigan and Lake Superior, offering practical insights into the application of the MDL principle in a real-world context.

3.2. Background

3.2.1. Characteristics of Microbiome Data: High-Dimensional, Sparse, and Compositional Nature. Microbiome data, often obtained through high-throughput sequencing technologies, contains information about the microbiota (community of microorganisms) and their "theatre of activity" (structural elements, metabolites/signal molecules, and the surrounding environmental conditions) from diverse habitats [**Marchesi and Ravel, 2015**, **Berg et al., 2020**]. Many relevant omic approaches have been proposed for microbiome studies, including metagenomics, metatranscriptomics, and metabolomics. Each type of microbiome data provides unique insights into the structure and function of microbial communities. Researchers often use a combination of these approaches to gain a comprehensive understanding of the microbiome in different environments, including the human body, soil, water, and other ecosystems. [**Aguiar-Pulido et al., 2016**].

Clustered sequences obtained from high-throughput sequencing technologies, commonly known as operational taxonomic units or OTUs, serve as a practical representation of microbial taxa and enable the analysis and characterization of microbial diversity within a sample. Typically, microbiome data is structured into large matrices, where the columns represent samples and the rows represent the counts of OTUs. These tables are often referred to as OTU tables.

Microbiome data is often characterized as high-dimensional and sparse due to several inherent features of microbial communities. Firstly, the high dimensionality arises from the great diversity of microbial taxa in a given environment. Microbial communities can consist of thousands of different species, and each species contributes to the overall dimensionality of the dataset. The sparsity of microbiome data can be attributed to the rarity of many microbial taxa and their uneven distribution across samples. In a typical microbial community, only a subset of taxa is abundant, while the majority are present in low abundance or are rare. This results in a large number of zero counts or low counts for many taxa across the samples, creating a sparse OTU table where most entries are zero [**Weiss et al., 2017**].

Furthermore, microbiome data is inherently compositional due to limitations in current sequencing technologies, which provide information on the relative abundance of microbial taxa within a sample rather than absolute counts. In other words, the data represents the proportion or percentage of each taxon relative to the total microbial community in a given sample. This compositional aspect

is characterized by the fact that the sum of the relative abundances across all taxa in a sample is constant [Gloor et al., 2017]. Acknowledging this compositionality is crucial for accurate analysis and interpretation of microbiome data, as traditional statistical methods may yield misleading results when applied to such datasets [Tsilimigras and Fodor, 2016].

3.2.2. Community Detection for Microbiome Data. Microbes interact with each other and form intricate structures, the microbial communities (also known as modules, clusters and groups). The insights gained from microbial communities help to understand their emergence and progression, with significant implications extending to diverse areas such as ecosystem management, disease prevention, and biotechnological advancements. Examples include studying soil bacterial community dynamics in developing ecosystems [Banning et al., 2011], studying the characteristics of bacterial communities and the crucial shift from oral health to plaque-related diseases [Sbordone and Bortolaia, 2003], as well as research on advancing drug development while considering the effects of antibiotics on microbial community structure in the natural environment [Caracciolo et al., 2015].

Therefore, community detection becomes essential, which also introduces challenges for the comprehensive examination and interpretation of the structure and dynamics of microbial communities within microbiome data [Faust, 2021]. Consequently, many methods have been proposed, generally categorized into three categories: correlation-based methods, conditional dependence/graphical models and network-based methods for trans-kingdom analysis [Matchado et al., 2021]. Among these, correlation-based techniques are the most popular, which involve grouping samples based on similarities in microbial composition, as exemplified by [Fang et al., 2015, Faust and Raes, 2016]. One notable advantage of correlation-based techniques lies in their simplicity and interpretability. These methods provide a quantitative measure of the strength and direction of relationships between microbial taxa, facilitating a clear understanding of the community structure. Additionally, correlation analyses often serve as a valuable initial step for exploratory analyses, generating hypotheses about potential ecological interactions within the microbial community.

However, it's crucial to acknowledge the limitations of correlation-based methods. Firstly, while theoretically, any similarity metric can be utilized to compute pairwise correlations and detect correlation networks, applying these methods to microbiome data is challenging due to the substantial size of the microbiome dataset and the even greater complexity of its interactions, along with the

compositional and sparse nature inherent in microbiome data. Research has indicated considerable variability in sensitivity and precision among traditional correlation detection strategies in microbial datasets [**Weiss et al., 2016**]. SparCC (Sparse Correlations for Compositional Data) was proposed as a solution for estimating correlation values from compositional data, particularly in the context of microbiome studies [**Friedman and Alm, 2012**, **Watts et al., 2019**]. It uses the log-ratio transformed data to get linear Pearson correlations and helps mitigate issues associated with compositional data, where changes in the abundance of one taxon inevitably affect the abundance of others. SparCC was shown to be better suited to avoid spurious correlations compared to Pearson and Spearman correlations [**Weiss et al., 2016**], and has therefore been adopted by algorithms such as SCNIC [**Shaffer et al., 2023**].

In addition, correlation-based methods have another drawback, the subjective nature of choosing a cut-off to determine correlation strength. This subjective decision can significantly influence the detected microbial communities. A more stringent cut-off may reveal only the most robust relationships, while a less stringent one may highlight a broader spectrum of associations. This subjectivity introduces potential bias, and careful consideration of the sensitivity of results to cut-off selection is necessary.

3.3. Methodology

3.3.1. MDL expression, MDL Derivation, MDL consistency Proof.

3.3.1.1. *What is MDL Principle.* The Minimum Description Length (MDL) principle used in information theory and statistics for model selection is to choose the model that provides the most concise and efficient representation of the data. It views a model as consisting of two parts: one part that describes the structure of the model (model complexity), and another part that encodes the specific data given the model (model fit). When comparing different models, MDL seeks the model that minimizes the total length of the description, considering both the complexity of the model and how well it fits the data. In the context of community detection methods applying to microbiome data, choosing the cut-off correlation threshold that minimize the MDL principle helps to identify a community structure that balances the complexity of the model and accurate representation, helping to address the challenge of a subjective cutoff in the analysis.

3.3.1.2. *MDL Derivation.* In this section we derive the MDL principle for Stochastic Block Model (SBM) [**Holland et al., 1983**]. According to MDL principle, the "best" model is the one that achieves optimal lossless compression of the data, in other words, it is capable of storing the data in the hardware memory with the shortest code length. We adopt the "two-part" version of MDL, where the initial part represents the code length of encoding the model, and the subsequent part represents the code length of encoding the residuals that are unexplained by the model. Denoting the code length of z as $CL(z)$, the code length $CL(data)$ of the observed data can be divided into two parts, the model $\mathcal{F}$ and its corresponding residuals $\hat{\epsilon}$, expressed as follows:

$$CL(data) = CL(\mathcal{F}) + CL(\hat{\epsilon}|\mathcal{F}).$$

Consider an undirected acyclic graph with N nodes, represented by a binary adjacency matrix $\boldsymbol{A}$ with dimension $N \times N$. $\boldsymbol{A}_{ij} = 1$ indicates the presence of an edge connecting node i and node j, while $\boldsymbol{A}_{ij} = 0$ indicates disconnection. The graph follows the property of being undirected, implying $\boldsymbol{A}_{ij} = \boldsymbol{A}_{ji}, \forall i, j$. Additionally, acyclic property indicates there is no self-loop in the networks, meaning $\boldsymbol{A}_{ii} = 0, \forall i$. In this scenario, the presence of edges follows a Bernoulli distribution independently, that is,

$$\boldsymbol{A}_{ij} \sim Bernoulli(\Omega_{ij}), \forall i \neq j.$$

Should there be communities among these N nodes, the Stochastic Block Model can be employed to model Ω_{ij}. In this model, the nodes are divided into Q blocks (also referred to as modules, clusters or communities), where Q is unknown, and the edges between nodes are generated based on probabilities determined by the community assignments, the linkage probabilities: let $\boldsymbol{c} = (c_1, c_2, ..., c_P)$ be the community assignment indicator, where $c_i \in \{1, 2, ..., Q\}$; specifically, $c_i = q$ means that node i is in block q. The linkage probabilities depend solely on the community assignment is defined as follows:

$$\Omega_{ij}|(c_i = q, c_j = l) = \pi_{ql}. \tag{3.1}$$

The probability parameter set is represented by $\boldsymbol{\pi} = \{\pi_{ql}, 1 \leq q \leq l \leq Q\}$ and total number of parameters in $\boldsymbol{\pi}$ is $Q(Q+1)/2$. The entire parameter set of SBM is denoted by $\boldsymbol{\psi} = (\boldsymbol{c}, \boldsymbol{\pi})$.

Given community assignments, estimate the link probabilities $\boldsymbol{\pi}$ between communities by maximizing the likelihood of the observed network. However, community assignments can be challenging to determine under the Stochastic Block Model.

Consider a network following the suggested model 3.1. In this case, $\mathcal{F} = \boldsymbol{\psi}$. Consequently $CL(\mathcal{F})$ can be expressed as

$$CL(\mathcal{F}) = CL(\boldsymbol{\psi}).$$

For a specified class assignment c, define $n_q(c) = \#\{i|c_i = q\}$ as the number of nodes in class q. Subsequently, the quantity of potential pairs within/between each block can be represented as

$$N_{ql}(\boldsymbol{c}) = \begin{cases} n_q n_l & q \neq l \\ n_q(n_q - 1)/2 & q = l \end{cases} \tag{3.2}$$

Consider a sequence of networks that can be denoted by a sequence of binary adjacency matrices $\{A_k|k = 1, ..., K\}$ of the same fixed $N \times N$ size, the explicit expression of the MDL criterion is

(3.3)

$$MDL(\boldsymbol{\psi}; \boldsymbol{A}) = (N+1)\log_2 Q + \sum_{q \leq l} \frac{1}{2} \log_2(N_{ql}(\boldsymbol{c})) - \sum_{k=1}^{K} \sum_{i<j} \left[\boldsymbol{A}_{ij} \log_2 \hat{\Omega}_{ij} + (1 - \boldsymbol{A}_{ij}) \log_2(1 - \hat{\Omega}_{ij})\right],$$

where N denotes the number of nodes, Q denotes the number of communities, and $\hat{\Omega}_{ij}$, represented by Equation 3.1, is obtained as the maximum likelihood estimate (MLE) of linkage probability given the community assignment. The derivation details of the MDL can be explored further below.

When assuming the observed network follows 3.1, $\mathcal{F} = \boldsymbol{\psi}$. Consequently $CL(\mathcal{F})$ can be expressed as

$$CL(\mathcal{F}) = CL(\boldsymbol{\psi}).$$

The parameter set $\boldsymbol{\psi}$ comprises both community assignments $\boldsymbol{c}$, and the parameters influencing the link probabilities $\boldsymbol{\pi}$. Therefore,

$$CL(\boldsymbol{\psi}) = CL(\boldsymbol{c}) + CL(\boldsymbol{\pi}|\boldsymbol{c}).$$

Encoding an integer I without a known upper bound requires approximately $\log_2 I$ bits, while with a known upper bound I_u, it takes approximately $\log_2(I_u)$ bits [**Rissanen, 1998**]. For partitioning

a node set of size N into non-overlapping communities, the code length $CL(\boldsymbol{c})$ is given by

$$CL(\boldsymbol{c}) = \log_2 Q + N \log_2 Q,$$

where the first term encodes the number of communities and the second term encodes the community assignment for each node, and Q represents the number of communities.

The code length needed to encode a maximum likelihood estimate of a parameter, derived from n observations, is shown to be $\frac{1}{2}\log_2(n)$ [**Rissanen, 1998**]. In this case,

$$CL(\boldsymbol{\pi}|\boldsymbol{c}) = \sum_{q \le l} \frac{1}{2} \log_2(N_{ql}(\boldsymbol{c})).$$

Combining the above components, we obtain

$$CL(\mathcal{F}) = (N+1)\log_2 Q + \sum_{q \le l} \frac{1}{2} \log_2(N_{ql}(\boldsymbol{c})).$$

Next, we compute the latter term $CL(\hat{\epsilon}|\mathcal{F})$, determined by the negative log-likelihood of the fitted model, [**Rissanen, 1998**]. Under the assumption of SBM that A_{ij} follows a Bernoulli distribution, we have

$$CL(\hat{\epsilon}|\mathcal{F}) = -\sum_{k=1}^{K} \sum_{i<j} \big[\boldsymbol{A}_{ij} \log_2 \hat{\Omega}_{ij} + (1 - \boldsymbol{A}_{ij}) \log_2(1 - \hat{\Omega}_{ij})\big],$$

where Ω_{ij} is determined by Equation 3.1 given $\mathcal{F}$.

Combining the code length components, the overall code length is

$$\begin{aligned} CL(\text{"}data\text{"}) &= CL(\mathcal{F}) + CL(\hat{\epsilon}|\mathcal{F}) \\ &= (N+1)\log_2 Q + \sum_{q \le l} \frac{1}{2} \log_2(N_{ql}(\boldsymbol{c})) - \sum_{k=1}^{K} \sum_{i<j} \big[\boldsymbol{A}_{ij} \log_2 \hat{\Omega}_{ij} + (1 - \boldsymbol{A}_{ij}) \log_2(1 - \hat{\Omega}_{ij})\big]. \end{aligned}$$

This completes the derivation of the MDL principle in Equation 3.3.

3.3.1.3. *MDL-based estimate and its consistency.* Consider K homogeneous networks with N nodes, represented by binary adjacency matrix $\{\boldsymbol{A}_k | k = 1, ..., K\}$ with dimension $N \times N$. Assume the same presence of communities among all N nodes for each k, which can be modeled by SBM presented in the previous section with parameter $\boldsymbol{\psi} = (\boldsymbol{c}, \boldsymbol{\pi})$. We introduce some notations and assumptions first.

The number of the observed within/between each block is denoted as

$$E_{k,ql}(\boldsymbol{c}) = \begin{cases} \sum_{c_i=q} \sum_{c_j=l} \boldsymbol{A}_{k,ij} & q \neq l \\ \sum_{c_i=q} \sum_{c_j=l} \boldsymbol{A}_{k,ij}/2 & q = l \end{cases},$$

then for each $1 \leq k \leq K$, the log-likelihood function for $\boldsymbol{\psi} = (\boldsymbol{c}, \boldsymbol{\pi})$ is

$$\begin{aligned} l_k((\boldsymbol{c}, \boldsymbol{\pi}); \boldsymbol{A}_k) &= \sum_{i<j} \left[\boldsymbol{A}_{k,ij} \log_2 \Omega_{k,ij} + (1 - \boldsymbol{A}_{k,ij}) \log_2(1 - \Omega_{k,ij}) \right] \\ &= \sum_{q \leq l} \left[E_{k,ql}(\boldsymbol{c}) \log_2 \boldsymbol{\pi}_{ql} + (N_{ql}(\boldsymbol{c}) - E_{k,ql}(\boldsymbol{c})) \log_2(1 - \boldsymbol{\pi}_{ql}) \right]. \end{aligned}$$

Recall that $\boldsymbol{\psi} = (\boldsymbol{c}, \boldsymbol{\pi})$, and $\mathcal{M}$ is the set of all possible $\boldsymbol{\psi}$, then the log-likelihood for the K observations can be written as:

$$\mathcal{L}_K(\boldsymbol{\psi}; \boldsymbol{A}) = \sum_{k=1}^{K} l_k((\boldsymbol{c}, \boldsymbol{\pi}); \boldsymbol{A}_k).$$

Then vector $\boldsymbol{\psi} = (\boldsymbol{c}, \boldsymbol{\pi})$ can specify a model for this sequence of networks, and the MDL can be written as

$$MDL(\boldsymbol{\psi}; \boldsymbol{A}) = (N+1) \log_2 Q + \sum_{q \leq l} \frac{1}{2} \log_2(N_{ql}(\boldsymbol{c})) - \mathcal{L}_K(\boldsymbol{\psi}; \boldsymbol{A}). \tag{3.4}$$

We propose to estimate parameter $\boldsymbol{\psi} = (\boldsymbol{c}, \boldsymbol{\pi})$ as the minimizer of observed MDL, which is statistically consistent with the true parameter $\boldsymbol{\psi}^0 = (\boldsymbol{c}^0, \boldsymbol{\pi}^0)$ when $K \to \infty$.

Assumption 1(v) : For any fixed $\boldsymbol{c}$, there exists a $\epsilon > 0$ such that,

$$\sup_{\boldsymbol{\pi} \in \Pi(\boldsymbol{c})} E \mid l_k\left((\boldsymbol{c}, \boldsymbol{\pi}); \boldsymbol{A}_k\right) \mid^{v+\epsilon} < \infty,$$

$$\sup_{\boldsymbol{\pi} \in \Pi(\boldsymbol{c})} E \mid l'_k\left((\boldsymbol{c}, \boldsymbol{\pi}); \boldsymbol{A}_k\right) \mid^{v+\epsilon} < \infty,$$

$$\sup_{\boldsymbol{\pi} \in \Pi(\boldsymbol{c})} E \mid l''_k\left((\boldsymbol{c}, \boldsymbol{\pi}); \boldsymbol{A}_k\right) \mid^{v+\epsilon} < \infty.$$

Note that Assumption 1(1) refer to Assumption 1 with $v = 1$.

Assumption 2 : For any fixed $\boldsymbol{c}$,

$$\sup_{\boldsymbol{\pi}\in\Pi(\boldsymbol{c})} \mid \frac{1}{K}\mathcal{L}_K((\boldsymbol{c},\boldsymbol{\pi};\boldsymbol{A}) - L\left((\boldsymbol{c},\boldsymbol{\pi})\right) \mid \xrightarrow{a.s.} 0,$$

$$\sup_{\boldsymbol{\pi}\in\Pi(\boldsymbol{c})} \mid \frac{1}{K}\mathcal{L}'_K((\boldsymbol{c},\boldsymbol{\pi};\boldsymbol{A}) - L'\left((\boldsymbol{c},\boldsymbol{\pi})\right) \mid \xrightarrow{a.s.} 0,$$

$$\sup_{\boldsymbol{\pi}\in\Pi(\boldsymbol{c})} \mid \frac{1}{K}\mathcal{L}''_K((\boldsymbol{c},\boldsymbol{\pi};\boldsymbol{A}) - L''\left((\boldsymbol{c},\boldsymbol{\pi})\right) \mid \xrightarrow{a.s.} 0,$$

where

$$L\left((\boldsymbol{c},\boldsymbol{\pi})\right) := E\left(l_k\left((\boldsymbol{c},\boldsymbol{\pi})\,;\boldsymbol{A}_k\right)\right),$$

$$L'\left((\boldsymbol{c},\boldsymbol{\pi})\right) := E\left(l'_k\left((\boldsymbol{c},\boldsymbol{\pi})\,;\boldsymbol{A}_k\right)\right),$$

$$L''\left((\boldsymbol{c},\boldsymbol{\pi})\right) := E\left(l''_k\left((\boldsymbol{c},\boldsymbol{\pi})\,;\boldsymbol{A}_k\right)\right).$$

Definition: We define c^b as a bigger model of c^s if $c_i^b = c_j^b$ leads to $c_i^s = c_j^s$ for any node i and j. That is, there exists a function $g : c_i^b \to c_i^s, \forall i, j \in \{1, ..., P\}$.

Under Assumption 1(1) and 2, we have

THEOREM 1. *Let $\{A_k|k = 1, ..., K\}$ be the observations specified by parameters $\boldsymbol{\psi}^0 = (\boldsymbol{c}^0, \boldsymbol{\pi}^0)$. We propose to estimate the parameter $\boldsymbol{\psi}^0$ by*

$$\hat{\boldsymbol{\psi}} = \arg\min_{\boldsymbol{\psi}\in\mathcal{M}} \frac{1}{K} MDL(\boldsymbol{\psi})$$

where $\mathcal{M}$ is the set of all possible values of parameter $\boldsymbol{\psi}$. For any $\hat{\boldsymbol{\psi}} = (\hat{\boldsymbol{c}}, \hat{\boldsymbol{\pi}})$, $\hat{\boldsymbol{\pi}}$ the MLE given K observations' log likelihood $L_K\left((\hat{\boldsymbol{c}}, \boldsymbol{\pi})\,;\boldsymbol{A}\right)$ with when $\hat{\boldsymbol{c}}$ denotes the estimated community assignment and $\boldsymbol{\Pi}\left(\hat{\boldsymbol{c}}\right)$ denotes the parameter space of $\boldsymbol{\pi}$ given $\hat{\boldsymbol{c}}$, that is,

$$\hat{\boldsymbol{\pi}} = \arg\max_{\boldsymbol{\pi}\in\boldsymbol{\Pi}(\hat{\boldsymbol{c}})} \mathcal{L}_K\left((\hat{\boldsymbol{c}}, \boldsymbol{\pi})\,;\boldsymbol{A}\right)$$

Then we have the estimated community assignment $\hat{\boldsymbol{c}}$ must be bigger than the true community assignment $\boldsymbol{c}^0$, and there exists a function $g : \hat{\boldsymbol{c}}_i \to \boldsymbol{c}_i^o$, such that

$$\hat{\boldsymbol{\pi}}_{ql} \xrightarrow{a.s.} \boldsymbol{\pi}^o_{g(q)g(l)}.$$

The detailed proof of Theorem 1 is in Appendix A.

3.3.2. SCNIC with MDL Thresholding. Sparse Correlation Network Investigation for Compositional Data (SCNIC) is a method designed for analyzing microbial community data, specifically compositional data generated from high-throughput sequencing technologies such as 16S rRNA gene sequencing. It first generates correlation with SparCC to identify meaningful correlations while accounting for the sparse and compositional nature of the microbiome data and avoiding spurious correlations. In particular, the method detects modules by initially employing complete linkage hierarchical clustering on correlation coefficients, resulting in a feature tree. Modules are then defined as subtrees in which the correlations between all pairs of tips exhibit an R-value surpassing the specified threshold. Consequently, its characteristic is the generation of modules only when all features demonstrate correlations above a user-defined threshold and different threshold leads to the identification of different modules.

By aiming to minimize the MDL, which ensures statistically consistent module assignment and linkage probability estimation, the MDL principle is employed to identify an optimal threshold for detecting significant co-occurrence relationships within a microbiome network. Among a set of potential thresholds, the one yielding the lowest MDL is chosen. This selection achieves a balance between preserving meaningful relationships and mitigating noise, resulting in a network structure that is more interpretable and reliable.

3.4. Simulation Studies

We conduct two simulations: one to validate the consistency of the MDL principle in determining the optimal community structure and the other to confirm the effectiveness of MDL in determining the threshold for community detection structure, utilizing SCNIC as an illustrative example of community detection methods.

To assess the quality of clustering algorithms, we use Normalized Mutual Information (NMI), a measure of similarity between two partitions of a set. It takes into account the chance agreement between the clustering results and ground truth.

Let $\boldsymbol{c} = (c_1, c_2, ..., c_n)$ be the true community assignment of n nodes in a network, $\hat{\boldsymbol{c}} = (\hat{c}_1, \hat{c}_2, ..., \hat{c}_n)$ be the estimated community assignment obtained from a clustering algorithm. Let $H(.)$ denote entropy, and $I(.)$denote mutual information. NMI is calculated using the formula:

$$NMI(\boldsymbol{c}, \hat{\boldsymbol{c}}) = \frac{I(\boldsymbol{c}; \hat{\boldsymbol{c}})}{\sqrt{H(\boldsymbol{c}) \cdot H(\hat{\boldsymbol{c}})}}$$

where

- $H(\boldsymbol{c}) = -\sum_{i=1}^{n} P(c_i) \cdot \log(P(c_i))$, where $P(c_i) = \frac{\text{Number of nodes in community } c_i}{n}$
- $H(\hat{\boldsymbol{c}}) = -\sum_{j=1}^{n} P(\hat{c}_j) \cdot \log(P(\hat{c}_j))$, where $P(\hat{c}_j) = \frac{\text{Number of nodes in community } \hat{c}_j}{n}$
- $I(\boldsymbol{c}; \hat{\boldsymbol{c}}) = \sum_{i=1}^{n} \sum_{j=1}^{n} P(c_i, \hat{c}_j) \cdot \log\left(\frac{P(c_i, \hat{c}_j)}{P(c_i) \cdot P(\hat{c}_j)}\right)$, where
  $P(c_i, \hat{c}_j) = \frac{\text{Number of nodes in both } c_i \text{ and } \hat{c}_j}{n}$

The NMI score ranges from 0 to 1, where 0 indicates no mutual information and 1 indicates perfect agreement. Higher NMI values suggest better agreement between the partitions. In the context of clustering evaluation, normalized mutual information is particularly useful when the number of clusters in the partitions may vary, as it normalizes the score to account for different clusterings with different numbers of clusters.

3.4.1. Simulation 1. We generate data using the stochastic block model and assess the consistency of MDL in determining the optimal community structure across varying sample sizes. The true network data, generated by the stochastic block model, consists of two modules, namely module 1 and module 2, each with a size of 100. For pairs of nodes within the same module (either module 1 or module 2), the correlation is set to 0.95 and 0.5, respectively. For pairs of nodes across module 1 and module 2, the correlation is set to 0.3. We consider sample sizes of $n = 10, 50, 100, 500, 1000, and 10000$. In each repetition For each specific n, we performed SCNIC module detection under different correlation thresholds ranging from 0 to 0.8 with increments of 0.02. The threshold that minimized MDL was denoted as r, and the corresponding values of mdl_r, nmi_r, number of detected modules Q_r, and sizes of the largest and second-largest modules, $S1_r$ and $S2_r$, were recorded. This process was repeated 100 times for each sample size to compute the mean and standard deviation of r, mdl_r, nmi_r, Q_r, $s1_r$, and $s2_r$ over the 100 repetitions.

The outcomes, illustrated in Figure 3.1, reveal the trends as the sample size n increases. With increasing sample size, the minimizer of MDL consistently approaches the true threshold of 0.5. Simultaneously, the minimized MDL experiences a decline, the Normalized Mutual Information (NMI) metric demonstrates an augmentation, and the number of detected modules converges toward the actual count. Additionally, we assessed the sizes of the top two detected modules at the

minimizer, observing that the largest size stabilizes at the true value when n reaches 100, while the second-largest size converges to the true value as n reaches 1000. In summary, an increase in sample size results in the stabilization of the minimized MDL, and the minimizer converges towards the true threshold. This, in turn, contributes to the consistency observed in module detection, as reflected by the NMI, the number of modules, and the sizes of the top two modules.

3.4.2. Simulation2. This subsection introduces simulations about exploring the application of the MDL principle in selecting the optimal threshold within the context of SCNIC. Under both non-sparse and sparse settings, we generated count data and calculated its sample correlation (Pearson's correlation for non-sparse count data, SparCC for sparse count data). Subsequently, SCNIC was applied for community detection at various thresholds, and the differences between the true community assignment and the predicted community assignment, where MDL is minimized, were examined.

Non-sparse Setting. Consider SBM settings where there are two, three, and four blocks, each block of size 100. Generate multivariate normal data as the count data based on the linkage correlation associated with the specific SBM setting. Calculate its sample correlation of the non-sparse count data. Apply the SCNIC method for module detection with a sequence of thresholds, which starts with the minimum value in the corresponding sample correlation matrix. The potential threshold range begins with the minimum value in the sample correlation and extends to the maximum threshold where all nodes are allocated into modules. As the threshold increases, the number of nodes assigned to modules decreases, resulting in an incomplete module assignment.
In Figure 3.2, the top row displays sample correlations under scenarios with two, three, and four blocks, meanwhile, the bottom row displays the MDL and NMI of the predicted community assignment, identified by SCNIC, at various thresholds corresponding to the correlations depicted above.
The MDL values in these non-sparse settings exhibit a semi-oscillating pattern, characterized by peaks followed by flat segments and repeating in this manner. The minima occur within the flat segments, signifying that MDL achieves consistent local minimums across various threshold ranges during these plateaus. In contrast, the NMI values show an overall contrasting pattern, reaching maximum values at the points corresponding to the MDL minima. We opt for the rightmost threshold value within each plateau as the minimizer of the MDL. These particular values signify

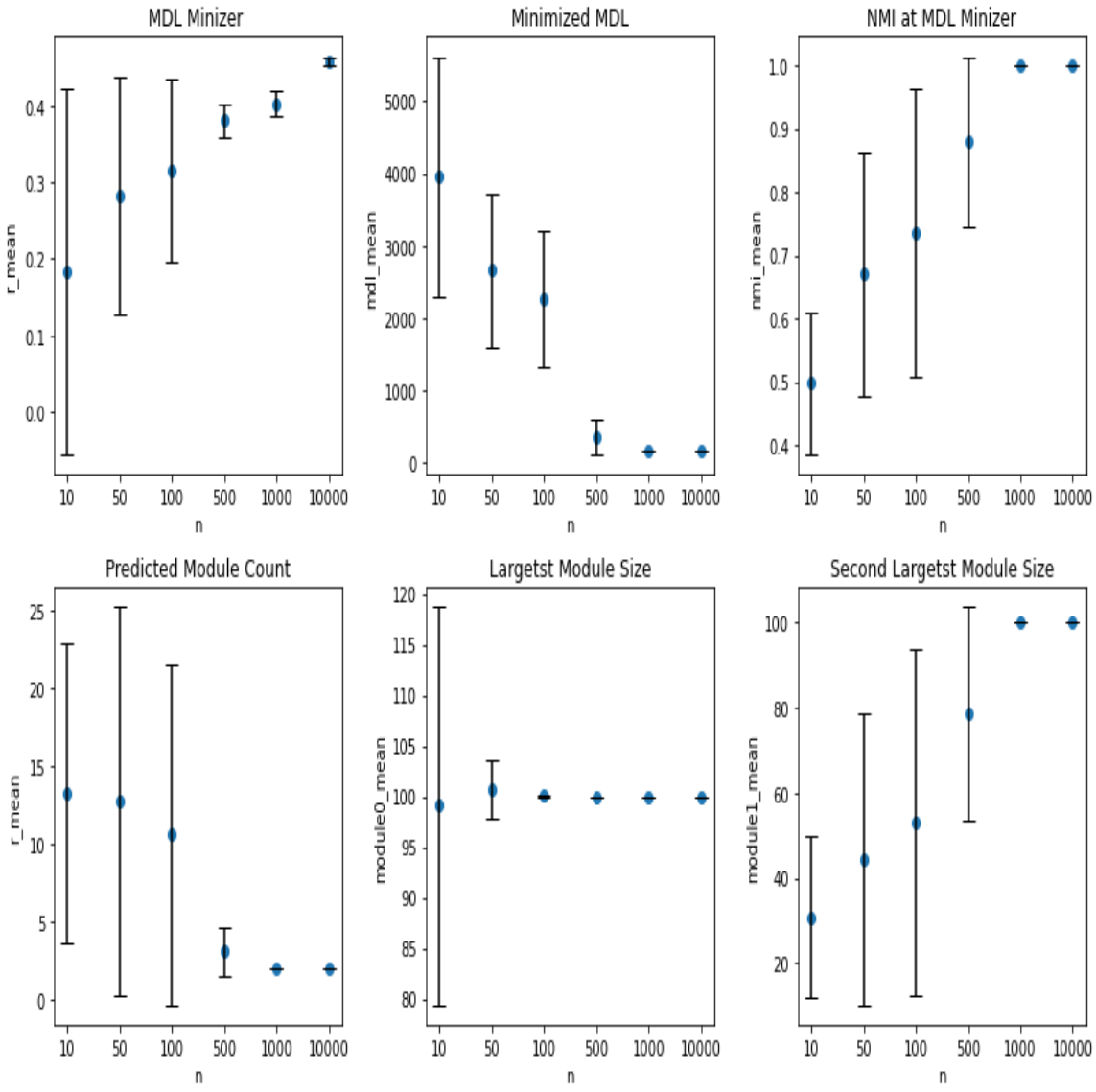


FIGURE 3.1. The figure showcases six plots each capturing metric and their associated means and standard deviations across different sample sizes (n), with error bars indicating the degree of variability. The first row, from left to right, includes subplots depicting the minimizer of MDL, the minimized MDL, and the Normalized Mutual Information (NMI). The second row showcases subplots representing the number of detected modules, the size of the largest detected module, and the size of the second-largest detected module. As n increases, the method of minimizing MDL provides results that align closely with the true values or characteristics of the data.

the MDL change points, corresponding to critical thresholds. Table 3.1 shows metrics in non-sparse settings involving two, three, and four blocks, including the sample correlation range within each block, the corresponding minimizer, the minimized MDL, and the NMI at the minimizer. It is

noteworthy that these minimizers align with the cut-off or boundary values for blockwise sample correlations.

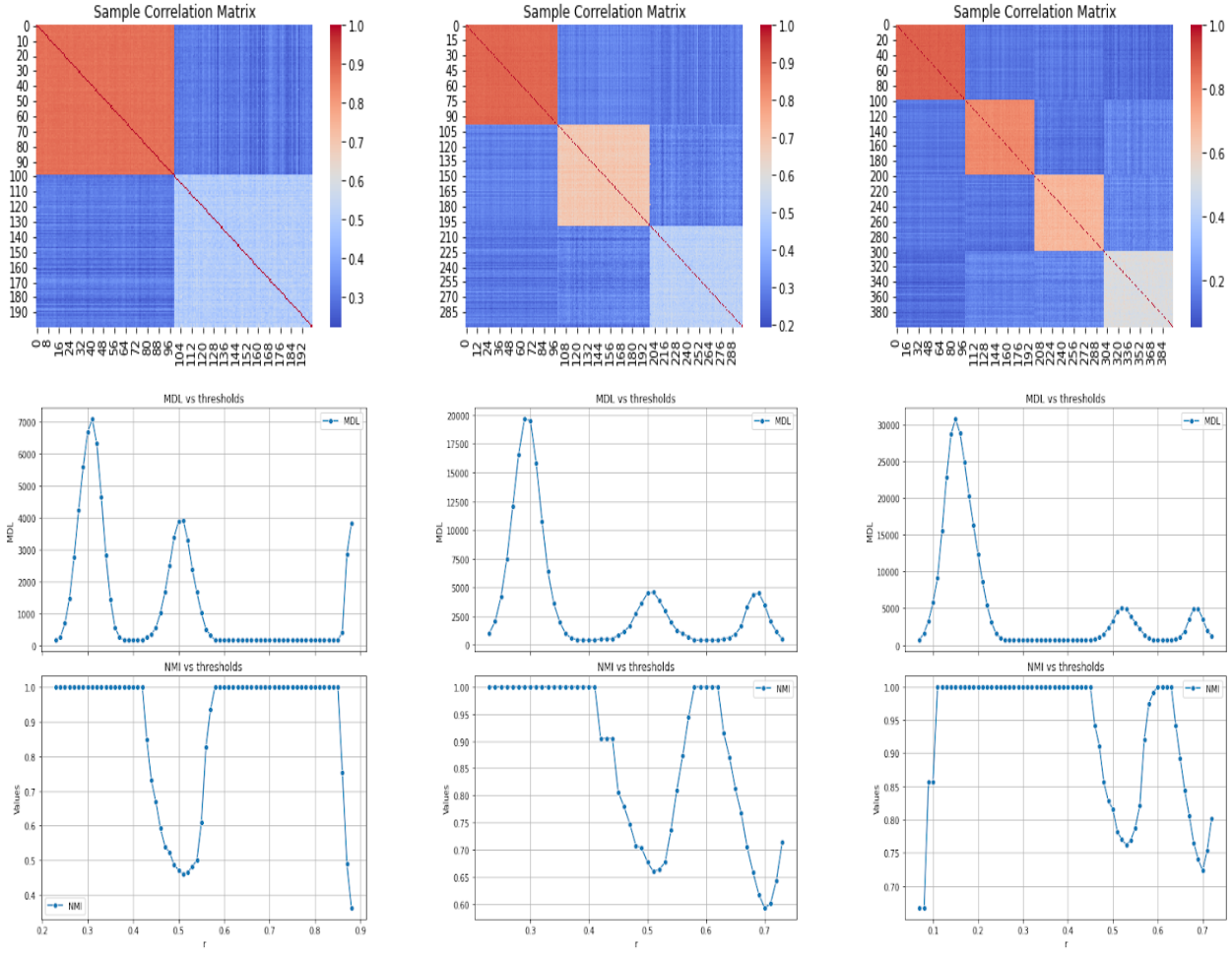


FIGURE 3.2. From left to right, sample correlations under Stochastic Block Model (SBM) settings with 2, 3, and 4 blocks in the top row, accompanied by the corresponding Minimum Description Length (MDL) and NMI of predicted community assignments detected by SCNIC at various thresholds in the bottom row.

Sparse Setting. Consider SBM settings involving one sparse block of size 50 along with one block (size 100), two blocks (each size 50), or three blocks (each size 50). Generate multivariate normal data in small sample sizes (50, 50, and 30 respectively) as the count data based on the linkage correlation associated with the specific SBM setting. Calculate its sample correlation with SparCC for the sparse count data. Apply the SCNIC method for module detection with a sequence of thresholds, which starts with the minimum value in the corresponding sample correlation matrix.

Blocks	Blockwise Sample Correlation Range	Minimizer	MDL	NMI
2	$\begin{bmatrix} \begin{bmatrix} 0.855 & 0.896 \\ 0.223 & 0.379 \end{bmatrix} & \begin{bmatrix} 0.223 & 0.379 \\ 0.426 & 0.576 \end{bmatrix} \end{bmatrix}$	0.42, 0.85	172.35	1.00
3	$\begin{bmatrix} \begin{bmatrix} 0.867 & 0.911 \\ 0.229 & 0.357 \end{bmatrix} & \begin{bmatrix} 0.229 & 0.357 \\ 0.623 & 0.733 \end{bmatrix} & \begin{bmatrix} 0.203 & 0.375 \\ 0.193 & 0.385 \end{bmatrix} \\ \begin{bmatrix} 0.203 & 0.375 \end{bmatrix} & \begin{bmatrix} 0.193 & 0.385 \end{bmatrix} & \begin{bmatrix} 0.415 & 0.574 \end{bmatrix} \end{bmatrix}$	0.41, 0.62	402.13	1.00
4	$\begin{bmatrix} \begin{bmatrix} 0.853 & 0.904 \\ 0.091 & 0.213 \end{bmatrix} & \begin{bmatrix} 0.091 & 0.213 \\ 0.764 & 0.83 \end{bmatrix} & \begin{bmatrix} 0.084 & 0.221 \\ 0.061 & 0.219 \end{bmatrix} & \begin{bmatrix} 0.052 & 0.197 \\ 0.108 & 0.261 \end{bmatrix} \\ \begin{bmatrix} 0.084 & 0.221 \\ 0.052 & 0.197 \end{bmatrix} & \begin{bmatrix} 0.061 & 0.219 \\ 0.108 & 0.261 \end{bmatrix} & \begin{bmatrix} 0.639 & 0.732 \\ 0.1 & 0.278 \end{bmatrix} & \begin{bmatrix} 0.1 & 0.278 \\ 0.453 & 0.592 \end{bmatrix} \end{bmatrix}$	0.45, 0.63	680.39	1.00

TABLE 3.1. Simulation Results Under Non-Sparse Settings

In Figure 3.3, the top row displays SparCC correlations under scenarios with two, three, and four blocks (including the sparse block), the bottom row displays the MDL and NMI of the predicted community assignment, identified by SCNIC, at various thresholds corresponding to the correlations depicted above.

The MDL values in these sparse settings exhibit a more irregular oscillating pattern, and not necessarily there are flat segments. The NMI values show an overall contrasting trend, and it reach maximum values at the thresholds corresponding to the MDL minimizer.

Table 3.2 shows metrics in sparse settings involving two, three, and four blocks, including the sample SparCC correlation range within each block, the corresponding minimizer, the minimized MDL, and the NMI at the minimizer. It is noteworthy that these minimizers align with the cut-off or boundary values between the non-sparse and sparse blocks.

The above observations suggest a consistent pattern in MDL behavior across different threshold ranges in both non-sparse and sparse settings. The alignment of MDL minimizers with block cut-off points suggests its capability to effectively detect the community structure. Therefore, in scenarios with multiple threshold candidates, selecting the one that minimizes MDL proves to be a valuable strategy for enhancing the accuracy of community detection.

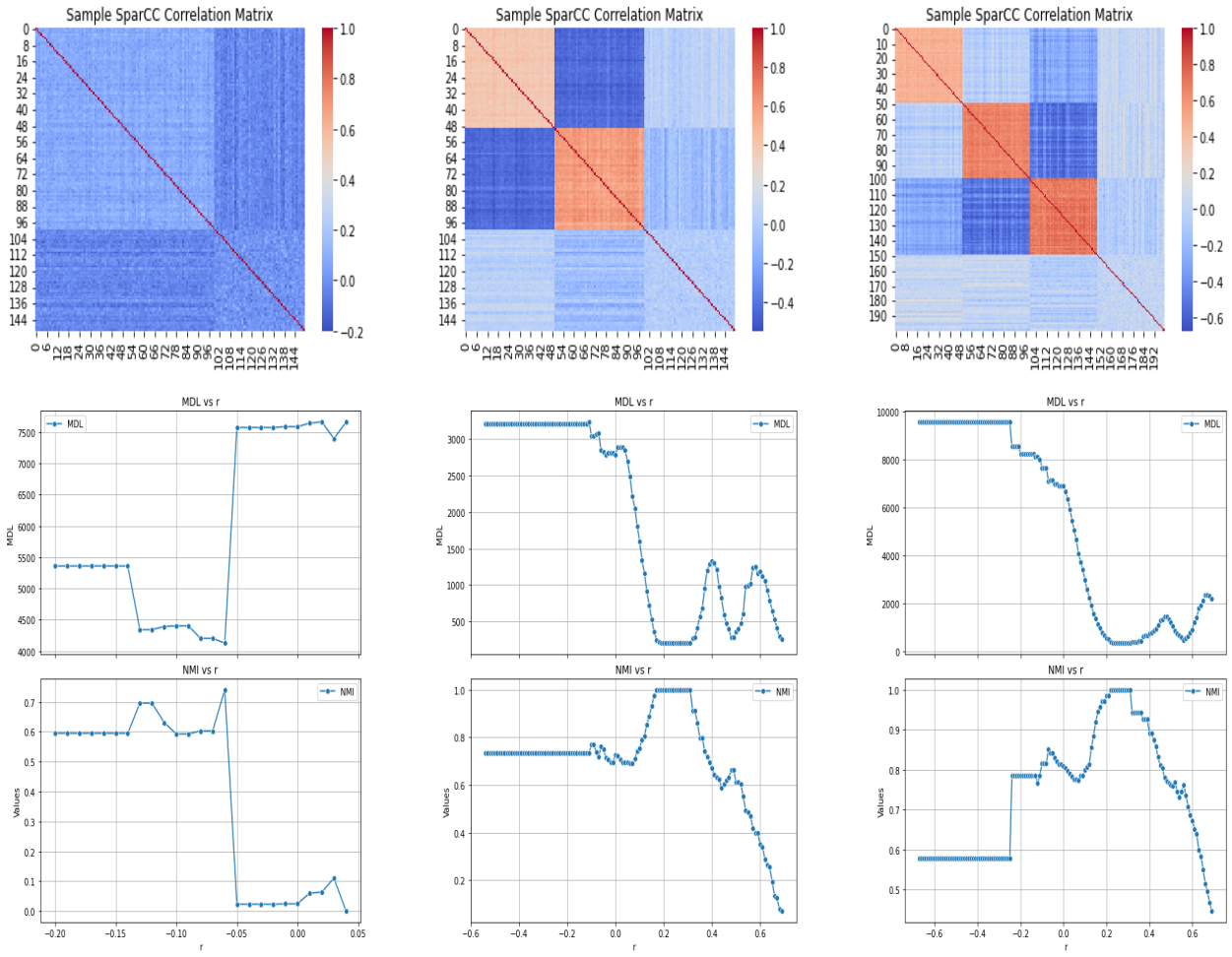


FIGURE 3.3. From left to right, sample SaprCC correlations under sparse Stochastic Block Model (SBM) settings with 2, 3, and 4 blocks in the top row, accompanied by the corresponding Minimum Description Length (MDL) and NMI of predicted community assignments detected by SCNIC at various thresholds in the bottom row

3.5. Real Data Experiment

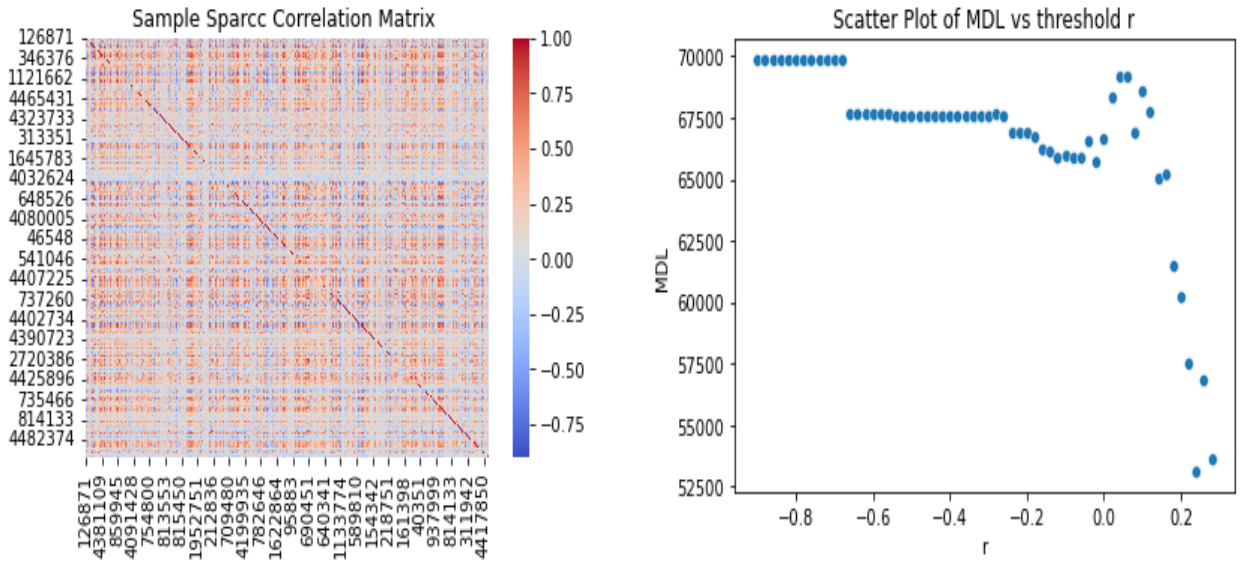
The Great Lakes dataset, initially featured as part of the Earth Microbiome Project [**Thompson et al., 2017**], serves as a valuable resource for understanding microbial patterns in two prominent lakes: Lake Michigan and Lake Superior. This study delves into the microbial relative abundance across various depths, ranging from 5 to 3654 meters, encompassing 16 samples from Lake Michigan and 33 samples from Lake Superior. Acquired through QIITA accession number 1041 [**Gonzalez et al., 2018**]). Preprocessing steps includes the demultiplexing and application of quality control to FASTQ data, the uniform trimming of sequences to the same length of 150, and the selection of

Blocks	Block Correlation	Minimizer	MDL	NMI
2	$\begin{bmatrix} \begin{bmatrix} -0.056 & 0.209 \\ -0.200 & 0.126 \end{bmatrix} & \begin{bmatrix} -0.200 & 0.126 \\ -0.124 & 0.172 \end{bmatrix} \end{bmatrix}$	-0.06	4124.61	0.74
3	$\begin{bmatrix} \begin{bmatrix} 0.311 & 0.500 \\ -0.548 & -0.304 \end{bmatrix} & \begin{bmatrix} -0.548 & -0.304 \\ 0.434 & 0.707 \end{bmatrix} & \begin{bmatrix} -0.098 & 0.187 \\ -0.292 & 0.112 \end{bmatrix} \\ \begin{bmatrix} -0.098 & 0.187 \\ -0.292 & 0.112 \end{bmatrix} & & \begin{bmatrix} -0.116 & 0.161 \end{bmatrix} \end{bmatrix}$	0.31	199.47	1.00
4	$\begin{bmatrix} \begin{bmatrix} 0.317 & 0.561 \\ -0.243 & 0.182 \end{bmatrix} & \begin{bmatrix} -0.243 & 0.182 \\ 0.529 & 0.757 \end{bmatrix} & \begin{bmatrix} -0.49 & -0.132 \\ -0.677 & -0.247 \end{bmatrix} & \begin{bmatrix} -0.202 & 0.266 \\ -0.221 & 0.225 \end{bmatrix} \\ \begin{bmatrix} -0.490 & -0.132 \\ -0.202 & 0.266 \end{bmatrix} & \begin{bmatrix} -0.677 & -0.247 \\ -0.221 & 0.225 \end{bmatrix} & \begin{bmatrix} 0.532 & 0.823 \\ -0.274 & 0.215 \end{bmatrix} & \begin{bmatrix} -0.274 & 0.215 \\ -0.207 & 0.211 \end{bmatrix} \end{bmatrix}$	0.31	336.25	1.00

TABLE 3.2. Simulation Results Under Sparse Settings

closed-reference Operational Taxonomic Units (OTUs) These steps are done with QIIME2, resulting in 4149 OTUs.

We begin with a table of 4149 OTUs, and 481 of these remained after removing OTUs not present in at least 50% of the samples. SCNIC is applied with SparCC and a sequence of thresholds, starting from -0.90, the minimum correlation value, to the threshold of 0.28, the maximum threshold where almost all nodes are allocated into modules. The MDL achieves the minimum at a threshold of 0.24. Starting from 0.28, not all nodes are included in the detected modules, which is an incomplete node assignment, therefore it is not included for considering the optimal threshold.



(a) (b)

FIGURE 3.4. This figure displays the SparCC heatmap of the filtered Great Lake data and the MDL at different thresholds. The minimum MDL is achieved at a threshold of 0.24.

3.6. Discussion

Simulation 1 results reveal that as the sample size increases, the MDL stabilizes, and the corresponding minimizer converges toward the true threshold. Concurrently, the number of detected modules, the top 2 module sizes, and NMI approach the true values as the standard deviation decreases, demonstrating consistency.

In Simulation 2, both in non-sparse and sparse settings, the local minimums correspond to the actual cut-off correlation in the sample correlation matrix. This provides evidence supporting the effectiveness of minimizing MDL for selecting the optimal threshold.

In the real data experiment, we achieve the MDL minimum. However, there are limitations. Firstly, the microbiome sample is of limited size, and the data is sparse and high-dimensional. To address this, we apply preprocessing steps, filtering out OTUs with at least 50% zeros across samples. Additionally, we manually select the upper bound of the threshold where most nodes are assigned to modules. Although an MDL minimizer is found in this case, further insights are needed to interpret the biological significance.

CHAPTER 4

Understanding Distortion Patterns of Adversarial Attacks

4.1. Introduction

Well-trained deep neural networks are capable of achieving outstanding performance in many areas, including image-related classification tasks [**Simonyan and Zisserman, 2014**, **Krizhevsky et al., 2017**, **He et al., 2016**]. However, various studies have shown that they may not be fully reliable and can be fooled by adversarial examples – images that are carefully crafted to fool such deep neural networks by introducing imperceptible perturbation to the original images [**Szegedy et al., 2013**, **Goodfellow et al., 2014**, **Carlini and Wagner, 2017**, **Chen et al., 2018**]. This raises serious security concerns for the AI community. Many works have been done to study and defend against adversarial attacks [**Zhang et al., 2019**, **Xie et al., 2017**, **Meng and Chen, 2017**, **Sadeghi et al., 2020**]. In particular, adversarial detection methods have been proposed to determine whether an input image is an adversarial example or not [**Metzen et al., 2017**, **Gong et al., 2017**, **Li and Li, 2017**, **Zheng and Hong, 2018**, **Feinman et al., 2017**]. Moreover, it is helpful for the defender if reverse engineering can be done to reveal more information about the attacks based on the detected adversarial examples. For example, there are three main attack families to perform attacks: gradient-based, score-based, and decision-based, which rely on the gradient, predicted score, and predicted label of the victim model, respectively. Based on the detected adversarial examples, if the defender can tell what type of attack is used, the defender will know what information has been leaked to the attacker. Consequently, the defender can modify the model accordingly to prevent further attacks. Some works have been done to study the reverse engineering of adversarial attacks: Pang *et al.* [**Pang et al., 2020**] proposed the query of interest (QOI) estimation model to infer the adversary's target class by model queries in black-box settings. Goebel *et al.* [**Goebel et al., 2021**] estimated adversarial setup from image sample for gradient-based attacks FGSM [**Goodfellow et al., 2014**] and PGD [**Madry et al., 2017**]. Gong *et al.* [**Gong et al., 2022**] proposed a general formulation of the reverse engineering of deceptions problem that

can estimate adversarial perturbations and provide the feasibility of inferring the intention of an adversary.

In this chapter, we first demonstrated that, given an adversarial example, the corresponding attack family can be accurately identified with a simple model. Once we had established this, we turned our attention to analyzing the specific features of each type of attack to understand the underlying differences between them better. Section 4.2 covers preliminary information presented in the chapter. Section 4.3 focuses on our image classifier that accurately identifies attack families (gradient-based, score-based, or decision-based). In Section 4.4, we provide an extensive analysis of the features associated with each type of attack.

4.2. Preliminaries

Notations: We consider an image classifier $f(\cdot)$ as the victim model of adversarial attacks. The input to the classifier is $\boldsymbol{x_0} \in [0,1]^{w,h,c}$, a c-channel image sample with width w and height h. The true label associated with $\boldsymbol{x_0}$ is denoted as y, and the adversarial example generated from $\boldsymbol{x_0}$ is denoted as $\boldsymbol{x^*}$. We denote $f(\boldsymbol{x_0})$ as the predicted score vector and $c(\boldsymbol{x_0}) = \arg\max_i f(\boldsymbol{x_0})$ as the predicted label, indicating the i^{th} label has the highest prediction score.

Adversarial Examples: An adversarial example $\boldsymbol{x^*}$ and the original image $\boldsymbol{x_0}$ are visually indistinguishable, but their predicted labels are different. That is, $\mathcal{D}(\boldsymbol{x_0}, \boldsymbol{x^*})$ is very small in some distance metric $\mathcal{D}$, while $c(\boldsymbol{x^*}) \neq c(\boldsymbol{x_0})$. Taking Fig.4.1 as an example, humans will recognize that the two images are of the same horse. However, the image on the right is generated by adding imperceptible perturbations to the original image on the left, which causes a particular classifier to classify it as a cat. Existing methods use L_p metrics to evaluate the distance between adversarial and original samples. This study focuses on L_2 and L_∞, the most commonly used metrics in adversarial attacks.

Data Sets and Victim Models: We use CIFAR10 [**Krizhevsky et al., 2009**] image data set with ten different classes of resolution 32×32. Another data set we use is Tiny Imagenet [**Deng et al., 2009**], which has 200 classes, and the resolution of the images is 64×64. For CIFAR-10, the victim model is VGG-16 with batch normalization [**Simonyan and Zisserman, 2014**], of which accuracy is 93.34%. For Tiny ImageNet, the victim model architecture is ResNet18 [**He et al., 2016**] with 68.64% accuracy.

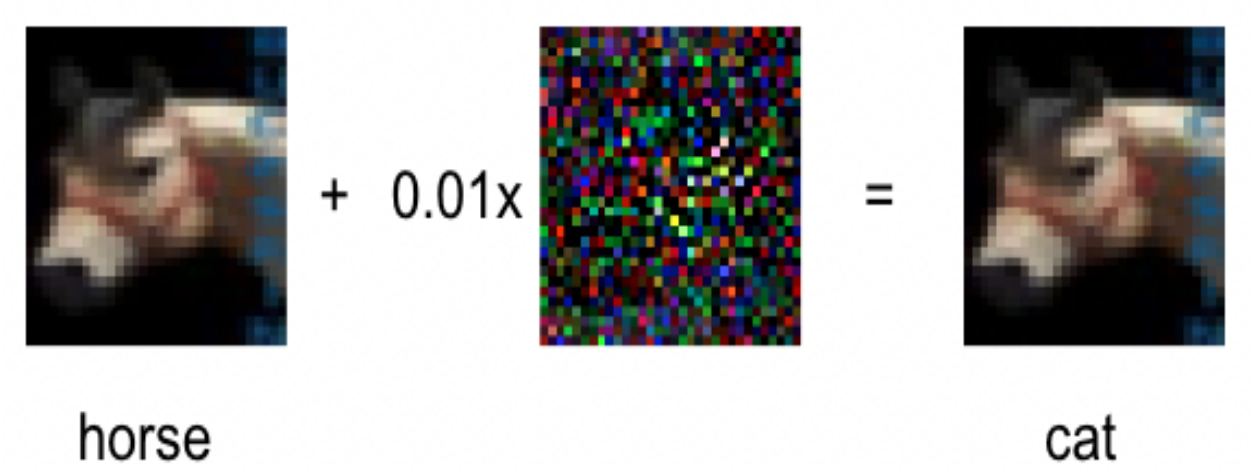


FIGURE 4.1. An adversarial example generated by Boundary attack: introducing adversarial perturbations to the horse image causes a classifier to label it as a cat.

Adversarial Attacks: Different attack methods can be classified into two categories according to their goals: untargeted and targeted. Untargeted attacks are successful as long as the adversarial example is misclassified. Targeted attacks, instead, are successful only when the adversarial example is classified into a target class. Take Fig.4.1 as an example; the untargeted attack is successful if the right-side image is not classified as a horse, while the targeted attack is successful only when it is predicted as a cat if the target class is a cat. In this study, all experiments are based on untargeted attacks.

Depending on the information required, existing attack methods can be divided into three categories: gradient-based, score-based, and decision-based. The gradient-based attack is also known as a white-box attack, in which all information of the victim model is revealed to the attacker so that the attackers can calculate gradients. Popular gradient-based attacks are FGSM [**Goodfellow et al., 2014**], PGD [**Madry et al., 2017**] and C&W [**Carlini and Wagner, 2017**]. If an attacker only has access to the predicted score of the victim model, it is a score-based attack, also known as a soft-label black-box setting. Popular score-based attacks include ZOO [**Chen et al., 2017**], NES [**Ilyas et al., 2018**] and Square [**Andriushchenko et al., 2020**]. In practical scenarios, the attacker only has access to the predicted labels of the model. Attacks under this setting are called decision-based attacks. Examples of such attacks include those described in [**Li et al., 2022**] and [**Zanddizari et al., 2021**], as well as popular methods like Boundary [**Brendel et al., 2017**], Sign-OPT [**Cheng et al., 2019**] and HopSkipJump (HSJ) [**Chen et al., 2020**]. Table 4.1 lists

six representative attacks under different settings in L_2 or L_∞ metrics. In this study, we conduct attack family classification with these attacks and study their perturbation patterns. Adversarial images are generated based on ART package [Nicolae et al., 2018].

TABLE 4.1. Representative attacks of different metrics from different families under L_2 and L_∞.

	L_2	L_∞
gradient-based	C&W	PGD
score-based	ZOO	Square
decision-based	Boundary	HopSkipJump

Perturbation Visualization: Perturbations are the differences between the adversarial example and the corresponding original image, showing how the original image is modified. Since perturbations are imperceptible, we amplify the perturbation by 100 times for visualization purposes in this study.

4.3. Reverse Engineering of Adversarial Attacks

Most current reverse engineering methods focus on analyzing specific attack methods. However, this section explores the potential for identifying attack families associated with adversarial examples. Successful detection of attack families (gradient-based, score-based, or decision-based) can be a useful tool for defenders, as it allows them to understand better the level of information that has been leaked during attacks so that defenders can properly assess the potential impact of that attack family.

When an adversarial attack is launched, it exploits weaknesses in the model: gradient-based attacks take advantage of the model gradients; score-based attacks rely on the predicted scores of the model; and decision-based attacks rely on the predicted labels. This knowledge can help develop an effective response to the attack. Overall, by identifying the specific attack families and taking targeted actions to address the vulnerability exploited by the attack, defenders can improve model resilience and minimize the damage caused by attacks.

4.3.1. Experiments: Classifying Attack Families. We generate adversarial examples of each attack family and two metrics (L_2 and L_∞) using attacks in Table 4.1 with data sets and victim models mentioned in Section 4.2.

For the L_2 attacks, the perturbation upper bounds are 1.00 and 5.00 on CIFAR10 and Tiny ImageNet, respectively. The perturbation upper bound is 0.03 for different L_∞ attacks on both CIFAR10 and Tiny ImageNet.

With the generated adversarial examples, we perform the following experiments: (1) classifying attack families in L_2 metric; (2) classifying attack families in L_∞ metric; and (3) classifying attack families with adversarial examples of both L_2 and L_∞ metrics. A classifier with VGG16 architecture is trained for multi-class classification to identify the attack family based on adversarial examples. The same architecture is used for both CIFAR10 and Tiny ImageNet in all the following experiments except in Experiment D, where the task is six-class classification, and the last layer has six neurons instead of three.

Experiment A: For L_2-norm based attacks, we choose C&W (gradient-based), ZOO (score-based), and Boundary (decision-based) as representatives of each attack family. If all three attacks can successfully fool the victim model by modifying the same original image under the perturbation bound, we keep the corresponding adversarial examples and split them into training and test sets for the attack family classification task. These adversarial examples are called successful adversarial examples across three attacks.

Experiment B: For L_∞-norm based attacks, we choose PGD (gradient-based), Square (score-based), and HopSkipJump (decision-based) as representative attacks. A similar procedure is applied as in Experiment A to obtain the training and test sets for the attack family classification task.

Experiment C: Adversarial examples in Experiments A and B are merged into three classes so that each class contains adversarial examples generated by attacks from the same attack family but different norm metrics. Similarly, we only keep successful adversarial examples across six attacks. Gradient-based class includes adversarial examples generated by C&W(L_2) and PGD(L_∞). Score-based class includes ZOO(L_2) and Square(L_∞). Decision-based class includes Boundary(L_2) and HopSkipJump(L_∞). The classification task is to do a three-class classification, identifying the attack family given an adversarial example.

Experiment D: To investigate if there are not just differences between attack families but also differences between attack methods, this experiment uses the same data as in Experiment C but performs six-class classification to identify specific attacks, not attack families.

TABLE 4.2. Accuracy of attack family classification task (Experiments A, B, C) and attack method classification task (Experiment D) on CIFAR10 and Tiny ImageNet without original images.

	CIFAR10	Tiny ImageNet
Experiment A	82.74%	81.08%
Experiment B	95.51%	96.96%
Experiment C	85.58%	85.77%
Experiment D	76.30%	73.84%

The first three rows (Experiments A, B, C) in Table 4.2 show the attack family classification accuracies on CIFAR10 and Tiny ImageNet datasets. The last row (Experiment D) shows the attack method classification accuracy. The first three experiments achieve high accuracies on different datasets, which suggests that attack families modify the image in different ways and machines can learn the pattern based on adversarial examples, although adversarial examples are indistinguishable from the original images to humans. The testing accuracies are not bad for Experiment D, which implies that attacks of the same family also have different patterns.

In many real-world scenarios, whether the input has been perturbed or not is often unknown to the models. We incorporate non-perturbed original images into the classification task to address this concern. The outcomes of the experiment can be found in Table 4.3. The experimental setup remains consistent, with the only variation being the inclusion of original images as a distinct category in the input. Except for Experiment A, all experiments stay at a high accuracy level. Experiment A experiences a decrease in accuracy due to its utilization of the L_2 norm attack, which considers the cumulative perturbations across all pixels, leading to smaller discrepancy to original images when a certain threshold of the cumulative sum is applied. On the other hand, the L_∞ norm attack focuses on the maximum perturbed pixel while allowing other pixels to be perturbed as long as their individual perturbations are below the threshold, leading to more noticeable perturbation patterns.

TABLE 4.3. Accuracy of attack family classification task (Experiments A, B, C) on CIFAR10 and Tiny ImageNet with original images.

	CIFAR10	Tiny ImageNet
Experiment A	74.84%	65.45%
Experiment B	92.15%	91.87%
Experiment C	80.65%	89.36%

4.3.2. Robustness of Attack Family Classification. This section presents evidence for the robustness of attack family identification, even when they have varying perturbation levels or involve ensemble attacks.

4.3.2.1. *With Various Norm Limits.* In this section, we demonstrate that attack family types of adversarial examples can be accurately identified despite having different perturbation levels. The CIFAR10 dataset was used for Experiment A and Experiment B to investigate the effect of different limits on the attack family classification under L_2 and L_∞ norms. Experiment A of classifying L_2 attacks from three different attack families achieved high levels of accuracy across a range of limit values, including 1.0, 0.8, and 0.6. Similarly, in Experiment B of classifying L_∞ attacks from three different attack families, high levels of accuracy were achieved across a range of limit values including 0.03, 0.02, and 0.01, see Table 4.4. However, we observed a decrease in accuracy as L_2 or L_∞ norm limit becomes smaller, which can be attributed to the limited number of successful adversarial samples across three attacks under smaller limits.

TABLE 4.4. Accuracy of attack family classification for various L_2 and L_∞ limits.

L_2 Norm Limit	1.0	0.8	0.6
Accuracy	82.74%	81.30%	76.30%

L_∞ Norm Limit	0.03	0.02	0.01
Accuracy	95.51%	92.63%	81.57%

4.3.2.2. *With Ensemble Attack.* Auto attack is an ensemble attack algorithm that includes four attacks: APGD-CE, APGD-DLR, FAB [**Croce and Hein, 2020a**], and Square Attack, where APGD-CE and APGD-CE are two extensions of the PGD attack overcoming failures due to suboptimal step size and problems of the objective function [**Croce and Hein, 2020b**]. This algorithm iterates over the list of attacks until an adversarial example is successfully generated. Though both gradient and score information are involved, we consider auto attack as a gradient-based attack for the purpose of the attack family classification task. In our evaluation, we classify adversarial examples of CIFAR10 generated by Auto-attack(gradient-based), ZOO(Score-based), and Boundary(decision-based) under L_2 norms and achieved an accuracy of 83.40%; under L_∞ norms, we evaluated Auto-attack(gradient-based), Square(Score-based), HopSkipJump(decision-based), and accuracy achieved 97.40%. These results demonstrated that different attack families

could be effectively classified even when the gradient-based attack involves more than just gradient information. Besides, the accuracy of the attack family classification remains consistent regardless of the specific attacks involved.

4.4. Exploring Characteristics of Attack Families

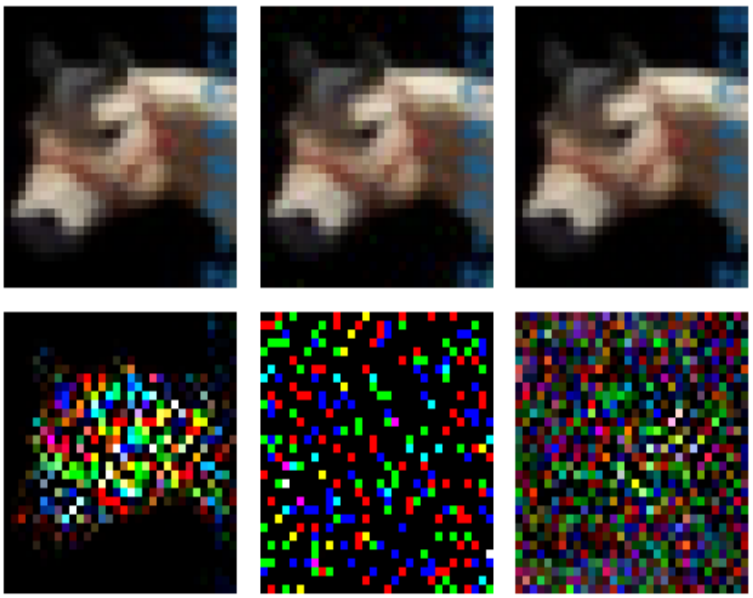
Although adversarial examples from different attack families appear to be indistinguishable, machines can learn and classify them with some subtle signatures. One question arises: What patterns does the classification model acquire to recognize the attack family and attack method? Since the differences in adversarial attacks are embedded in the perturbations, we propose to investigate the reasons behind the ease of identifying attack families by analyzing the perturbation patterns exhibited in various attacks. Visualization examples for representative L_2 attacks and L_∞ attacks are displayed in Fig. 4.2 and Fig. 4.3. More examples are in the Appendix.

4.4.1. L_2 Attacks. Different L_2 attacks modify the original images in different ways, resulting in different perturbation patterns; see Fig. 4.2, each subfigure lists adversarial examples from C&W, ZOO, and Boundary and corresponding amplified perturbations from left to right. It is obvious that the perturbations of the three attacks are different. The perturbations of the C&W attack seem to focus on the location of the object. ZOO introduces large perturbations for some pixels. The perturbations of the Boundary attack are relatively smaller and all over the place. In the following sections, we study the characteristics of C&W, ZOO, and Boundary and discuss why they generate perturbations of different patterns.

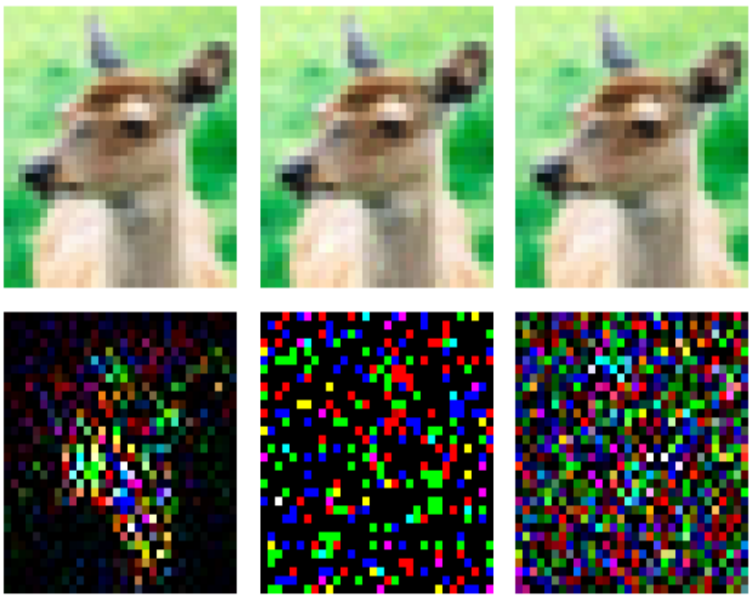
4.4.1.1. *C&W Attack.* C&W attack is one of the strongest gradient-based attacks to date. It can perform targeted and untargeted attacks with L_2 or L_∞ metric. Although L_∞ norm is feasible, L_2 norm is widely used in C&W attacks and can be formulated as the following regularized optimization problem:

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}\in[0,1]^n}{\operatorname{argmin}}\{\|\boldsymbol{x}-\boldsymbol{x_0}\|_2^2 + cg(\boldsymbol{x})\}. \tag{4.1}$$

The first term $\|\boldsymbol{x}-\boldsymbol{x_0}\|_2^2$ enforces a slight distortion to the original input $\boldsymbol{x_0}$ and the second term $g(\boldsymbol{x})$ is a loss function that measures how successful the attack is. The parameter $c > 0$ controls the trade-off between distortion and attack success.

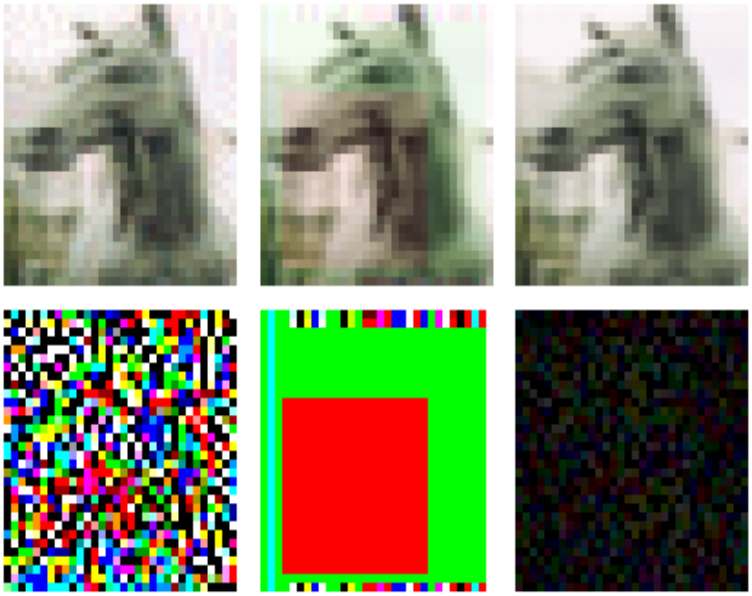


(a) Horse

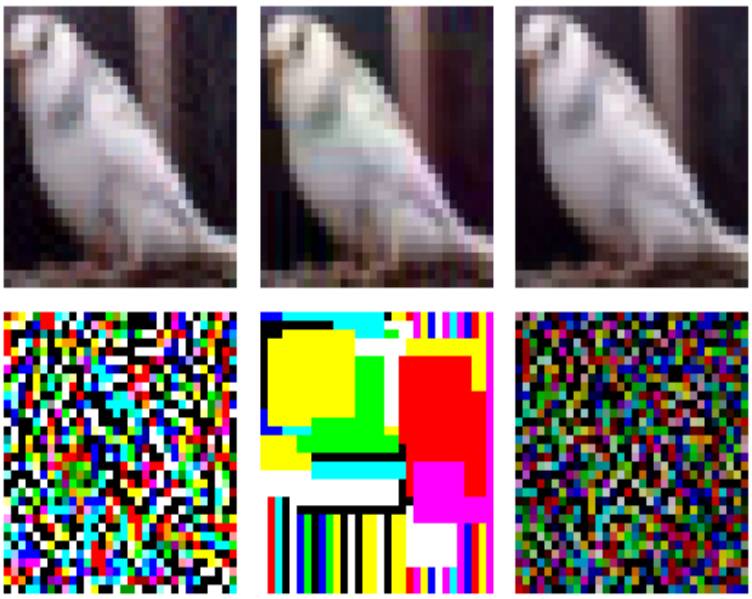


(b) Deer

FIGURE 4.2. Visualization examples for C&W, ZOO, and Boundary are displayed in each subfigure, sampled from CIFAR10. From left to right, the first row shows the adversarial image generated by C&W, ZOO, and Boundary, and the second row shows corresponding amplified perturbations. Though adversarial examples are indistinguishable, perturbations show different patterns: C&W's perturbations focus on the main object; ZOO introduces scattered bright per-pixel perturbations; Boundary's perturbations are more uniform across the image.



(a) Horse



(b) Bird

FIGURE 4.3. Visualization examples for PGD, Square, and HopSkipJump are displayed in each subfigure, sampled from the CIFAR10 data set. From left to right, the first row shows the adversarial image generated by PGD, Square, and HopSkipJump, and the second row shows corresponding amplified perturbations. PGD and HSJ have cluttered perturbation patterns, but HSJ is darker due to smaller perturbations. Square's perturbations consist of vertical strips covered by square-shaped regions, though vertical strips may not be obvious since too many squares cover them.

Compared to the other two attacks, it seems that the perturbations of C&W concentrate on the object, see Fig. 4.2. To verify if this observation is true, we draw a bounding box of the horse in Fig. 4.2 and compute the proportion of L_2 perturbations inside the box for all three attacks, see Fig. 4.4: the proportion of perturbation inside the bounding box for C&W is 96.40%, while for ZOO and Boundary, the proportions are 69.25% and 79.51% respectively.

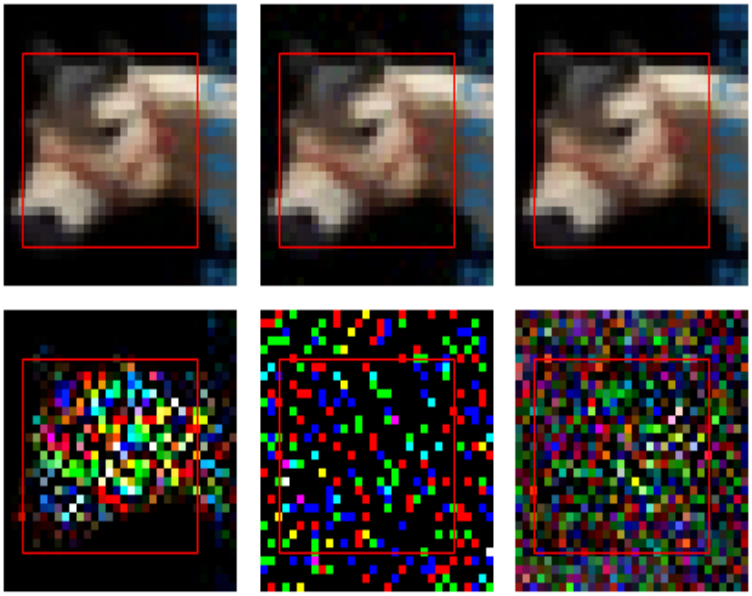


FIGURE 4.4. The proportion of perturbations inside the bounding box for C&W, ZOO, and Boundary are 96.40%, 69.25%, and 79.51% respectively, from left to right.

To verify if this pattern is true for most cases, we randomly sample five images with success across three attacks from each class of CIFAR10 and draw bounding boxes for all 50 images per attack to calculate the proportions of perturbations inside bounding boxes. The proportion is calculated per sampled image for each attack. Fig. 4.5 shows the histograms of in-box perturbation proportion for each attack. It is evident that C&W has the most left-skewed distribution, indicating that C&W focuses on perturbing the main object in the image.

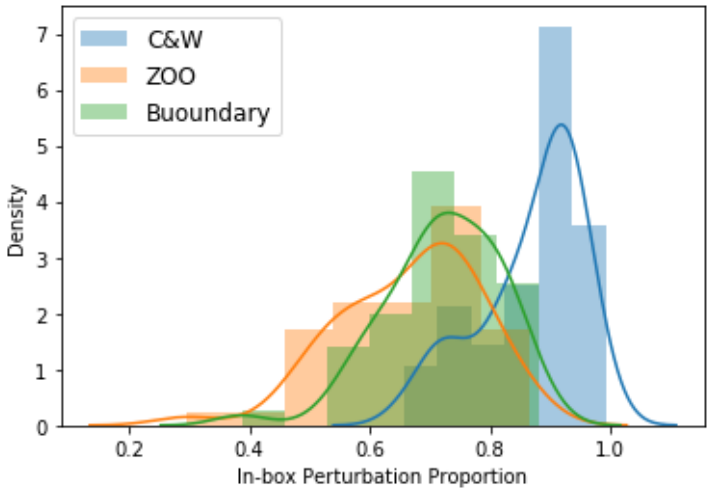


FIGURE 4.5. In-box perturbation proportion histograms for C&W, ZOO, and Boundary. C&W's distribution is most left-skewed, indicating C&W focuses on attacking the main object.

Two reasons might explain why C&W attacks the object: 1) C&W has access to the true gradients; 2) C&W method starts attacking from the original image. Gradients w.r.t. the input indicates the important areas in the input image and usually concentrate on the objects because the victim model is trained to do object classification. Therefore, it is expected to see C&W focus on modifying the object. Besides, the initial point of the optimization process is the original image, which excludes the possibility of unnecessary perturbations outside the object area.

To support the above hypothesis, we compare C&W with its two variants: estimated-gradient C&W and random-start C&W. Instead of using true gradients, estimated-gradient C&W uses gradients estimated by Natural Evolution Strategy [**Wierstra et al., 2014**], which was also used by Ilyas *et al.* [**Ilyas et al., 2018**] to do score-based attack. Random-start C&W starts the attack process with a random adversarial point instead of the original image. The random adversarial point is a random noise image that is not classified into the class of the original image. The point is already misclassified but not close to the original image.

We generate adversarial images with the original C&W and its two variants, then train a VGG16-based model to classify the three types of adversarial images. The classification accuracy reaches 96.03%, indicating that the three types of adversarial attacks are significantly different. Therefore, both gradients and random start affect the patterns of the C&W perturbations.

Fig. 4.6 lists the adversarial examples and perturbations of C&W, estimated-gradient C&W, and random-start C&W from left to right. The perturbations of estimated-gradient C&W still roughly focus on the object area but are less accurate than those of the original C&W. Also, the overall perturbations are larger: with estimated gradients, it cannot converge to the same level as C&W, resulting in a larger distortion level. With a random adversarial start, C&W gets noisier in the background, even though many perturbations are in the object area. In conclusion, C&W's perturbations focusing on the object area come from two factors: starting from original images and accurate gradients. See more examples in Appendix B.1.

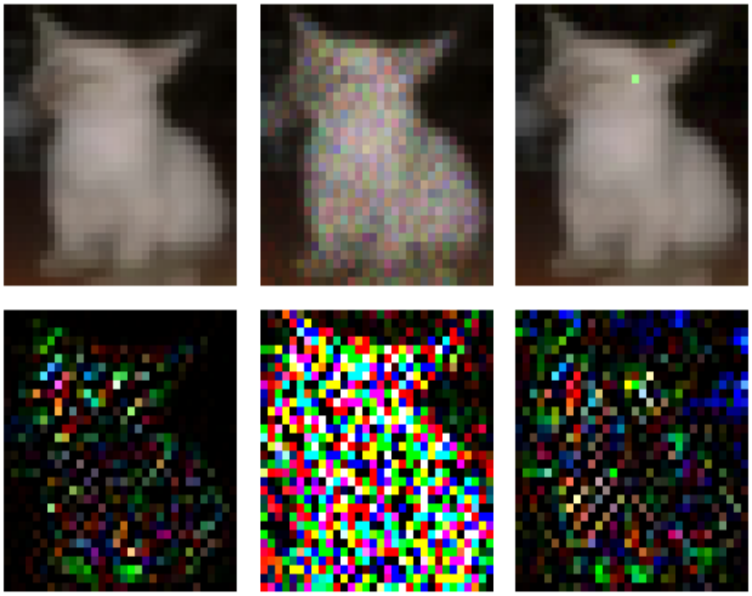


FIGURE 4.6. From left to right, a cat image is attacked by C&W, estimated-gradient C&W, and random-start C&W. Even though the perturbations of estimated-gradient C&W and random-start C&W also roughly focus on the object area, it is not as obvious as in the perturbations of the original C&W.

4.4.1.2. *ZOO Attack.* Zeroth Order Optimization Based Attack (ZOO) uses the finite difference method to approximate the gradients of the loss with respect to the input. The objective function is the same as that of C&W attack but using coordinate descent with estimated gradient:

$$\frac{\partial f}{\partial \boldsymbol{x_i}} \approx \frac{f(\boldsymbol{x} + h\boldsymbol{e_i}) - f(\boldsymbol{x} - h\boldsymbol{e_i})}{2h}, \tag{4.2}$$

where h is a small constant, $\boldsymbol{e_i}$ is a standard basis vector with a single nonzero entry with value 1 as the i-th element, and i ranges from 1 to the input dimension. That is, ZOO is another variant of C&W but with estimated gradient and coordinate descent.

From Fig. 4.2, we can see that ZOO's perturbations are made of a few bright pixels, which is expected as it uses coordinate descent to optimize each coordinate iteratively. Unlike gradient descent, that updates all coordinates at once, coordinate descent updates the coordinates by mini-batch. The nature of coordinate descent can lead to ZOO's perturbation pattern. To show the effect of coordinate descent on perturbation patterns, we compare ZOO with the estimated gradient C&W. The difference between them is the optimization method: ZOO uses coordinate descent while estimated-gradient C&W uses gradient descent, but both methods need to estimate the gradient. A VGG16-based binary classifier achieves 97.62% accuracy in classifying the adversarial examples generated by the two methods, implying that different optimization methods will result in different perturbation patterns. Fig. 4.7 shows the adversarial examples and amplified perturbations of ZOO and estimated-gradient C&W. More examples are available in Appendix B.2. Compared to the estimated-gradient C&W, ZOO has more spread perturbations because of the optimization method. In Appendix 4.4.1.1, we verified that the estimated gradient makes the perturbations larger and less accurate by comparing estimated-gradient C&W with the original C&W. This also helps explain why the perturbations of ZOO are so prominent and scattered. Therefore, coordinate descent and the estimated gradient together lead to ZOO's prominent scattered pixel-level perturbation pattern.

4.4.1.3. *Boundary Attack.* Boundary attack starts with a random adversarial point from a different class, then seeks to minimize the perturbations by randomly walking on the boundary of two classes while remaining adversarial. Compared to C&W, the Boundary attack does not start from the original image and has no access to the gradient information. From Fig. 4.2, we noticed that the Boundary attack's perturbations distribute over the entire image compared to C&W and ZOO. In fact, we verified in Section 4.4.1.1 that starting from an adversarial point instead of the original image will spread the perturbations, and the gradient information is the key to an accurate attack on the object. This explanation applies to the perturbation patterns of Boundary attacks as well. Fig. 4.8 shows adversarial examples and perturbations of C&W, random-start C&W, and Boundary. Compared to C&W, the other two attacks show noisy and spread perturbations, even

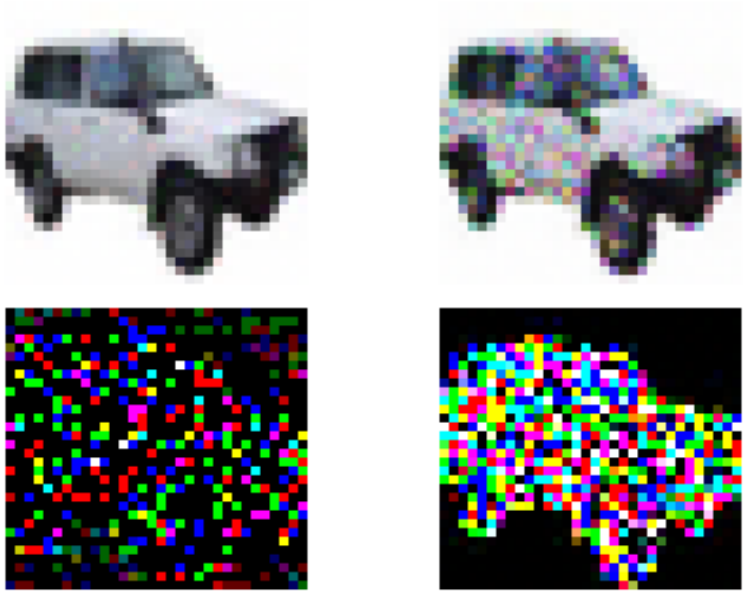


FIGURE 4.7. An automobile image is attacked by ZOO(left) and estimated-gradient C&W(right). The first row contains adversarial examples, and the second row contains amplified perturbations. ZOO's amplified perturbations are more spread due to coordinate descent.

though random-start C&W has most perturbations focused on the frog area. More examples are available in Appendix B.3.

Besides, unlike random-start C&W, Boundary's updating procedure relies on a random walk instead of gradients, which draws random perturbation from a proposal distribution at each iteration. Hence, Boundary's perturbations are more blurry than the random-start C&W. A VGG16-based three-class model achieves 88.12% accuracy in classifying the three attacks, indicating that the differences are obvious and easy to detect. Therefore, both random adversarial start and lack of gradient information contribute to Boundary's specific perturbation patterns.

4.4.2. L_∞ Attacks.

L_∞ attacks in different attack families show different perturbation patterns as well. In this section, we study the L_∞-norm version of PGD (gradient-based), Square (score-based), and HopSkipJump (decision-based). In our experiments, perturbations are bounded by 0.03. Fig. 4.3 shows adversarial examples and perturbation patterns of PGD, Square, and HopSkipJump (HSJ). The perturbations of Square consist of vertical strips covered by square-shaped regions. Both PGD and HSJ have

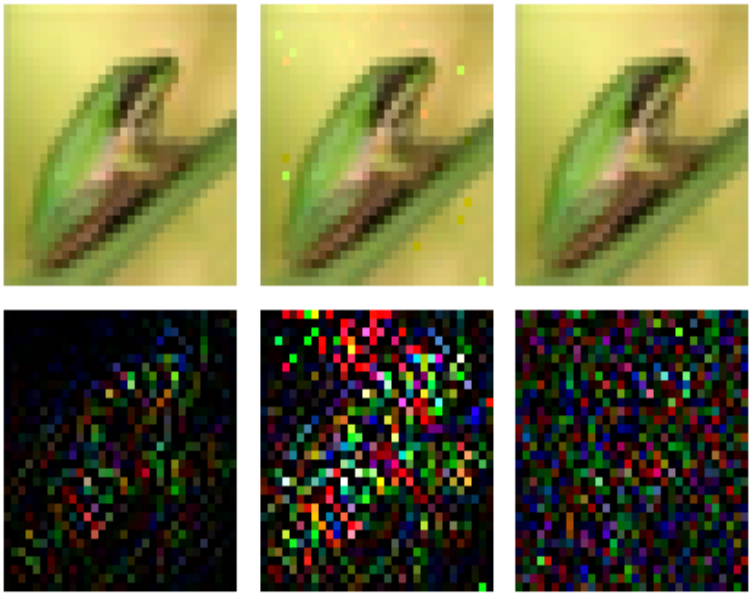


FIGURE 4.8. A frog image is attacked by C&W, random-start C&W and Boundary in turn. From left to right, the perturbations are getting noisier, and the frog outline is blurring. It indicates both random start and random walk iteration without gradient information contribute to Boundary's noisy perturbations.

clutter perturbation patterns, but the perturbations of HSJ are darker. In the following sections, we discuss the characteristics of Square first and then compare PGD and HSJ.

4.4.2.1. *Square Attack.* The Square attack is score-based, but unlike other score-based attacks, such as ZOO or NES, it does not estimate the gradients when generating adversarial examples. Instead, it adopts an iterative randomized search scheme: at each iteration, a local square update is chosen at random locations and projected to the input space, then this update is added to the current iteration if the objective function improves. This explains the square-shaped regions in the perturbation pattern. As for initialization, Square uses vertical stripes of width 1, where the color of each stripe is randomly and uniformly sampled. In some cases, it takes many iterations to generate a successful adversarial example, so the stripes are nearly covered by squares.

4.4.2.2. *PGD and HopSkipJump Attack.* Projected-Gradient Descent Attack (PGD) crafts adversarial examples by solving the constraint optimization problem iteratively with projected gradient descent, widely used with L_∞ norm. It can be formulated as

$$\boldsymbol{x}^* = \underset{\|\boldsymbol{x}-\boldsymbol{x}^*\|_\infty<\epsilon}{\operatorname{argmax}} L(\boldsymbol{\theta}, \boldsymbol{x}, y), \tag{4.3}$$

where L is the loss function used to train the victim model, $\boldsymbol{\theta}$ is a fixed model parameter, and $(\boldsymbol{x}, y)$ is the input pair of the original image and label. It uses a multi-step iteration scheme: at each iteration, take a small step α according to the sign of the gradient and clip the result to the ϵ-ball of the original input:

$$\boldsymbol{x}^{t+1} = \Pi_\epsilon\{\boldsymbol{x}^t + \alpha \cdot \text{sign}\left(\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \boldsymbol{x}^t, y)\right), \boldsymbol{x}_0\}. \tag{4.4}$$

HopSkipJump attack finds optimal adversarial examples by iterative procedure and gradient estimate. Like Boundary, it starts from an adversarial point of a different class. For each iteration, it first moves towards the boundary of the two classes (true class vs. a wrong class) through binary search, then updates the step size along the estimated gradient direction through geometric progression until perturbation is successful, and lastly projects the perturbed sample back to the boundary again.

Though PGD and HSJ belong to different attack families, both have cluttered perturbations, except that the perturbations of HSJ are dimmer due to smaller perturbations. Though both methods are L_∞-norm based and bounded by 0.03, HSJ has perturbations of different scales ranging from -0.03 to 0.03, while PGD has more extremely perturbed pixels with a perturbation value of 0.03. From Fig. 4.9, we can see that the histograms of the perturbations of PGD and HSJ are very different. The histogram of PGD perturbations is like a bar plot because it updates depending on the sign of the gradients with a fixed step-size α, which explains the discrete bars in the distribution of PGD's perturbations. While HSJ does not use a fixed step size to update, it does not have such a pattern. We also test if the perturbations of PGD and HSJ focus on the object area. The same bounding box method in Section 4.4.1.1 is used to calculate the proportion of significant perturbations inside the box for both attacks. A significant perturbation is defined as a perturbation whose absolute value is larger than the 90% quantile. In Fig. 4.10, we can see the in-box significant perturbation proportion histogram. HSJ's distribution is more left-skewed than PGD's; the average in-box significant perturbation proportions of PGD and HSJ are 50.37% and 60.38%, respectively. Therefore, even though PGD has access to the true gradient information, HSJ has more significant perturbations in the object area.

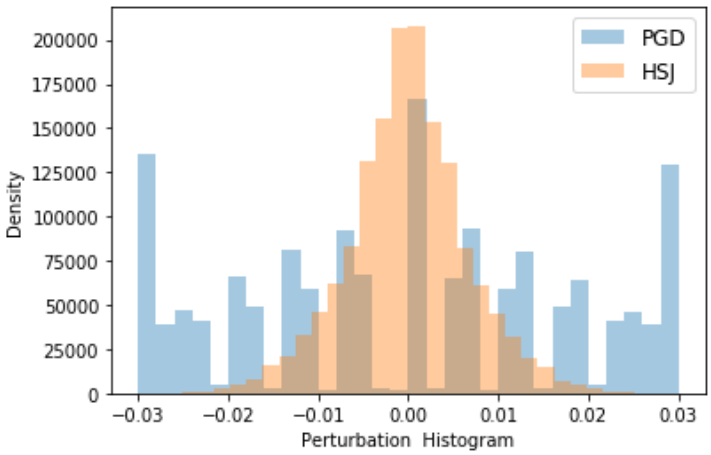


FIGURE 4.9. Histogram of perturbation values of PGD and HSJ. PGD has a bar-plot-like perturbation distribution because it uses a fixed step size to update, while HSJ has a normal-like perturbation distribution.

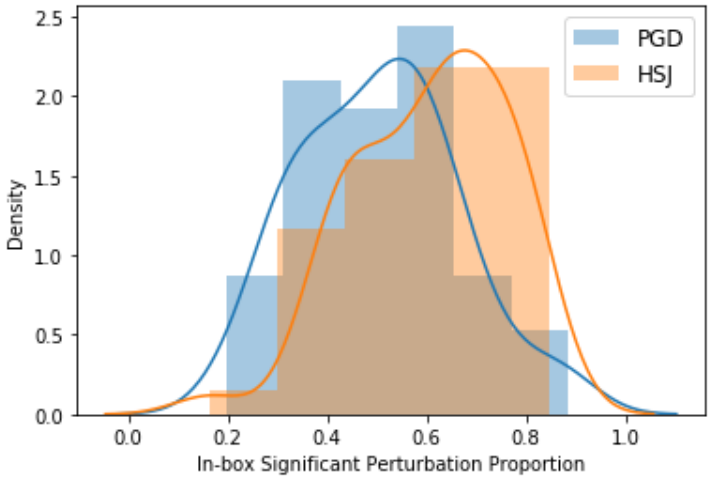


FIGURE 4.10. Histogram of In-box significant perturbation proportion of PGD and HSJ. HSJ's distribution is more left-skewed than PGD, indicating it has more significant perturbations in the object area.

4.5. Concluding Remarks

Our findings demonstrate attack methods from different attack families (gradient-based, score-based, decision-based) possess different characteristics. Given adversarial examples, the machine can learn such characteristics to identify which attack family they belong to. Further studies show that even attacks from the same family can be different. We systematically study the properties of the perturbation patterns of different attacks and explore where their differences come from. We

hope that our work can shed light on a deeper understanding of adversarial attacks and help with the reverse engineering of adversarial attacks.

APPENDIX A

Appendix for Chapter 3

This appendix provides statistical consistency proof for Theorem 1.

Consider K homogeneous networks with N nodes, represented by binary adjacency matrix $\{\boldsymbol{A}_k | k = 1, ..., K\}$. Assume the same presence of communities among all N nodes for each k, which can be modeled by SBM presented in the previous section with parameter $\boldsymbol{\psi} = (\boldsymbol{c}, \boldsymbol{\pi})$, $\boldsymbol{c}$ is community assignment vector and $\boldsymbol{\pi}$ is linkage probability.

For a specified class assignment $\boldsymbol{c}$, define $n_q(\boldsymbol{c}) = \#\{i | c_i = q\}$ as the number of nodes in class q. Subsequently, the quantity of potential pairs within/between each block is denoted as

$$N_{ql}(\boldsymbol{c}) = \begin{cases} n_q n_l & q \neq l \\ n_q(n_q - 1)/2 & q = l \end{cases},$$

the number of the observed within/between each block is denoted as

$$E_{k,ql}(\boldsymbol{c}) = \begin{cases} \sum_{c_i=q} \sum_{c_j=l} \boldsymbol{A}_{k,ij} & q \neq l \\ \sum_{c_i=q} \sum_{c_j=l} \boldsymbol{A}_{k,ij}/2 & q = l \end{cases},$$

then for each $1 \leq k \leq K$, the log-likelihood function for $\boldsymbol{\psi} = (\boldsymbol{c}, \boldsymbol{\pi})$ is

$$\begin{aligned} l_k((\boldsymbol{c}, \boldsymbol{\pi}); \boldsymbol{A}_k) &= \sum_{i<j} \left[\boldsymbol{A}_{k,ij} \log_2 \Omega_{k,ij} + (1 - \boldsymbol{A}_{k,ij}) \log_2(1 - \Omega_{k,ij}) \right] \\ &= \sum_{q \leq l} \left[E_{k,ql}(\boldsymbol{c}) \log_2 \boldsymbol{\pi}_{ql} + (N_{ql}(\boldsymbol{c}) - E_{k,ql}(\boldsymbol{c})) \log_2(1 - \boldsymbol{\pi}_{ql}) \right]. \end{aligned}$$

Recall that $\boldsymbol{\psi} = (\boldsymbol{c}, \boldsymbol{\pi})$, and $\mathcal{M}$ is the set of all possible $\boldsymbol{\psi}$, then the log-likelihood for the K observations can be written as:

$$\mathcal{L}_K(\boldsymbol{\psi}; \boldsymbol{A}) = \sum_{k=1}^{K} l_k((\boldsymbol{c}, \boldsymbol{\pi}); \boldsymbol{A}_k).$$

Then vector $\boldsymbol{\psi} = (\boldsymbol{c}, \boldsymbol{\pi})$ can specify a model for this sequence of networks, and the MDL can be written as

$$MDL(\boldsymbol{\psi}; \boldsymbol{A}) = (N+1)\log_2 Q + \sum_{q \le l} \frac{1}{2}\log_2(N_{ql}(\boldsymbol{c})) - \mathcal{L}_K(\boldsymbol{\psi}; \boldsymbol{A}). \tag{A.1}$$

From Theorem 1, then the MDL-based estimate is given by

$$\hat{\boldsymbol{\psi}} = \arg\min_{\boldsymbol{\psi} \in \mathcal{M}} \frac{1}{K} MDL(\boldsymbol{\psi})$$

where $\mathcal{M}$ is the set of all possible values of parameter $\boldsymbol{\psi}$. For any $\hat{\boldsymbol{\psi}} = (\hat{\boldsymbol{c}}, \hat{\boldsymbol{\pi}})$, $\hat{\boldsymbol{\pi}}$ the MLE given K observations' log likelihood $L_K\left((\hat{\boldsymbol{c}}, \boldsymbol{\pi}); \hat{\boldsymbol{A}}\right)$ with $\hat{\boldsymbol{c}}$ denotes the estimated community assignment, $\hat{\boldsymbol{A}}$ denotes the estimated sequence of networks, and $\boldsymbol{\Pi}(\hat{\boldsymbol{c}})$ denotes the parameter space of $\boldsymbol{\pi}$ given $\hat{\boldsymbol{c}}$, that is,

$$\hat{\boldsymbol{\pi}} = \arg\max_{\boldsymbol{\pi} \in \boldsymbol{\Pi}(\hat{\boldsymbol{c}})} L_K\left((\hat{\boldsymbol{c}}, \boldsymbol{\pi}); \hat{\boldsymbol{A}}\right).$$

Then we have the estimated community assignment $\hat{\boldsymbol{c}}$ must be bigger than the true community assignment $\boldsymbol{c}^0$, and there exists a function $g : \hat{\boldsymbol{c}}_i \to \boldsymbol{c}_i^o$, such that

$$\hat{\boldsymbol{\pi}}_{ql} \xrightarrow{a.s.} \boldsymbol{\pi}^o_{g(q)g(l)}.$$

We list the necessary regularity conditions for the conditional log-likelihood function for the standard properties of maximum likelihood estimation, as well as the proposition and lemma for proof of Theorem 1.

Assumption $1(v)$: For any fixed $\boldsymbol{c}$, there exists a $\epsilon > 0$ such that,

$$\sup_{\boldsymbol{\pi} \in \Pi(\boldsymbol{c})} E \mid l_k\left((\boldsymbol{c}, \boldsymbol{\pi}); \boldsymbol{A}_k\right) \mid^{v+\epsilon} < \infty,$$

$$\sup_{\boldsymbol{\pi} \in \Pi(\boldsymbol{c})} E \mid l'_k\left((\boldsymbol{c}, \boldsymbol{\pi}); \boldsymbol{A}_k\right) \mid^{v+\epsilon} < \infty,$$

$$\sup_{\boldsymbol{\pi} \in \Pi(\boldsymbol{c})} E \mid l''_k\left((\boldsymbol{c}, \boldsymbol{\pi}); \boldsymbol{A}_k\right) \mid^{v+\epsilon} < \infty.$$

Note that Assumption 1(1) refer to Assumption 1 with $v = 1$.

Assumption 2 : For any fixed $\boldsymbol{c}$,

$$\sup_{\boldsymbol{\pi}\in\Pi(\boldsymbol{c})} \mid \frac{1}{K}\mathcal{L}_K((\boldsymbol{c},\boldsymbol{\pi};\boldsymbol{A}) - L\left((\boldsymbol{c},\boldsymbol{\pi})\right) \mid \xrightarrow{a.s.} 0,$$

$$\sup_{\boldsymbol{\pi}\in\Pi(\boldsymbol{c})} \mid \frac{1}{K}\mathcal{L}'_K((\boldsymbol{c},\boldsymbol{\pi};\boldsymbol{A}) - L'\left((\boldsymbol{c},\boldsymbol{\pi})\right) \mid \xrightarrow{a.s.} 0,$$

$$\sup_{\boldsymbol{\pi}\in\Pi(\boldsymbol{c})} \mid \frac{1}{K}\mathcal{L}''_K((\boldsymbol{c},\boldsymbol{\pi};\boldsymbol{A}) - L''\left((\boldsymbol{c},\boldsymbol{\pi})\right) \mid \xrightarrow{a.s.} 0,$$

where

$$L\left((\boldsymbol{c},\boldsymbol{\pi})\right) := E\left(l_k\left((\boldsymbol{c},\boldsymbol{\pi})\,;\boldsymbol{A}_k\right)\right),$$

$$L'\left((\boldsymbol{c},\boldsymbol{\pi})\right) := E\left(l'_k\left((\boldsymbol{c},\boldsymbol{\pi})\,;\boldsymbol{A}_k\right)\right),$$

$$L''\left((\boldsymbol{c},\boldsymbol{\pi})\right) := E\left(l''_k\left((\boldsymbol{c},\boldsymbol{\pi})\,;\boldsymbol{A}_k\right)\right).$$

To lighten notations, we skip some k's in the following proposition.

PROPOSITION A.0.1. *The true model $\boldsymbol{\psi}^o \in \mathcal{M}$ satisfies*

$$\boldsymbol{\psi}^o = \arg\max_{\boldsymbol{\psi}\in\mathcal{M}} E\left(l(\boldsymbol{\psi};\boldsymbol{A})\right).$$

Furthermore, $\boldsymbol{\psi}^o$ is uniquely identifiable, that is, if there exists a $\boldsymbol{\pi}^*$ such that $l\left(\left(\boldsymbol{c}^0,\boldsymbol{\pi}^0\right);\boldsymbol{A}\right) = l\left(\left(\boldsymbol{c}^0,\boldsymbol{\pi}^*\right);\boldsymbol{A}\right)$ almost everywhere for $\boldsymbol{A}$, then $\boldsymbol{\pi}^* = \boldsymbol{\pi}^0$. Additionally, suppose there exists another model $\boldsymbol{\psi}^b = (\boldsymbol{c}^b,\boldsymbol{\pi}^b)$ such that $l\left(\left(\boldsymbol{c}^b,\boldsymbol{\pi}^b\right);\boldsymbol{A}\right) = l\left(\left(\boldsymbol{c}^0,\boldsymbol{\pi}^0\right);\boldsymbol{A}\right)$ almost everywhere, then $\boldsymbol{c}^b$ must be a bigger model of $\boldsymbol{c}^0$. That is, there exists a function $g : \boldsymbol{c}^b_i \to \boldsymbol{c}^o_i$, such that $\boldsymbol{\pi}^b_{ql} \xrightarrow{a.s.} \boldsymbol{\pi}^o_{g(q)g(l)}$.

PROOF. Define $\overline{\boldsymbol{\pi}}_{ql}(\boldsymbol{c}) := \frac{1}{N_{ql}(\boldsymbol{c})}\sum_{i\neq j, c_i=q, c_j=l}\boldsymbol{\pi}^o_{c^o_i,c^o_j}$, where $\overline{\boldsymbol{\pi}}(\boldsymbol{c}^0) = \boldsymbol{\pi}^0$ is a special case. Let $\boldsymbol{\pi}^*$ be another link probability with $\boldsymbol{c}$ as the community assignment. Then,

$$\begin{aligned} E\left(l((\boldsymbol{c},\boldsymbol{\pi}^*);\boldsymbol{A})\right) &= E\Bigg(\sum_{i<j}\Big(\boldsymbol{A}_{ij}\log\left(\boldsymbol{\pi}^*_{c_i,c_j}\right) + (1-\boldsymbol{A}_{ij})\log\left(1-\boldsymbol{\pi}^*_{c_i,c_j}\right)\Big)\Bigg) \\ &= \sum_{q\leq l}\sum_{i\neq j, c_i=q, c_j=l}\Big(\boldsymbol{\pi}^o_{c^o_i,c^o_j}\log\left(\boldsymbol{\pi}^*_{c_i,c_j}\right) + (1-\boldsymbol{\pi}^o_{c^o_i,c^o_j})\log\left(1-\boldsymbol{\pi}^*_{c_i,c_j}\right)\Big) \qquad \text{(A.2)} \\ &= \sum_{q\leq l} N_{ql}(\boldsymbol{c})\Big(\overline{\boldsymbol{\pi}}_{ql}(\boldsymbol{c})\log\left(\boldsymbol{\pi}^*_{ql}\right) + (1-\overline{\boldsymbol{\pi}}_{ql}(\boldsymbol{c}))\log\left(1-\boldsymbol{\pi}^*_{ql}\right)\Big). \end{aligned}$$

Similarly, we have

$$E\left(l((\boldsymbol{c},\overline{\boldsymbol{\pi}}(\boldsymbol{c}));\boldsymbol{A})\right)=\sum_{q\leq l}N_{ql}(\boldsymbol{c})\Big(\overline{\boldsymbol{\pi}}_{ql}(\boldsymbol{c})\log\left(\overline{\boldsymbol{\pi}}_{ql}(\boldsymbol{c})\right)+(1-\overline{\boldsymbol{\pi}}_{ql}(\boldsymbol{c}))\log\left(1-\overline{\boldsymbol{\pi}}_{ql}(\boldsymbol{c})\right)\Big) \tag{A.3}$$

Combining above Equation A.2 and A.3, we have

(A.4)

$$\begin{aligned}E\left(l((\boldsymbol{c},\overline{\boldsymbol{\pi}}(\boldsymbol{c}));\boldsymbol{A})\right)-E\left(l((\boldsymbol{c},\boldsymbol{\pi}^{*});\boldsymbol{A})\right)&=\sum_{q\leq l}N_{ql}(\boldsymbol{c})\left(\overline{\boldsymbol{\pi}}_{ql}(\boldsymbol{c})\log\left(\frac{\overline{\boldsymbol{\pi}}_{ql}(\boldsymbol{c})}{\boldsymbol{\pi}^{*}_{ql}}\right)+(1-\overline{\boldsymbol{\pi}}_{ql}(\boldsymbol{c}))\log\left(\frac{1-\overline{\boldsymbol{\pi}}_{ql}(\boldsymbol{c})}{1-\boldsymbol{\pi}^{*}_{ql}}\right)\right)\\&=\sum_{q\leq l}N_{ql}(\boldsymbol{c})D_{KL}\left(\overline{\boldsymbol{\pi}}_{ql}(c)\|\boldsymbol{\pi}^{*}_{ql}\right)\\&\geq 0,\end{aligned}$$

where $D_{KL}\left(\overline{\boldsymbol{\pi}}_{ql}(c)\|\boldsymbol{\pi}^{*}_{ql}\right)$ is the Kullback–Leibler divergence of $Bernoulli(\overline{\boldsymbol{\pi}}_{ql}(\boldsymbol{c}))$ distribution from $Bernoulli(\boldsymbol{\pi}^{*}_{ql})$.

Additionally, according to Lemma 1 in [**Han et al., 2015**], we have the following result for $\boldsymbol{c}$ does not underestimate $\boldsymbol{c}^{0}$:

$$E\left(l((\boldsymbol{c}^{0},\overline{\boldsymbol{\pi}}(\boldsymbol{c}^{0}));\boldsymbol{A})\right)-E\left(l((\boldsymbol{c},\overline{\boldsymbol{\pi}}(\boldsymbol{c}));\boldsymbol{A})\right)\geq\frac{1}{2}\delta\min_{q}n_{q}(\boldsymbol{c}^{0}), \tag{A.5}$$

where $n_q(\boldsymbol{c}^0)$ denotes the number of nodes assigned to community q under $\boldsymbol{c}^0$ and

$$\delta=\min_{q,l}\max_{r}\sigma\left(\boldsymbol{\pi}^{o}_{qr}\right)+\sigma\left(\boldsymbol{\pi}^{o}_{lr}\right)-2\sigma\left(\frac{\boldsymbol{\pi}^{o}_{qr}+\boldsymbol{\pi}^{o}_{lr}}{2}\right),$$

with $\sigma(x):=x\log(x)+(1-x)\log(1-x)$.

Derived from A.4 and A.5, we have

$$E\left(l\left((\boldsymbol{c}^{o},\boldsymbol{\pi}^{o});\boldsymbol{A}\right)\right)-E\left(l\left((\boldsymbol{c},\boldsymbol{\pi});\boldsymbol{A}\right)\right)\geq\frac{1}{2}\delta\min_{q}n_{q}(\boldsymbol{c}^{0})$$

□

LEMMA 1. *Suppose the true community assignment vector $\boldsymbol{c}^0$ is specified for the K observations, then*

$$\hat{\boldsymbol{\pi}}_{K}-\boldsymbol{\pi}^{o}=O\left(\sqrt{\frac{\log\log(K)}{K}}\right)\ a.s.$$

When a specific community assignment $\boldsymbol{c}$ *is bigger than the ture* $\boldsymbol{c}$*, which means there exists a function* $g : \hat{\boldsymbol{c}}_i \to \boldsymbol{c}_i^o$*, then*

$$\hat{\boldsymbol{\pi}}_{K,ql} - \boldsymbol{\pi}^o_{g(q)g(l)} = O\left(\sqrt{\frac{\log\log(K)}{K}}\right) \ a.s.$$

PROOF. Please refer to Lemma 2 in [**Davis and Yau, 2013**] for the detailed proof. □

Now we are ready for the proof of Theorem 1.

PROOF. Let $\hat{\boldsymbol{\psi}} = (\hat{\boldsymbol{c}}, \hat{\boldsymbol{\pi}})$ be the estimate of the community assignment and linkage probability for the K observations. Since $\mathcal{M}$ is a finite set, without loss of generality, we can assume that $\hat{\boldsymbol{c}}$ converges to $\boldsymbol{c}^*$. Similarly, $\boldsymbol{\Pi} = \boldsymbol{\Pi}(\hat{\boldsymbol{c}})$ is compact for any $\hat{\boldsymbol{c}}$, we assume that $\hat{\boldsymbol{\pi}}$ converges to $\boldsymbol{\pi}^*$. For a sufficiently large K,

$$\begin{aligned}\frac{1}{K}MDL(\hat{\boldsymbol{\psi}}; \boldsymbol{A}) &= h_K - \frac{1}{K}\mathcal{L}_K((\hat{\boldsymbol{c}}, \hat{\boldsymbol{\pi}}); \boldsymbol{A}) \\ &= h_K - L((\boldsymbol{c}^*, \boldsymbol{\pi}^*); \boldsymbol{A}),\end{aligned}$$

where h_K is deterministic with order $O(\log(K)/K)$.
If $\boldsymbol{\psi}^*$ underestimates $\boldsymbol{\psi}^0$, then according to Proposition A.0.1, we have

$$E\left(l\left((\boldsymbol{c}^o, \boldsymbol{\pi}^o); \boldsymbol{A}\right)\right) - E\left(l\left((\boldsymbol{c}^*, \boldsymbol{\pi}^*); \boldsymbol{A}\right)\right) > 0 \tag{A.6}$$

According to definitions in Assumption 2, it is equivalent to

$$L(\boldsymbol{\psi}^0; \boldsymbol{A}) - L(\boldsymbol{\psi}^*; \boldsymbol{A}) > 0. \tag{A.7}$$

Then for sufficiently large K,

$$\begin{aligned}\frac{1}{K}MDL(\hat{\boldsymbol{\psi}}; \boldsymbol{A}) &= h_K - \frac{1}{K}\mathcal{L}_K(\hat{\boldsymbol{\psi}}; \boldsymbol{A}) \\ &= h_K - L(\boldsymbol{\psi}^*; \boldsymbol{A}) \\ &> h_K - L(\boldsymbol{\psi}^0; \boldsymbol{A}) \\ &= \frac{1}{K}MDL(\boldsymbol{\psi}^0; \boldsymbol{A}) \\ &\geq \frac{1}{K}MDL(\hat{\boldsymbol{\psi}}; \boldsymbol{A}),\end{aligned} \tag{A.8}$$

which is a contradiction, therefore $\boldsymbol{\psi}^*$ must be a bigger model of $\boldsymbol{\psi}^0$.

Furthermore, according to Lemma 2, we have the consistency result of $\hat{\boldsymbol{\pi}}$, which completes the proof of Theorem 1. □

APPENDIX B

Appendix for Chapter 4

This supplementary material provides more illustrative examples and details of those classification experiments. As mentioned in Section 4.4, Fig.B.1 provides extra adversarial examples and corresponding perturbation patterns for C&W, ZOO, and Boundary, and Fig. B.2 provides extra adversarial examples and corresponding perturbation patterns for PGD, Square, and HopSkipJump.

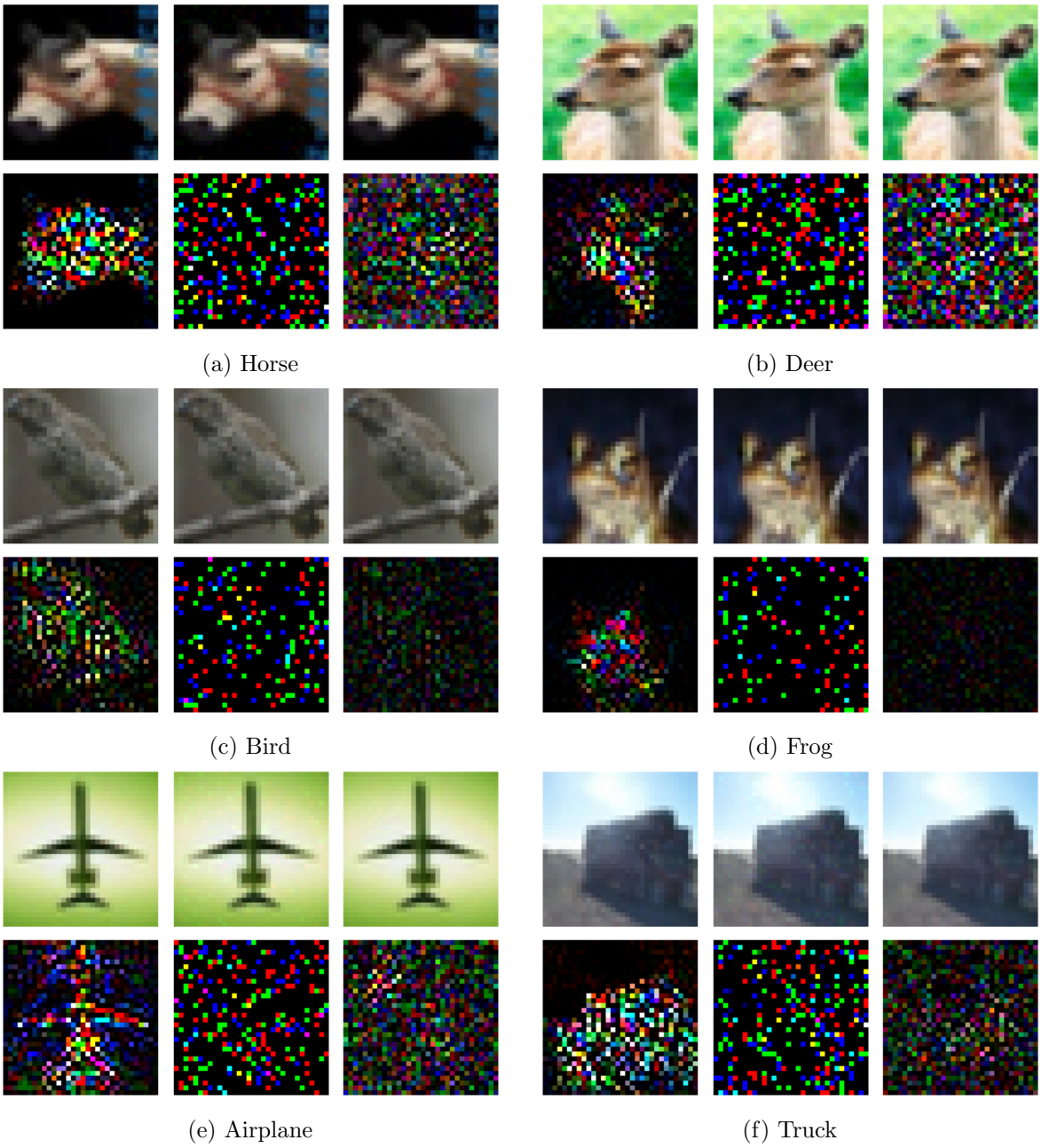
B.1. Supplementary Examples and Experiment in Section 4.4.1.1

In Section 4.4.1.1, we proposed that the plausible reasons for C&W attacking the main object are true gradients and starting the attack process from the original image. To verify the idea, we generate adversarial images based on two variants of C&W: the estimated-gradient C&W uses estimated gradients from NES instead of the true gradients, and random-start C&W generates adversarial images starting from a random adversarial image instead of the original image. More examples are displayed in Fig. B.3.

Select those images that have been successfully attacked by all three attacks and split them into training and test sets of size 1764 and 756, respectively. Train a VGG16-based classifier to evaluate whether there's a difference among them. Accuracy reaches 96.03%. Table B.1 records the confusion matrix of this classification task; we can see that both variants can be easily distinguished from C&W. This result further explains that the true gradients and original start affect C&W's performance.

TABLE B.1. Confusion Matrix for C&W, estimated-gradient C&W and random-start C&W

		Predicted		
		C&W	estimated-gradient C&W	random-start C&W
Actual	C&W	247	0	5
	estimated-gradient C&W	2	249	1
	random-start C&W	22	0	230

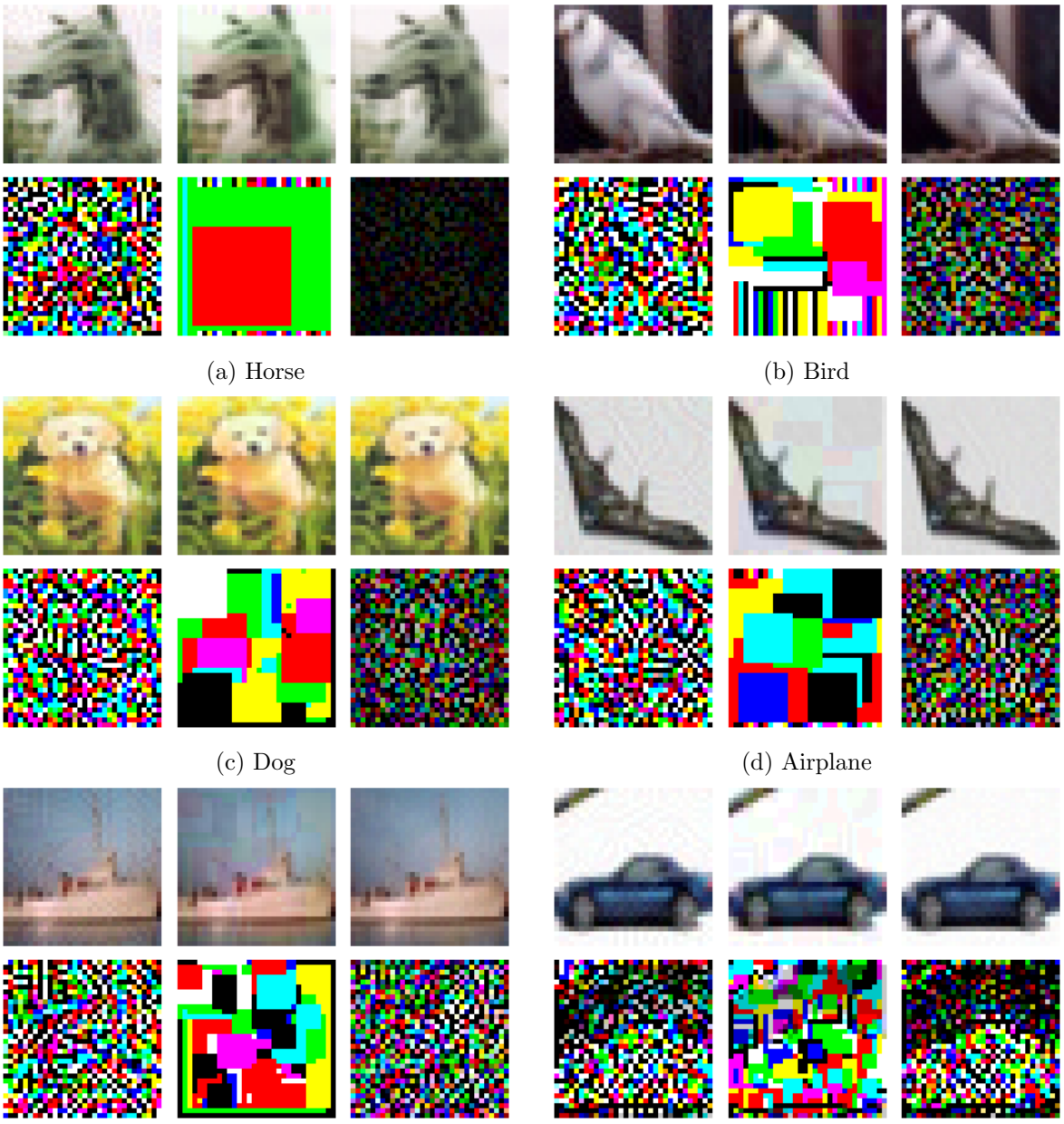


(a) Horse (b) Deer

(c) Bird (d) Frog

(e) Airplane (f) Truck

FIGURE B.1. Visualization examples for C&W, ZOO, and Boundary are displayed in each subfigure, sampled from CIFAR10. From left to right, the first row shows the adversarial image generated by C&W, ZOO, and Boundary, and the second row shows corresponding amplified perturbations. Though adversarial examples are indistinguishable, perturbations show different patterns: C&W's perturbations focus on the main object; ZOO introduces scattered bright per-pixel perturbations; Boundary's perturbations are more uniform across the image.

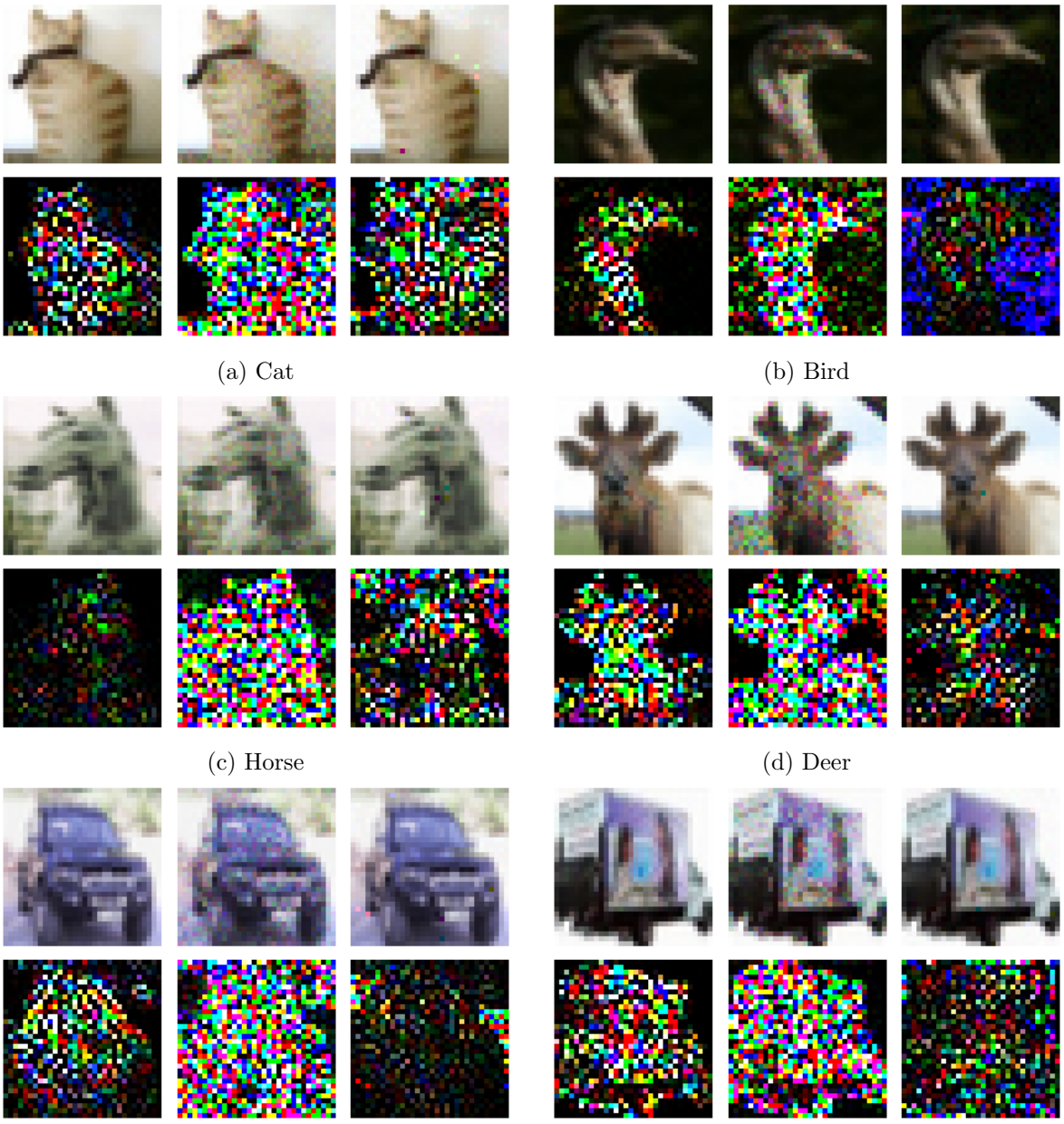


(a) Horse (b) Bird

(c) Dog (d) Airplane

(e) Ship (f) Automobile

FIGURE B.2. Visualization examples for PGD, Square, and HopSkipJump are displayed in each subfigure, sampled from the CIFAR10 data set. From left to right, the first row shows the adversarial image generated by PGD, Square, and HopSkipJump, and the second row shows corresponding amplified perturbations. PGD and HSJ have cluttered perturbation patterns, but HSJ is darker due to smaller perturbations. Square's perturbations consist of vertical strips covered by square-shaped regions, though vertical strips may not be obvious since it's covered by too many squares.



(a) Cat (b) Bird

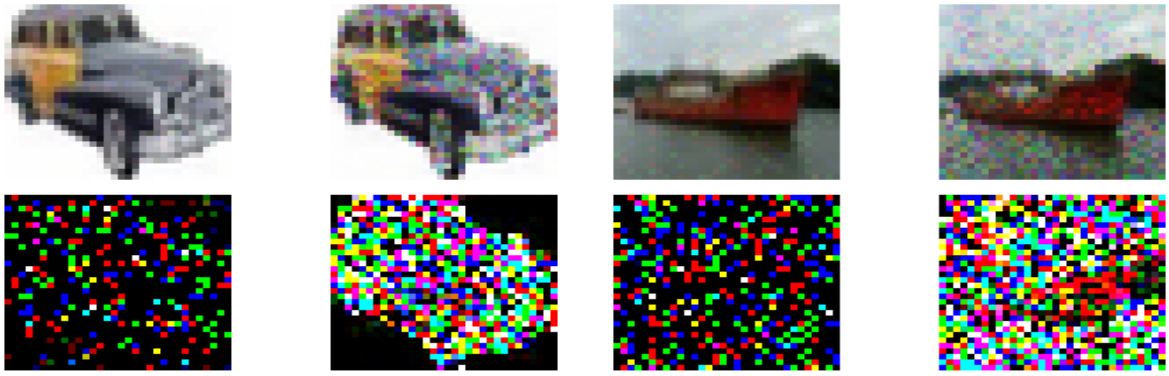
(c) Horse (d) Deer

(e) Automobile (f) Truck

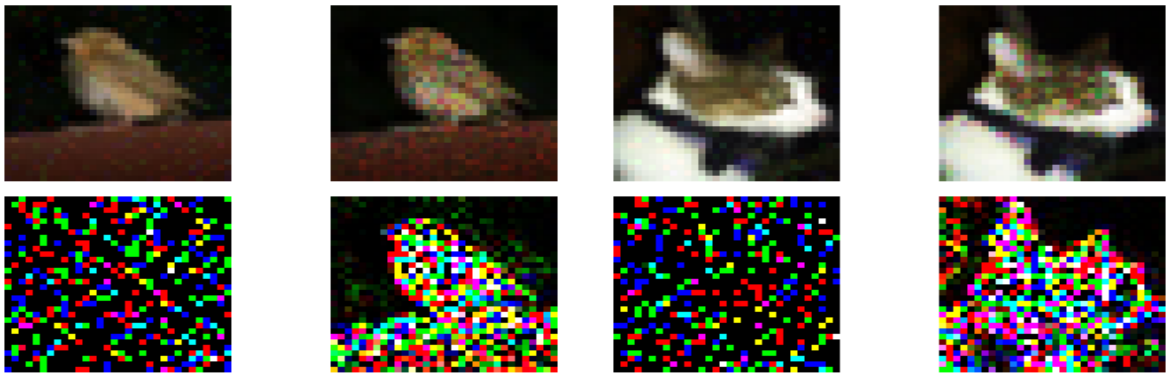
FIGURE B.3. Each subfigure displays adversarial images and perturbations of C&W, estimated-gradient C&W, random-start C&W from left to right, sampled from CIFAR10 dataset.

B.2. Supplementary Examples and Experiment in Section 4.4.1.2

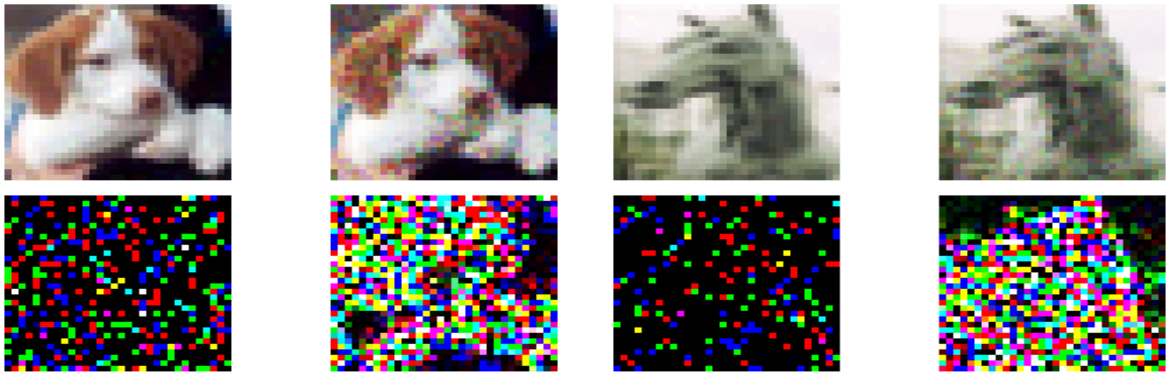
ZOO is another variant of C&W with estimated gradients and coordinate descent. In Section 4.4.1.2, to evaluate the optimization method's effect on perturbation patterns, we compare ZOO with estimated-gradient C&W; more examples are displayed in Fig. B.4.



(a) Automobile (b) Ship



(c) Bird (d) Cat



(e) Dog (f) Horse

FIGURE B.4. Additional visualization examples for ZOO and estimated-gradient C&W are displayed in each subfigure from left to right, sampled from the CIFAR10 dataset.

Select those images that have been successfully attacked by ZOO and estimated-gradient C&W and split them into training and test sets of size 2013 and 863, respectively. Table B.2 records the confusion matrix of the classification result. The two attacks are separated by a highly accurate classifier, which shows an obvious effect when using different optimization methods.

TABLE B.2. Confusion Matrix for ZOO and estimated-gradient C&W

		Predicted	
		ZOO	estimated-gradient C&W
Actual	ZOO	825	38
	estimated-gradient C&W	3	860

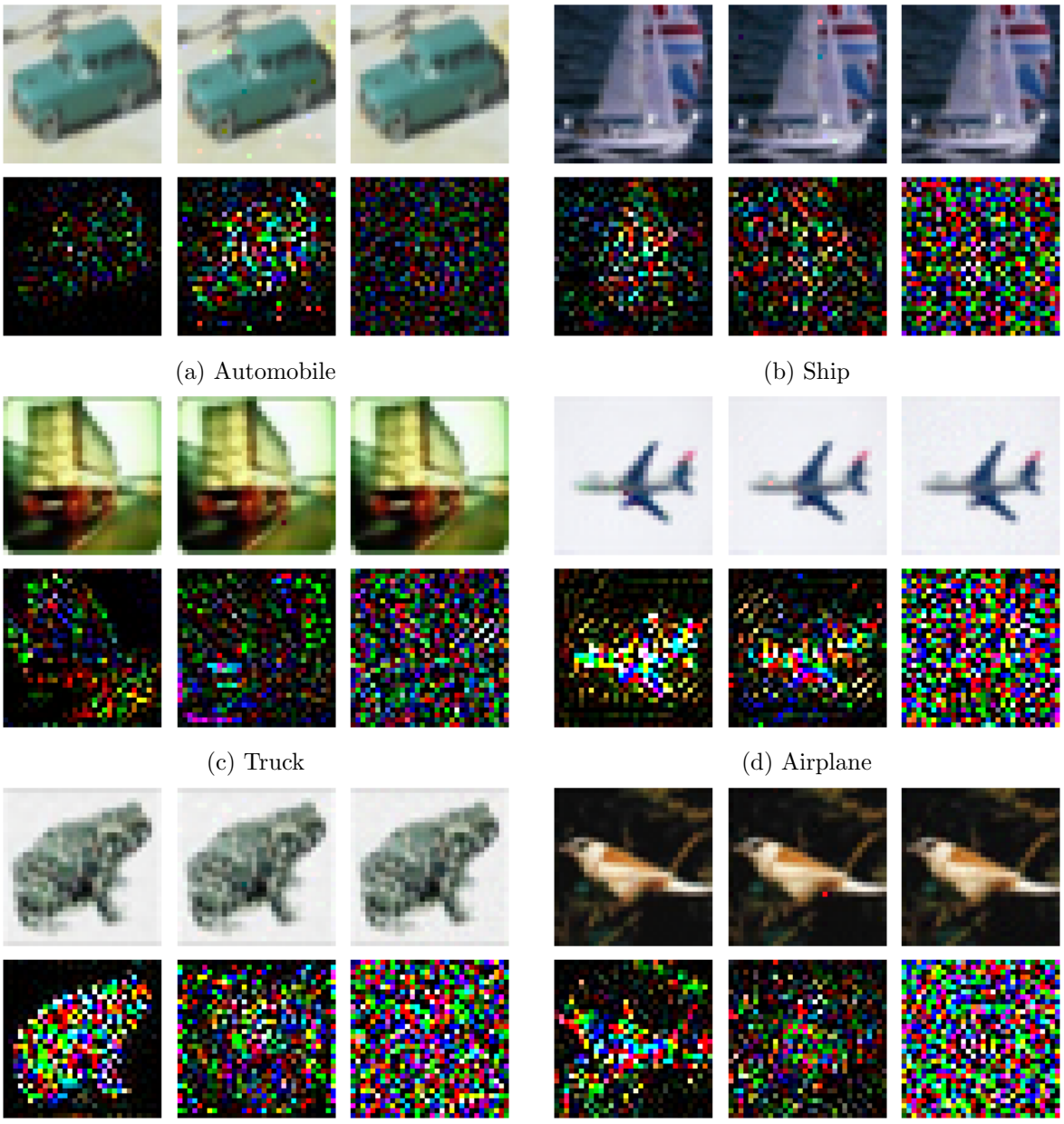
B.3. Supplementary Examples and Experiment in Section 4.4.1.3

Boundary attack starts with a random adversarial image and uses a random walk for each update. In Section 4.4.1.3, we study the effect of random start and lack of gradient information by comparing C&W, random-start C&W, and Boundary; more examples are displayed in Fig. B.5.

Select those images that have been successfully attacked by all three attacks and split them into training and test sets of size 3645 and 1566, respectively. Table B.3 records the confusion matrix. The three attacks can be classified by a high accuracy machine, indicating an obvious pattern among the attacks. This classification result proves that Boundary's blurry perturbations are caused by random start and random walk without gradient information.

TABLE B.3. Confusion Matrix for C&W, random-start C&W and Boundary

		Predicted		
		C&W	random-start C&W	Boundary
Actual	C&W	477	5	40
	random-start C&W	13	500	19
	Boundary	115	4	403



(a) Automobile (b) Ship

(c) Truck (d) Airplane

(e) Frog (f) Truck

FIGURE B.5. Additional visualization examples for C&W, random-start C&W, and Boundary are displayed in each subfigure from left to right, sampled from the CIFAR10 dataset.

Bibliography

[Aguiar-Pulido et al., 2016] Aguiar-Pulido, V., Huang, W., Suarez-Ulloa, V., Cickovski, T., Mathee, K., and Narasimhan, G. (2016). Metagenomics, metatranscriptomics, and metabolomics approaches for microbiome analysis: supplementary issue: bioinformatics methods and applications for big metagenomics data. *Evolutionary Bioinformatics*, 12:EBO–S36436.

[Alberg and Samet, 2003] Alberg, A. J. and Samet, J. M. (2003). Epidemiology of lung cancer. *Chest*, 123(1):21S–49S.

[Andriushchenko et al., 2020] Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. (2020). Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pages 484–501. Springer.

[Ardila et al., 2019] Ardila, D., Kiraly, A. P., Bharadwaj, S., Choi, B., Reicher, J. J., Peng, L., Tse, D., Etemadi, M., Ye, W., Corrado, G., et al. (2019). End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nature Medicine*, 25(6):954–961.

[Banning et al., 2011] Banning, N. C., Gleeson, D. B., Grigg, A. H., Grant, C. D., Andersen, G. L., Brodie, E. L., and Murphy, D. (2011). Soil microbial community successional patterns during forest ecosystem restoration. *Applied and environmental microbiology*, 77(17):6158–6164.

[Berg et al., 2020] Berg, G., Rybakova, D., Fischer, D., Cernava, T., Vergès, M.-C. C., Charles, T., Chen, X., Cocolin, L., Eversole, K., Corral, G. H., et al. (2020). Microbiome definition re-visited: old concepts and new challenges. *Microbiome*, 8(1):1–22.

[Blandin Knight et al., 2017] Blandin Knight, S., Crosbie, P. A., Balata, H., Chudziak, J., Hussell, T., and Dive, C. (2017). Progress and prospects of early detection in lung cancer. *Open Biology*, 7(9):170070.

[Brendel et al., 2017] Brendel, W., Rauber, J., and Bethge, M. (2017). Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*.

[Breslow, 1972] Breslow, N. E. (1972). Discussion of Professor Cox's paper. *J Royal Stat Soc B*, 34:216–217.

[Cao et al., 2020] Cao, W., Wu, R., Cao, G., and He, Z. (2020). A comprehensive review of computer-aided diagnosis of pulmonary nodules based on computed tomography scans. *IEEE Access*, 8:154007–154023.

[Cao et al., 2007] Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., and Li, H. (2007). Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136.

[Caracciolo et al., 2015] Caracciolo, A. B., Topp, E., and Grenni, P. (2015). Pharmaceuticals in the environment: biodegradation and effects on natural microbial communities. a review. *Journal of pharmaceutical and biomedical analysis*, 106:25–36.

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

[Chen et al., 2020] Chen, J., Jordan, M. I., and Wainwright, M. J. (2020). Hopskipjumpattack: A query-efficient decision-based attack. In *2020 ieee symposium on security and privacy (sp)*, pages 1277–1294. IEEE.

[Chen et al., 2018] Chen, P.-Y., Sharma, Y., Zhang, H., Yi, J., and Hsieh, C.-J. (2018). Ead: elastic-net attacks to deep neural networks via adversarial examples. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

[Chen et al., 2017] Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. (2017). Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26.

[Chen et al., 2021] Chen, Q., Xie, W., Zhou, P., Zheng, C., and Wu, D. (2021). Multi-crop convolutional neural networks for fast lung nodule segmentation. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(5):1190–1200.

[Cheng et al., 2019] Cheng, M., Singh, S., Chen, P., Chen, P.-Y., Liu, S., and Hsieh, C.-J. (2019). Sign-opt: A query-efficient hard-label adversarial attack. *arXiv preprint arXiv:1909.10773*.

[Cox, 1972] Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202.

[Cox, 1975] Cox, D. R. (1975). Partial likelihood. *Biometrika*, 62(2):269–276.

[Croce and Hein, 2020a] Croce, F. and Hein, M. (2020a). Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, pages 2196–2205. PMLR.

[Croce and Hein, 2020b] Croce, F. and Hein, M. (2020b). Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR.

[Davis and Yau, 2013] Davis, R. A. and Yau, C. Y. (2013). Consistency of minimum description length model selection for piecewise stationary time series models.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

[Deng, 2012] Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142.

[Du et al., 2022] Du, J., Guan, K., Zhou, Y., Li, Y., and Wang, T. (2022). Parameter-free similarity-aware attention module for medical image classification and segmentation. *IEEE Transactions on Emerging Topics in Computational Intelligence*.

[Fang et al., 2015] Fang, H., Huang, C., Zhao, H., and Deng, M. (2015). Cclasso: correlation inference for compositional data through lasso. *Bioinformatics*, 31(19):3172–3180.

[Faust, 2021] Faust, K. (2021). Open challenges for microbial network construction and analysis. *The ISME Journal*, 15(11):3111–3118.

[Faust and Raes, 2016] Faust, K. and Raes, J. (2016). Conet app: inference of biological association networks using cytoscape. *F1000Research*, 5.

[Feinman et al., 2017] Feinman, R., Curtin, R. R., Shintre, S., and Gardner, A. B. (2017). Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*.

[Friedman and Alm, 2012] Friedman, J. and Alm, E. J. (2012). Inferring correlation networks from genomic survey data. *PLoS computational biology*, 8(9):e1002687.

[Gao et al., 2019] Gao, R., Huo, Y., Bao, S., Tang, Y., Antic, S. L., Epstein, E. S., Balar, A. B., Deppen, S., Paulson, A. B., Sandler, K. L., et al. (2019). Distanced lstm: time-distanced gates in long short-term memory models for lung cancer detection. In *International Workshop on Machine Learning in Medical Imaging*, pages 310–318. Springer.

[Gloor et al., 2017] Gloor, G. B., Macklaim, J. M., Pawlowsky-Glahn, V., and Egozcue, J. J. (2017). Microbiome datasets are compositional: and this is not optional. *Frontiers in microbiology*, 8:2224.

[Goebel et al., 2021] Goebel, M., Bunk, J., Chattopadhyay, S., Nataraj, L., Chandrasekaran, S., and Manjunath, B. (2021). Attribution of gradient based adversarial attacks for reverse engineering of deceptions. *Electronic Imaging*, 2021(4):300–1.

[Gong et al., 2022] Gong, Y., Yao, Y., Li, Y., Zhang, Y., Liu, X., Lin, X., and Liu, S. (2022). Reverse engineering of imperceptible adversarial image perturbations. *arXiv preprint arXiv:2203.14145*.

[Gong et al., 2017] Gong, Z., Wang, W., and Ku, W.-S. (2017). Adversarial and clean data are not twins. *arXiv preprint arXiv:1704.04960*.

[Gonzalez et al., 2018] Gonzalez, A., Navas-Molina, J. A., Kosciolek, T., McDonald, D., Vázquez-Baeza, Y., Ackermann, G., DeReus, J., Janssen, S., Swafford, A. D., Orchanian, S. B., et al. (2018). Qiita: rapid, web-enabled microbiome meta-analysis. *Nature methods*, 15(10):796–798.

[Goodfellow et al., 2014] Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

[Han et al., 2015] Han, Q., Xu, K., and Airoldi, E. (2015). Consistent estimation of dynamic and multi-layer block models. In *International Conference on Machine Learning*, pages 1511–1520. PMLR.

[Harrell et al., 1982] Harrell, F. E., Califf, R. M., Pryor, D. B., Lee, K. L., and Rosati, R. A. (1982). Evaluating the yield of medical tests. *JAMA*, 247(18):2543–2546.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

[Holland et al., 1983] Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137.

[Hou et al., 2016] Hou, L., Samaras, D., Kurc, T. M., Gao, Y., Davis, J. E., and Saltz, J. H. (2016). Patch-based convolutional neural network for whole slide tissue image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2424–2433.

[Ilyas et al., 2018] Ilyas, A., Engstrom, L., Athalye, A., and Lin, J. (2018). Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, pages 2137–2146. PMLR.

[Ishaq et al., 2021] Ishaq, A., Sadiq, S., Umer, M., Ullah, S., Mirjalili, S., Rupapara, V., and Nappi, M. (2021). Improving the prediction of heart failure patients' survival using smote and effective data mining techniques. *IEEE Access*, 9:39707–39716.

[Katzman et al., 2018] Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., and Kluger, Y. (2018). Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18(1):1–12.

[Krizhevsky et al., 2009] Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

[Krizhevsky et al., 2017] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90.

[Lee et al., 2019] Lee, C., Yoon, J., and Van Der Schaar, M. (2019). Dynamic-deephit: A deep learning approach for dynamic survival analysis with competing risks based on longitudinal data. *IEEE Transactions on Biomedical Engineering*, 67(1):122–133.

[Li et al., 2018] Li, T., Zhou, F., Zhu, Z., Shu, H., and Zhu, H. (2018). A label-fusion-aided convolutional neural network for isointense infant brain tissue segmentation. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 692–695.

[Li and Li, 2017] Li, X. and Li, F. (2017). Adversarial examples detection in deep networks with convolutional filter statistics. In *Proceedings of the IEEE international conference on computer vision*, pages 5764–5772.

[Li et al., 2022] Li, Z., Cheng, H., Cai, X., Zhao, J., and Zhang, Q. (2022). Sa-es: Subspace activation evolution strategy for black-box adversarial attacks. *IEEE Transactions on Emerging Topics in Computational Intelligence*.

[Liao et al., 2019] Liao, F., Liang, M., Li, Z., Hu, X., and Song, S. (2019). Evaluate the malignancy of pulmonary nodules using the 3-d deep leaky noisy-or network. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3484–3495.

[Liu et al., 2020] Liu, S., Setio, A. A. A., Ghesu, F. C., Gibson, E., Grbic, S., Georgescu, B., and Comaniciu, D. (2020). No surprises: Training robust lung nodule detection for low-dose CT scans by augmenting with adversarial attacks. *IEEE Transactions on Medical Imaging*, 40(1):335–345.

[Madry et al., 2017] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

[Marchesi and Ravel, 2015] Marchesi, J. R. and Ravel, J. (2015). The vocabulary of microbiome research: a proposal.

[Matchado et al., 2021] Matchado, M. S., Lauber, M., Reitmeier, S., Kacprowski, T., Baumbach, J., Haller, D., and List, M. (2021). Network analysis methods for studying microbial communities: A mini review. *Computational and structural biotechnology journal*, 19:2687–2698.

[Meng and Chen, 2017] Meng, D. and Chen, H. (2017). Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 135–147.

[Metzen et al., 2017] Metzen, J. H., Genewein, T., Fischer, V., and Bischoff, B. (2017). On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*.

[Mielke et al., 2009] Mielke, M. M., Kozauer, N., Chan, K., George, M., Toroney, J., Zerrate, M., Bandeen-Roche, K., Wang, M.-C., Pekar, J., Mori, S., et al. (2009). Regionally-specific diffusion tensor imaging in mild cognitive impairment and Alzheimer's disease. *Neuroimage*, 46(1):47–55.

[Nicolae et al., 2018] Nicolae, M.-I., Sinn, M., Tran, M. N., Buesser, B., Rawat, A., Wistuba, M., Zantedeschi, V., Baracaldo, N., Chen, B., Ludwig, H., Molloy, I., and Edwards, B. (2018). Adversarial robustness toolbox v1.2.0. *CoRR*, 1807.01069.

[Pang et al., 2020] Pang, R., Zhang, X., Ji, S., Luo, X., and Wang, T. (2020). Advmind: Inferring adversary intent of black-box attacks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1899–1907.

[Rissanen, 1998] Rissanen, J. (1998). *Stochastic complexity in statistical inquiry*, volume 15. World scientific.

[Sadeghi et al., 2020] Sadeghi, K., Banerjee, A., and Gupta, S. K. S. (2020). A system-driven taxonomy of attacks and defenses in adversarial machine learning. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 4(4):450–467.

[Sbordone and Bortolaia, 2003] Sbordone, L. and Bortolaia, C. (2003). Oral microbial biofilms and plaque-related diseases: microbial communities and their role in the shift from oral health to disease. *Clinical oral investigations*, 7:181–188.

[Shaffer et al., 2023] Shaffer, M., Thurimella, K., Sterrett, J. D., and Lozupone, C. A. (2023). Scnic: Sparse correlation network investigation for compositional data. *Molecular Ecology Resources*, 23(1):312–325.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[Singh et al., 2020] Singh, S. P., Wang, L., Gupta, S., Goli, H., Padmanabhan, P., and Gulyás, B. (2020). 3d deep learning on medical images: a review. *Sensors*, 20(18):5097.

[Spiro and Silvestri, 2005] Spiro, S. G. and Silvestri, G. A. (2005). One hundred years of lung cancer. *American Journal of Respiratory and Critical Care Medicine*, 172(5):523–529.

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

[Team, 2011] Team, N. L. S. T. R. (2011). The national lung screening trial: overview and study design. *Radiology*, 258(1):243–253.

[Thompson et al., 2017] Thompson, L. R., Sanders, J. G., McDonald, D., Amir, A., Ladau, J., Locey, K. J., Prill, R. J., Tripathi, A., Gibbons, S. M., Ackermann, G., et al. (2017). A communal catalogue reveals earth's multiscale microbial diversity. *Nature*, 551(7681):457–463.

[Tsilimigras and Fodor, 2016] Tsilimigras, M. C. and Fodor, A. A. (2016). Compositional data analysis of the microbiome: fundamentals, tools, and challenges. *Annals of epidemiology*, 26(5):330–335.

[Wang et al., 2019] Wang, J., Gao, R., Huo, Y., Bao, S., Xiong, Y., Antic, S. L., Osterman, T. J., Massion, P. P., and Landman, B. A. (2019). Lung cancer detection using co-learning from chest CT images and clinical demographics. In *Medical Imaging 2019: Image Processing*, volume 10949, page 109491G. International Society for Optics and Photonics.

[Watts et al., 2019] Watts, S. C., Ritchie, S. C., Inouye, M., and Holt, K. E. (2019). Fastspar: rapid and scalable correlation estimation for compositional data. *Bioinformatics*, 35(6):1064–1066.

[Weiss et al., 2016] Weiss, S., Van Treuren, W., Lozupone, C., Faust, K., Friedman, J., Deng, Y., Xia, L. C., Xu, Z. Z., Ursell, L., Alm, E. J., et al. (2016). Correlation detection strategies in microbial data sets vary widely in sensitivity and precision. *The ISME journal*, 10(7):1669–1681.

[Weiss et al., 2017] Weiss, S., Xu, Z. Z., Peddada, S., Amir, A., Bittinger, K., Gonzalez, A., Lozupone, C., Zaneveld, J. R., Vázquez-Baeza, Y., Birmingham, A., et al. (2017). Normalization and microbial differential abundance strategies depend upon data characteristics. *Microbiome*, 5:1–18.

[Wierstra et al., 2014] Wierstra, D., Schaul, T., Glasmachers, T., Sun, Y., Peters, J., and Schmidhuber, J. (2014). Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980.

[Wu et al., 2018] Wu, L., Hsieh, C.-J., and Sharpnack, J. (2018). Sql-rank: A listwise approach to collaborative ranking. In *International Conference on Machine Learning*, pages 5315–5324. PMLR.

[Xie et al., 2017] Xie, C., Wang, J., Zhang, Z., Ren, Z., and Yuille, A. (2017). Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*.

[Zanddizari et al., 2021] Zanddizari, H., Zeinali, B., and Chang, J. M. (2021). Generating black-box adversarial examples in sparse domain. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(4):795–804.

[Zhang et al., 2020] Zhang, D., Li, L., Sripada, C., and Kang, J. (2020). Image-on-scalar regression via deep neural networks. *arXiv preprint arXiv:2006.09911*.

[Zhang et al., 2019] Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. (2019). Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR.

[Zheng and Hong, 2018] Zheng, Z. and Hong, P. (2018). Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks. *Advances in Neural Information Processing Systems*, 31.

[Zhong et al., 2023] Zhong, G., Ding, W., Chen, L., Wang, Y., and Yu, Y.-F. (2023). Multi-scale attention generative adversarial network for medical image enhancement. *IEEE Transactions on Emerging Topics in Computational Intelligence*.

[Zhu et al., 2016] Zhu, X., Yao, J., and Huang, J. (2016). Deep convolutional neural network for survival analysis with pathological images. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 544–547. IEEE.

[Zuidhof, 2017] Zuidhof, G. (2017). Full preprocessing tutorial.