Understanding and Improving Representational Robustness of Machine Learning Models

by

Ching-Yun Ko

B.S., Wuhan University (2017)
M.Phil., University of Hong Kong (2019)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2024

Authored by: Ching-Yun Ko
Department of Electrical Engineering and Computer Science
May 17, 2024

Certified by: Luca Daniel
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by: Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Understanding and Improving Representational Robustness of Machine Learning Models

by

Ching-Yun Ko

Submitted to the Department of Electrical Engineering and Computer Science
on May 17, 2024, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

The fragility of modern machine learning models has drawn a considerable amount of attention from both academia and the public. In this thesis, we will do a systematic study on the understanding and improvement of several machine learning models, including smoothed models and generic representation networks. Specifically, we put our focus on studying representational robustness, which we define as the "robustness" (or generally trustworthy properties) in the induced hidden space of a given network. For a generic representation network, this corresponds to the representation space itself, while for a smoothed model, we will treat the logits of the network as the target space. Representational robustness is fundamental to many trustworthy AI areas, such as fairness and robustness. In the thesis, we discover that the certifiable robustness of randomized smoothing is at the cost of class unfairness. We further analyze ways to improve the training process of the base models and their limitations. For generic non-smooth representation models, we find a link between self-supervised contrastive learning and supervised neighborhood component analysis, which naturally allows us to propose a general framework that achieves better accuracy and robustness. Furthermore, we realize that the current evaluation practice of foundational representation models involves extensive experiments across various real-world tasks, which are computationally expensive and prone to test set leakage. As a solution, we propose a more lightweight, privacy-preserving, and sound evaluation framework for both vision and language models by utilizing synthetic data.

Thesis Supervisor: Luca Daniel
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

Undoubtedly I owe my adviser Prof. Luca Daniel a great deal. He has greatly supported me in every possible aspect of this long and difficult journey (it was really difficult!). He has been a continuous source of support and inspiration. I learned so much from him that even the three sentences before this one are learned from his PhD thesis! The first year of my PhD is welcomed by the pandemic and thanks to that I could echo a line I read about a long time ago "That's, I guess a part of gaining maturity: you become a lot less certain about everything in life as you age." Luca has been the support when my personal life has been characterized by a few setbacks. Thank you for teaching me to draw a line and keep hoping for things to be better in the future.

I would like to thank my thesis committee members for their valuable and constructive comments. I thank Prof. Duane Boning for learning every detail of the four pieces of work included in this thesis by agreeing to be my RQE and PhD thesis committee member. All the questions and thoughts being shared have turned this thesis into a better shape. I promise you will not be forced to read these same contents anymore in the future. I also thank Dr. Pin-Yu Chen for a million reasons. This thesis would not be possible without him. I truly feel I can address Pin-Yu as a co-adviser for this work. He is always enthusiastic about research and seems always on top of things (and active on Slack). I have enjoyed all our discussions together and I am hopeful that this will not be it - there are more to come!

I would like to thank my previous mentors, Dr. Lily Weng, and Dr. Kim Batselier, and my Master's advisor Dr. Ngai Wong for their invaluable guidance along my academic journey. Lily was a senior Ph.D. student in the group when I joined MIT and she has been a role model for me to learn from. I remember in a post against one of Lily's earliest papers, *CLEVER*, she responded to every question asked clearly and accurately, and put "CLEVER Authors" as the signature. That was so clever and tough! I have always used this example to encourage myself during my own paper rebuttals. I need to thank Kim as well. Kim was a postdoctoral fellow in

the group when I joined HKU for my master's. Kim believed in me and taught me through a project from the very first week of my life there, which later made the first peer-reviewed paper I have, an Automatica journal paper. He let me work out all the puzzles together in those days and showed me the perseverance an adequate researcher should have (and the tolerance a Belgian can have to spicy food). I must also thank Dr. Wong. Dr. Wong has been a continuous source of inspiration. During my time in Hong Kong, I always bumped into him reading papers while waiting for the elevators at HKU station. He wholeheartedly supported my career and introduced me to all research opportunities. I miss all the discussions with him and thank the acknowledgements he gave which built a solid foundation for my future academic path. My accomplishment will not be possible without him.

I would also like to thank all my research collaborators (and mentors), Dr. Payel Das, Dr. Sijia Liu, Prof. Riccardo Lattanzi, Dr. Ilias Giannakopoulos, Dr. Zhaoyang Lyu, and many many more. Thank you for continuing to inspire me!

The time spent with my smart and nice colleagues and friends at MIT is truly amazing. I would like to thank Xiaolin Fang, Itay Fayer, Yanwei Felix Wang, Bowen Pan, Molin Zhang, Cathy Chen, Connie Wang, Dousabel Tay, Charlotte Loh, Yang Liu, Peiqi Mark Wang, Cheng-I Jeff Lai, Yung-Sung Chuang, and Wei Liao. Special thanks to my labmates Jeet Mohapatra and Wang Zhang for all the discussions we have and fun times together chatting about everything but research. Many of you are also my collaborators and I look forward to more collaborations with you!

My thanks also go to my friends outside MIT. I did not know Dr. Zheng Zhang until 2016 when he helped Dr. Wong to look for new graduate students. He encouraged me to take an academic path. Following his steps, I joined the same groups as he did years ago at HKU and later at MIT. I thank Fangyue Peng, Ankita Gupta, Chang Sherry Liu, Xuangui Huang, Ben Huh, Nima Dehmamy, Yuke Zhang, Zhuolun Leon He, Jingming Wang, Jing Li, and Xiaolu Hsi for all the enlightening conversations with me and our time together. I thank Hongbo Michael Lin for the company and the joy you brought. Thank all of you who invited me to ski and play tennis. You are my work-life balance heroes.

I also greatly appreciate MIT-IBM Watson AI lab for supporting my research throughout my PhD journey.

Lastly, I would like to give my deepest gratitude to my family, who taught me to be a woman of integrity and believe in myself. My family said they are already very proud of me, thank you for saying that. My brother thanked me in his thesis and I should thank him back so here you go - Thank you! I thank my grandfather who passed away before I was born. I heard a lot about you from mom and I know you are a great scientist before you go. I will always carry your spirit and do great research.

My start at MIT was not smooth, and Luca wrote to us in the 2020 New Year greeting email "You will realize that everyone went through an initial almost inevitable frustrating stage. It takes time, but it eventually pays off. And if you have a chance to talk to any of the people that are a few years down the road after leaving the group, they will all confirm that they felt well equipped to attack any task, or fence any adversity." I can now confirm that this is true.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The vulnerability of deep neural networks to human-imperceptible adversarial perturbations has attracted great attention within the machine learning community since the seminal works [170, 7]. This has remained an important concern for various machine learning fields, ranging for instance from computer vision [170] to speech recognition [17]. In particular, for safety-critical applications, such as self-driving cars and surveillance, there is almost zero tolerance for erroneous decisions. As a result, the existence of adversarial examples in deep neural networks has motivated efforts toward robustness quantification, as well as toward designing training algorithms that can enhance such robustness [42, 47, 95]. In this thesis, we intend to understand and improve the representational robustness of modern machine learning models.

1.1.1 The Representational robustness of machine learning models

Representational robustness refers to the reliability of the induced hidden space within a neural network model. This concept is particularly pertinent in machine learning since the hidden layers of a network ought to capture intricate patterns from the input data. In this thesis, we define representational robustness as the ability

of these hidden representations to maintain desirable trustworthy properties across different inputs or perturbations. The desirable trustworthy properties could include accuracy, fairness, adversarial robustness, etc. For a generic representation network $g(\cdot)$, the natural choice of the induced hidden space is indeed the output space of the representation network. These constructed spaces are specifically trained to encode crucial information about the input data through representation learning, enabling the network to perform various tasks like classification, regression, or generation through a simple task-specific downstream network. On the other hand, in the context of a smoothed model, the smoothing filter is applied to the entire base network $\arg\max_j \mathbb{P}[j = \arg\max_i g_i(x)], x \sim \mathcal{N}(x_0, \sigma^2\mathcal{I})$. Thus, we will directly treat the logits of the network $\mathbb{P}[j = \arg\max_i g_i(x)]$ as the target space for assessing representational robustness. In this case, we are particularly interested in the different behavior between the base and the smoothed network.

Studying representational robustness is fundamental to advancing the field of machine learning for several reasons. First of all, as will be discussed in the later chapters of the thesis, having a deeper understanding of what each component (representation network, smoothing operator, etc.) does helps us to be more cautious and more conscious about the potential side effects of the operations. This understanding will also build up the foundation for improving these network designs. Secondly, robust representations become increasingly vital as the machine learning community gradually shifts the focus to task-agnostic pretraining and task-specific finetuning. In safety-critical applications, erroneous predictions due to brittle representations can have severe consequences. From this perspective, representational robustness is fundamental to many trustworthy AI areas, as the pretrained representation network will contribute to the overall trustworthiness of any machine learning systems built on it. By investigating and enhancing representational robustness, one can build more resilient AI systems and prevent error propagation.

1.1.2 The limit and cost of smoothed models

To date, there are two popular ways to approach the problem of robustness evaluation: 1) attack evaluation and 2) formal verification. From the attack perspective, the adversary would like to develop strong adversarial attacks that can fool the network classifier with the smallest adversarial distortions [14, 60, 122, 22]. Whereas the purpose of formal verification methods is to guarantee that intrinsic robustness conditions will always hold. For example, one key goal, within robustness verification, is to show that no adversarial examples can ever exist within an μ-neighborhood of the original test sample. Furthermore, ideally, the formal verification algorithms should identify the largest possible μ. As a result of the shared concern of unverified models in real-life deployment, the focus has shifted to seek trust-worthy and attack-agnostic robustness verification [72, 194, 165, 218, 82].

However, due to the intrinsic hardness (NP-completeness) of the robustness verification problem, these certifiable verification methodologies do not scale to large networks [114]. To cope with this, one emerging branch of studies, *randomized smoothing* [31, 99, 103], proposes transforming the original network into a "smoothed" counterpart. This new counterpart now returns the class with the highest probability by querying isotropic Gaussian noise $N(0, \sigma^2 I)$ corrupted data. This corresponds to applying low-pass filters (*cf.* Gauss–Weierstrass transform, Gaussian blur, or Gaussian filter in signal processing) to score functions. In the first part of the thesis, we will study the representational robustness of smoothed models, where we treat the logits of the smoothed network as representations. We will disclose the unwanted shrinking effects of current randomized smoothing workflows, which might cause undesirable group unfairness. Furthermore, as randomized smoothing is commonly accompanied by noise augmentation during the training, we will also show the limit and caveat of such procedures.

1.1.3 Understanding and improving generic self-supervised representation learning

Given the knowledge of the cost and limit of smoothed models, we shift our focus to generic non-smooth machine learning models. Here, we will consider the seminal tool in self-supervised representation learning, contrastive learning, which typically constructs a higher-dimensional representation space. If the representation network is good enough, then the downstream classifier can be as simple as a linear layer. Contrastive learning essentially constructs optimization objectives that aim to leverage pairs of positive and negative samples for representation learning, which relates to exploiting neighborhood information in a feature space. From this perspective, we are inspired to formally establish the connection between the supervised Neighborhood Component Analysis (NCA) and the self-supervised contrastive learning, and propose generalized contrastive loss (named NaCl) which outperforms the existing paradigm.

Even though contrastive learning (or representation learning in general) has drawn much attention in the past years, this method still faces several challenges. For example, the definition of positive and negative pairs heavily relies on the downstream tasks, and the computation of positive and negative pairs grows quadratically with the size of the dataset. Most importantly, over the past years, representation learning has been evaluated mostly only by how they cluster or metrics such as the standard downstream classification accuracy. However, as will be shown in the corresponding chapter, there is a concerning insufficiency of those methods in addressing robustness. Thus, we urge the necessity of establishing contrastive learning methods that score high in not only the standard accuracy but also the adversarial accuracy. As such, we will further propose an integrated framework (named IntNaCl) that accounts for both standard accuracy and adversarial cases.

1.1.4 Evaluating robustness-accuracy of large models using synthetic data

Generally, over the past few years, the ML community has witnessed a paradigm shift in deep learning from task-centric model design to task-agnostic representation learning and task-specific fine-tuning. The pretrained representation networks developed from contrastive learning and beyond, are being used more widely and evaluated extensively across various real-world tasks. They are now often used as a foundation for different downstream tasks, and hence also named foundation models. When gauging the usefulness of a foundation, it is a convention to conduct evaluations on selected public datasets. For example, ViT [40] reports accuracy on 25 tasks, CLIP [141] on 27 datasets, and PLEX [181] on over 40 datasets to systematically evaluate different reliability dimensions on both vision and language domains. However, new concerns about proper performance evaluation have been raised.

The first important concern is the inconclusiveness of this type of evaluations. For instance, ViT-L/16 is reportedly performing better than ViT-B/16 on 23 out of 27 vision tasks in [141], but worse than ViT-B/16 on FoodSeg103 and magnetic resonance imaging according to [202, 182, 128]. That said, a poor probing result might come from either (1) evaluation data bias, (2) true model deficiency, or both. Secondly, the trending practice of pretraining and fine-tuning also signifies immediate damage to all adapted applications if the foundation model has hidden risks [12], such as lacking robustness to adversarial examples. Therefore, we propose to design new benchmarks for both vision and language foundation models (named SynBench and SynTextBench) that are based on synthetic data whose optimal performance can be characterized and referenced. These two new evaluation paradigms will thereby evaluate the representational robustness by considering the same synthetic tests in the vision or language domain. By construction, our use of synthetic data will also circumvent the real private user data leakage through API calls during evaluation.

1.2 Thesis contributions and organization

Table 1.1: Thesis contributions and organization. Contents not covered in this thesis are colored in gray. The analysis of vision non-smooth models applies to language non-smooth models, but will not be discussed explicitly in this thesis.

Models		Understanding	Improving	
			training/inference	evaluation
Vision	Smoothed models	Chapter 3	Chapter 3	Mohapatra et al.
	Non-smooth models	Chapter 4	Chapter 4	Chapter 5
Language	Non-smooth models	Chapter 4*	Ko et al.	Chapter 6

1.2.1 Contributions of this thesis

This thesis focuses on understanding and improving representational robustness of machine learning models. The novel contributions include three parts.

In the first part, we point out the hidden risks of current randomized smoothing workflows and study the improvement data augmentation can bring to mitigate those risks.

- **Contribution 1.** In Chapter 3, we prove the hidden cost of randomized smoothing is class-wise fairness, *i.e.*, decision boundaries of smoothed classifiers will shrink, resulting in disparity in class-wise accuracy. Specifically, we identify sufficient conditions under which Gaussian smoothing leads to a decrease in classification accuracy and characterize the theoretical lower bound of the shrinking rate. We also show that data augmentation in the training process does not necessarily resolve the shrinking issue due to the inconsistent learning objectives. We analyze the effect of noise augmentation and show that it may leads to low classification accuracy for large σ on both synthetic and real datasets.

The second part of the thesis tries to understand a seminal approach in self-supervised representation learning, contrastive learning, from the perspective of supervised neighborhood component analysis (NCA), and propose a generalized training method to improve the accuracy and robustness.

- **Contribution 2.** In Chapter 4, we establish the relationship between contrastive learning and NCA, and propose new contrastive loss dubbed **NaCl** (Neighborhood analysis Contrastive loss). We provide theoretical analysis on NaCl and show better generalization bounds over the baselines. Building on top of NaCl, we propose a generic framework called Integrated contrastive learning (**IntCl** and **IntNaCl**) that could simultaneously achieve good accuracy and robustness on downstream tasks. We show that the spectrum of recently-proposed contrastive learning losses [29, 154, 75] can be included as special cases of our framework.

The third part of the thesis identifies the drawbacks of current evaluation practices of representation networks and proposes improved evaluation benchmarks using synthetic data.

- **Contribution 3.** In Chapter 5, to circumvent the need for real-world data in evaluation, we explore the use of synthetic binary classification tasks with Gaussian mixtures to probe pretrained models and compare the robustness-accuracy performance on pretrained representations with an idealized reference. Our approach offers a holistic evaluation, revealing intrinsic model capabilities and reducing the dependency on real-life data for model evaluation. Evaluated with various pretrained image models, the experimental results confirm that our task-agnostic evaluation correlates with actual linear probing performance on downstream tasks and can also guide parameter choice in robust linear probing to achieve a better robustness-accuracy trade-off.

- **Contribution 4.** In Chapter 6, we propose a new evaluation workflow that generates steerable synthetic language datasets and proxy tasks for benchmarking the performance of pertrained LMs on sentence classification tasks. This approach allows for better characterization of the joint analysis on the robustness and accuracy of LMs without risking sensitive information leakage. It also provides a more controlled and private way to evaluate LMs that avoids overfitting specific test sets. Verified on various pretrained LMs, the proposed approach

demonstrates promising high correlation with real downstream performance.

1.2.2 Thesis outline

This thesis is organized as follows:

- In Chapter 2, we give an introduction to some background. We aim to make this background chapter as brief as possible.
- Chapter 3 to Chapter 6 present the details of our four novel contributions, including definitions and severity of the hidden cost of randomized smoothing, how to interpret contrastive learning from NCA and propose extensions, and how to evaluation those large pretrained vision/language models in a real-data-free and task-agnostic way to avoid data leakage and lift the computational burden.
- Finally, Chapter 7 summarizes the results of this thesis and discusses some future work in this field.

Chapter 2

Backgrounds

2.1 Adversarial robustness

Despite neural networks' supremacy in achieving impressive performance, they have been proved vulnerable to human-imperceptible perturbations [61, 170, 124, 122]. In the supervised learning setting, an adversarial perturbation δ is defined to render inconsistent classification result of the input x: $f(x_0 + \delta) \neq f(x)$, where f is a neural network classifier. A stronger adversarial attack means it can find δ with higher success attack rate under the same ϵ-budget ($\|\delta\|_p \leq \epsilon$). One of the most popular and classical attack algorithms is FGSM [61], where with a fixed perturbation magnitude ϵ, FGSM finds adversarial perturbation by 1-step gradient descent. Another popular attack method we consider in this thesis is PGD [115], which assembles the iterative-FGSM [96] but with different initializations and learning rate constraints.

2.1.1 Randomized smoothing with Gaussian filtering

Generally, the prediction of a model f for input x_0 is given by taking the highest output of the score function (a neural network) $g(x_0)$. Let e_i denote the i^{th} basis vector with all components 0 and the i^{th} component be 1. Then the base classifier can be given as

$$f(x_0) = e_{\xi_A}; \quad \xi_A = \arg\max_j \ g_j(x_0). \tag{2.1}$$

Correspondingly, under randomized smoothing the prediction for a model g is given as the "most likely" standard prediction output by the model when noise is added to the input. Conventionally, the resulting classifier is referred to as the *smoothed classifier* and the type of noise added to the input is denoted as the *smoothing measure*. When isotropic Gaussian distribution $\mathcal{N}(0, \sigma^2\mathcal{I})$ is used as the smoothing measure, the smoothed function f_σ is given as

$$f_\sigma(x_0) = e_{\xi_A};$$
$$\xi_A = \arg\max_j \ \mathbb{P}[j = \arg\max_i g_i(x)],\ x \sim \mathcal{N}(x_0, \sigma^2\mathcal{I}).$$

There has been a lot of research in developing robustness verification techniques for the *base classifier* in Equation equation 2.1 [72, 194, 56, 144, 195, 199, 189, 106], *i.e.* given g, x_0, ξ_A and p, find the maximum value of r such that $\arg\max_j g_j(x_0 + \delta) = \xi_A,\ \forall\ \|\delta\|_p \leq r$. However, due to the intrinsic hardness of the problem [85, 194, 178], the above approaches can hardly scale to state-of-the-art deep neural networks such as ResNet-50 and VGG-19 nets. On the other hand, it is also possible to perform robustness verification on the *smoothed classifier*. To solve the problem of certification, [99] first applied differential privacy techniques to derive a non-trivial lower bound of r for $p = 1, 2$. The bound was later improved by [103] via the tools in information theory for $p = 2$. Recently, [31] proved a tighter bound of r for $p = 2$ below:

$$r = \frac{\sigma}{2}\left[\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})\right], \tag{2.2}$$

where σ is the smoothing factor in the Gaussian noise, Φ^{-1} is the inverse of standard Gaussian CDF, and $\underline{p_A}$ and $\overline{p_B}$ are the lower/upper bound on the probability with class ξ_A and ξ_B (ξ_A is the top-1 class of the smoothed classifier and ξ_B is the "runner-up" class), respectively. In practice, [31] sets $\overline{p_B} = 1 - \underline{p_A}$ and abstains when $\underline{p_A} < 0.5$, implying that no radius can be certified in this case.

2.1.2 Data augmentation with Gaussian corruptions

In the seminal work of randomized smoothing , [31] and [99] suggest to apply randomized smoothing during training (noise augmentation) for better classification accuracy. We first recall that a standard learning problem takes the form of

$$\mathcal{R} = \mathbb{E}_{x \in \mathcal{X}}[l(f(x), h(x))],$$

where $\mathcal{X}$, $\mathcal{Y}$, l, f, and h are the input space, the output space, the loss function, a neural network, and the ground-truth classifier, respectively. Given some probability distribution $\mathfrak{D}_p$ the noise smoothing risk takes the form of

$$\begin{aligned} \mathcal{R}_{\text{RS}} &= \mathbb{E}_{x \in \mathcal{X}}[l(f_\sigma(x), h(x))] \\ &= \mathbb{E}_{x \in \mathcal{X}}[l(\mathbb{E}_{z \sim \mathfrak{D}_p}[f(x+z)], h(x))]. \end{aligned}$$

[31] motivate the use of corrupted samples during training by arguing that, when l is chosen to be the cross entropy and $\mathfrak{D}_p = \mathcal{N}(0, \sigma^2 I)$, the noise augmentation risk

$$\mathcal{R}_{\text{RS-train}} = \mathbb{E}_{x \in \mathcal{X}}[\mathbb{E}_{z \sim \mathcal{D}_p}[l(f_{\text{train},\sigma}(x+z), h(x))]]$$

constitutes a lower bound of $\mathcal{R}_{\text{RS}}$. We distinguish $f_{\text{train},\sigma}$ from f since they are learned from different objectives. Throughout this thesis, we abbreviate Gaussian noise augmentation (*i.e.* $\mathfrak{D}_p$ be the Gaussian centered at the origin) as data augmentation.

2.1.3 (Robust) Bayes optimal classifier for Gaussian models

Despite the difficulty of characterizing the optimal classifier with the minimum loss for generic data, for data drawn from class-conditional Gaussian distribution, the explicit optimal strategy is given by Fisher's linear discriminant rule [81, 137]. Likewise, the optimal classification strategy can also be given for such data in the presence of input perturbations [4, 36].

Let $\mathcal{N}(\mu, \Sigma)$ denote Gaussian distribution with mean μ and variance Σ. Generally,

for binary classification problems with data pair (x, y) generated from a probability distribution $P_{\mu,\Sigma}$:

$$x|y = 1 \sim \mathcal{N}(\mu, \Sigma),\ x|y = -1 \sim \mathcal{N}(-\mu, \Sigma),$$

the classifier that minimizes the adversarial loss [2] $\max_{x':\|x'-x\|\leq\epsilon} \mathbb{1}(f(x') \neq y)$, the robust Bayes optimal classifier [4, 36], is given by

$$\text{sign}(w_0^T x),$$

where $w_0 = \Sigma^{-1}(\mu - z_\Sigma(\mu))$ and z_Σ is the solution of the convex problem

$$\arg\min_{\|z\|_2\leq\epsilon} (\mu - z)^T \Sigma^{-1} (\mu - z). \tag{2.3}$$

Putting ϵ to 0 naturally lead to the naive Bayes optimal classifier $\text{sign}(\mu^T \Sigma^{-1} x)$.

2.2 Sentence representations and sentiment lexicons

2.2.1 Sentence representations

To obtain performant language models, learning universal sentence representations that capture rich information for various downstream NLP tasks without task-specific fine-tuning is an active research field and has also been studied extensively in the past years [91, 33, 53, 104, 169, 57, 54, 30]. While learning to extract ideal sentence embeddings, [53, 104, 46] have pinpointed the anisotropic behavior in the sentence embedding vector space as a reason behind sentence embeddings' poor capture of semantic information. To remedy the situation, Bert-flow [104] and Bert-whitening [169] transformed the sentence embedding distribution into an isotropic Gaussian distribution through normalizing flow and whitening post-processing. Through contrastive learning, SimCSE [54] and DiffCSE [30] also achieved new state-of-the-art sentence embedding performance by promoting uniformity and alignment [190].

2.2.2 Sentiment lexicons

SentiWordNet 3.0 [3] is a lexical resource that provides sentiment information for each word in WordNet [119], a widely-used lexical database of English words and their relationships. SentiWordNet 3.0 is an improved version of SentiWordNet 1.0 [44], 1.1 [45], 2.0 [43]. SentiWordNet automatically assigns synsets of WordNet according to notions of "positivity", "negativity", and "neutrality". The sentiment scores of a synset are assigned on a scale from 0.0 to 1.0 and sum to 1, reflecting a fine-grained opinion-related word-level labeling. SentiWordNet has been used in a variety of natural language processing tasks, such as sentiment analysis [37, 127, 87], opinion mining [78, 35], representation learning [86], and curriculum learning [148]. Besides SentiWrodNet, other sentiment lexicons include Affective Norms for English Words (ANEW) [13], Warriner lexicon [192], a new ANEW [126], and ANEW+ [160]. In this thesis, we will demonstrate the use of sentiment lexicon with word-level labels in constructing synthetic datasets using SentiWordNet 3.0; however, the framework proposed in this thesis can take any lexicon with word-level labels. We also envision our framework to benefit from a richer vocabulary and extend to other value lexicons like moral lexicons [152].

Chapter 3

The limit and cost of smoothed models

3.1 Introduction

Current mainstream methods to evaluate robustness of DNNs against adversarial examples [170, 7] employ robustness verification. Such techniques can guarantee that no adversarial examples can exist within a specified distance r from a given input. As computing the largest possible r has been proven to be NP-complete [85], one popular approach is to derive a certified lower bound of r through convex/linear relaxation [72, 194, 165, 218], which can be computed efficiently. Nevertheless, these techniques can hardly scale to state-of-the-art DNNs on ImageNet, motivating the idea of applying *randomized smoothing* [31, 99, 103, 80, 100] (*i.e.* a spatial low-pass filter) to transform the original classifier into a "smoothed" counterpart. This new smoothed classifier now returns the class with the highest probability by querying input data that has been purposely corrupted by isotropic Gaussian noise $N(0, \sigma^2\mathcal{I})$.

Although randomized smoothing allows non-trivial robustness verification for the smoothed classifier on ImageNet, the side-effects of randomized smoothing have not yet been rigorously studied, except for a case-study of one specific binary classifier in [64] and some impossibility results on accuracy-certification trade-off [207, 10, 94]. The main motivation of this chapter is to take a deep dive into the hidden cost of randomized smoothing for general multi-class classifiers.

The development of this chapter is as follows: in Section 3.2 we fully expose a major

Table 3.1: A look-up table of theoretical (**T**) and numerical (**N**) contributions in Section 4.

region geometry	shrinking	vanishing rate σ_{van}	shrinking rate	certified radius
bounded	**T** (Thm. 3)	**T** - lower bnd. (Thm. 4)	**N** - lower bnd. (Fig. 2)	**N** - case study
semi-bounded	**T** (Thm. 5)	not applicable	**T** - lower bnd. (Thm.7)	**N** - case study

hidden cost of randomized smoothing – biased predictions, by providing evidences from both real-life and synthetic datasets; in Section 3.3 we provide a comprehensive theory exposing the root of the biased prediction – referred to as the *shrinking phenomenon* in the remainder of the thesis; in Section 3.4 we hold a discussion on the effects of data augmentation on the shrinking phenomenon and implications given by our theoretical analysis. We give rigorous theoretical analysis and empirical evidences to facilitate a thorough understanding of the problems of existing randomized smoothing methods.

3.1.1 Our contributions

- We provide theoretical characterization for the shrinking phenomenon incurred by randomized smoothing. The classes with relatively small decision regions (compact geometric distribution) shrink with enlarging σ and thus resulting in highly-unfair class-wise accuracy

- We give theoretical characterization for the severity of the shrinking phenomenon and show the rate of the shrinkage (*cf.* Table 3.1).

- We analyze the effect of Gaussian noise augmentation (*cf.* data augmentation in [31]) and conclude that this can lead to a loss in mutual information. In terms of the classification accuracy, applying Gaussian noise augmentation during training may lead to low classification accuracy for large σ on both synthetic and real datasets.

3.1.2 Related works

Randomized smoothing was initially introduced as a heuristic defense by [111] and [204]. Later, [99] formulated it as a certification method using ideas from differential

privacy, which was then improved by [103] using Renyi divergence. For gaussian-smoothed classifiers, [31] made the certified bounds worst-case-optimal in the context of certified ℓ_2 norm radii by using the Neyman-Pearson Lemma, while authors of [156] combined the certification method with adversarial training to further improve the empirical results. Along another line of works, some extended existing certification methods to get better ℓ_p norm certified radii using different smoothing distributions, e.g. a discrete distribution for ℓ_0 certificates [100], the Laplace distribution for ℓ_1 certificates [173, 186], and the generalized Gaussian distribution for ℓ_∞ certificates [215]. Recently, [207] proposed a general method for finding the optimal smoothing distribution given any threat model, as well as a framework for calculating the certified robustness for the smoothed classifier.

Recently, a number of works have shown that for ℓ_p norm threat models with large p, it is impossible to give a big certified radius $\left(O(d^{\frac{1}{p}-\frac{1}{2}}\right)$ where d is the input dimension) while retaining a high standard accuracy. In particular, the results on ℓ_∞ threat model given in [10, 94] and the results on ℓ_p (for sufficiently large p) threat models given in [207] establish a certification/accuracy trade-off, which also exaggerates the need for an extended and generalized framework that breaks the confined trade-off and impossibility results.

3.2 Two motivating examples

The major highlight of randomized smoothing techniques in the scope of adversarial robustness is its ability to provide non-trivial robustness guarantees (certified radii) for large networks. With this in mind, as pointed out in [31], for randomized smoothing with parameter σ, the maximum achievable certified radius is around 4σ, implying larger smoothing factor σ is needed for a larger maximum achievable certified radius[1]. This need is further justified in [31] by pointing out the trade-off between the sample complexity and certified radii with a fixed smoothing factor. Therefore, one has to use

[1]One can also gain insights from that the certified radius r is proportional to the smoothing factor σ (*cf. equation 2.2*).

Table 3.2: The mean certified radii (with ± std.) of CIFAR10 classifiers learned with data augmentation and inferred by the randomized smoothing prediction rule. "certified radius (c)" denotes the correct certified radius.

training σ	0.12	0.25	0.50	1.00	1.50	2.00	3.00
min & max	$(67.8 \pm 1.9,$	$(55.4 \pm 4.8,$	$(42.4 \pm 4.8,$	$(20.8 \pm 1.3,$	$(9.8 \pm 1.3,$	$(5.4 \pm 0.9,$	$(1.2 \pm 0.8,$
class-wise acc.(%)	$93.4 \pm 1.3)$	$89.2 \pm 1.3)$	$81.9 \pm 2.2)$	$72.8 \pm 1.5)$	$61.2 \pm 3.1)$	$53.2 \pm 3.9)$	$41.0 \pm 1.0)$
certified radius	0.28 ± 0.01	0.42 ± 0.02	0.51 ± 0.03	0.50 ± 0.01	0.44 ± 0.01	0.38 ± 0.01	0.32 ± 0.01
certified radius (c)	0.34 ± 0.01	0.56 ± 0.01	0.80 ± 0.02	1.07 ± 0.01	1.25 ± 0.03	1.40 ± 0.03	1.80 ± 0.07

large σ to achieve the state-of-the-art robustness guarantees while avoiding impractical sample complexity.

In Table 3.2, we validate this point by calculating the certified radii of CIFAR10 smoothed classifiers with base classifier trained with data augmentation[2]. In this experiment, we vary the smoothing factor σ from 0.12 to 3.00, which is used simultaneously in data augmentation and randomized smoothing. When reporting their certified radius, we consider two metrics: 1) certified radius - the mean of all certified radii in the testing set, with the radius assigned to zero for wrongly-classified samples; and 2) correct certified radius - the mean of certified radii of correctly-classified samples in the testing set. We then see that with the increasing smoothing factor σ, the average certified radius of correctly-classified samples keeps rising from only 0.34 to 1.80, obtaining indeed non-trivial robustness guarantees.

On the other hand, the average certified radius of all samples climbs to around 0.5 and then decreases to 0.32. This is because the classification accuracy also drops as one uses larger σ, pushing more samples to have zero certified radius. In order to better understand the drop in accuracy and the affected examples, we provide a case study over a synthetic dataset.

[2]Throughout the chapter, all the classification results and certified radii are obtained with the open-source code provided by [156].

3.2.1 Synthetic datasets

Consider the binary-classification problem on the dataset ($\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$) given as mixture of Gaussians:

$$\mathcal{X}_1 = (\frac{1}{2} - \epsilon) \cdot \mathcal{N}(-a, \sigma_o^2) + \epsilon \cdot \mathcal{N}(ka, \sigma_o^2);$$
$$\mathcal{X}_2 = \frac{1}{2} \cdot \mathcal{N}(0, \sigma_o^2);$$

where $a, k, \sigma_o \in \mathbb{R}^+/\{0\}$. Then we have

Theorem 1. *Consider a classifier f_{train,σ_t} given as the naive-Bayes classifier obtained by training on the dataset $\mathcal{X}$ with data augmentation of variance σ_t. Let the class-wise accuracy of f_{train,σ_t} using the randomized smoothing prediction rule be given as $Acc_1(\sigma_t), Acc_2(\sigma_t)$. Then we define the bias ($\Delta(\sigma_t)$) to be the gap between class-wise accuracies ($\Delta(\sigma_t) = |Acc_1(\sigma_t) - Acc_2(\sigma_t)|$). For $k > \frac{1}{2\epsilon} - 1$, class I decision region grows in size at a rate of $O(\sigma_t^2)$ and thus the bias is large for large σ_t.*

It is quite well-known that using higher σ leads to lowering of accuracy. In general, previous works have stated the existence of a robustness-accuracy trade-off. Here, we notice another interesting and quite important problem that is created by randomized smoothing: randomized smoothing based models for high values of σ_t are biased in their predictions. Some classes are favored a lot more than others, resulting in huge difference in class-wise accuracies.

In order to better understand the extent of the bias possible, we also study the limiting case of $\sigma_o \to 0$. This allows us to effectively study large bias without having $\sigma_t \to \infty$. In particular, we consider the dataset($\mathcal{X}'$) with probability mass function :

$$\rho(0, 1) = \frac{1}{2}; \quad \rho(-a, 2) = \frac{1}{2} - \epsilon; \quad \rho(ka, 2) = \epsilon,$$

with a, k defined as before. For this new dataset, we see that

Theorem 2. *Consider a classifier f_{train,σ_t} given as the naive-Bayes classifier obtained by training on the dataset $\mathcal{X}'$ with data augmentation of variance σ_t. The bias of the*

Table 3.3: The class-wise accuracy (%) in percentile of classifiers and smoothing factors used in [31].

	CIFAR10					ImageNet				
percentile	1st	25th	50th	75th	100th	1st	25th	50th	75th	100th
$\sigma = 0.00$	78	88	91	93	96	14	66	78	88	100
0.12	0	8	15	24	100	0	36	52	66	96
0.25	0	0	0	0	72	0	2	10	20	82
0.50	0	0	0	0	98	0	0	0	0	56

classifier f_{train,σ_t} *using the randomized smoothing prediction rule is* $1-\epsilon$*, if* $k > \frac{e^2}{\epsilon} - 1$ *and* $\sigma_t \geq a\sqrt{\frac{k(k+1)}{2ln(2\epsilon(k+1)) - \frac{2k}{k+2}}}$.

To give intuitive understanding of the critical smoothing factor in Theorem 2, we fix the scale of the dataset $a(k+1)$ to be $[0, 1]$ as is common-practice in the literature [31, 156]. Then, we observe the shrinking effects happen at $\sigma \approx 0.7$ which is well within the realm of smoothing factors used in practice ([31, 156] use smoothing factors upto 1.0 for data augmentation and randomized smoothing). This idea can be extended to several more general and interesting cases: a multi-class case giving accuracy $\frac{1}{c} + \epsilon$ by having class 1 with the same distribution and the rest of the classes with distributions similar to that of class 2's; and a binary-class case where adopting data augmentation does not change the optimal solution but the subsequent randomized smoothing inference still gets low accuracy for a high enough smoothing factor σ.

3.2.2 Real-life datasets

In the existing literature, randomized smoothing remains a legitimate way of providing adversarial robustness. However, the results on the synthetic datasets suggest randomized smoothing is biased towards some classes. In order to see if the bias is present in real-life datasets we consider a new metric, namely the min and max class-wise accuracy, where we calculate separately for each class their classification accuracy and report the minimum and the maximum. In Table 3.2 we give the performance of randomized smoothing based classifiers under the new metric. With this metric, one can then readily see that despite the increasing trend in certified radii, the class-wise

accuracies becomes more imbalanced at higher smoothing factor σ. Specifically, when the smoothing factor $\sigma = 0.12$, the smoothed network with base classifier being trained by data augmentation with the same magnitude of Gaussian noise classifies "cat" samples with 67% accuracy and "automobile" samples with 92% accuracy. However, when $\sigma = 1.00$, this gap evolves to 22% accuracy ("cat") versus 68% accuracy ("ship"). This comes as an unpleasant surprise since it essentially means despite the current success of randomized smoothing in adversarial robustness, the method can lead to biased predictions, causing fairness issues.

As remarked earlier, a randomized smoothing model differs from other models in two phases, data augmentation during training and smoothing during inference. As the statistical guarantees given by randomized smoothing depend on the smoothing during inference, we focus on its role in producing the bias. Before proceeding, we verify that the bias problem still persists in the absence of augmentation during training. We conduct the smoothing experiments on the pretrained models provided by Cohen et al. (2019). In Table 3.3, we report the smoothing factors σ and corresponding class-wise accuracies (sorted ascendingly) in percentile of [1st,25th,50th,75th,100th]. That is, the 1st and 100th in the percentile correspond to the lowest (min) and highest (max) class-wise accuracy, respectively For CIFAR10, the [25th, 75th] percentile corresponds to the [3rd, 8th] lowest per-class accuracy. One can then see that originally more than 3/4 of the classes in datasets have reasonable accuracy, which decreases as σ goes bigger. Eventually, when $\sigma = 0.5$, more than 3/4 of the classes have 0 accuracy. Notably, $\sigma = 0.5$ is a reasonable number under the current randomized smoothing regime since the largest sigma used by [31] and [156] is 1.0. Thus, we see that randomized smoothing produces biased results even in the absence of data augmentation during training. In the next section, we analyze how biased predictions are caused by randomized smoothing depending on the geometry of the underlying data distribution.

3.3 Theoretical characterization of the shrinking phenomenon

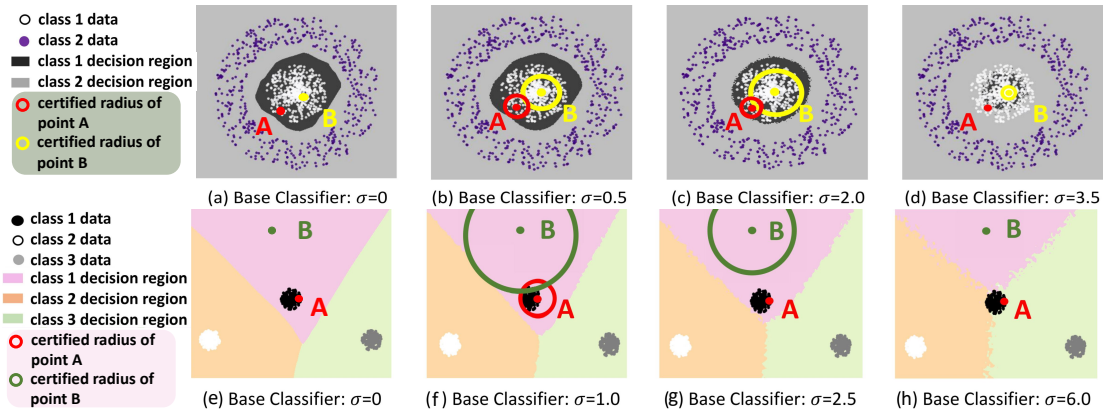


Figure 3-1: The 1st row shows examples of **bounded** decision regions for smoothed classifiers. The 2nd row shows examples of **semi-bounded** decision regions. The class 1 decision regions shrink as the smoothing factor σ increases from left to right. In case (h) with larges σ, the decision region has shrunk so much that class 1 data are completely misclassified. We also plot the certified radius (equation 2.2) of point A and B and show that it may decrease as σ increases.

Before we start our theoretical characterization, we first give a visual inspection of how randomized smoothing can change the decision regions. Specially, Figure 3-1 illustrates two toy examples, in which the decision regions of class 1 data (the dark green region in the first row and the pink region in the second row) shrink with larger smoothing factors σ. As consequences of the shrinkage, the class-wise accuracy for class 1 data drops drastically, leading to the biased prediction.

Indeed, in this section, we aim to take a close look at this *shrinking phenomenon* of randomized smoothing, uncovering the fundamental problem of the technique. Moreover, we conduct a rigorous study providing also the bounds of extreme values, beyond which the shrinking phenomenon will happen. Our results are tight and prove the prevalence of such phenomena. In order to facilitate this analysis we perform the following reductions.

Problem Reductions. By the definition of randomized smoothing, the smoothed function depends on the base classifier only through the indicator function f. As the

smoothed function f_σ only depends on the partitioning of the input space created by the base classifier g, we shift our focus from the output of g to how it partitions the input space, *i.e.*, we are interested in characterizing all possible partitions of the input space that can lead to biased prediction as one applies randomized smoothing with a high σ. As it is hard to measure a decrease in accuracy directly from the geometry of the classifier, we approximate the decrease in accuracy using the mismatch in partitions of input space provided by f and by f_σ.

However, the problem of characterizing the partitions of the space into multiple classes is intractable. So we instead focus on tracking the behaviour of the decision boundary of a single class with respect to randomized smoothing. Without loss of generality, we set the concerned class as class 1. In this case, we analyze the misclassification rate for class 1 by the region size of the input space that is partitioned as class 1 under f but not under f_σ. Considering that for any $x \in \mathbb{R}^d$, the necessary condition for it to be classified as class 1 is to have $f_\sigma(x)_1 \geq \frac{1}{c}$, so we do a worst-case analysis by assuming the reformed class 1 partition is defined by exactly $f_\sigma(x)_1 \geq \frac{1}{c}$. If this overestimated reformed class 1 partition is still smaller than the original, then for sure the actual misclassification rate will be higher than the analysis herein.

Problem Formulation. We formulate our problem as to characterize the "decision regions" that will shrink or drift after applying randomized smoothing. Formally, the decision region $\mathcal{D}$ of class 1 data is determined by the classifier f via $\mathcal{D} = \{x \mid f(x)_1 = 1\}$. By adopting randomized smoothing, we obtain $f_\sigma(x) = \int_{x' \in \mathbb{R}^d} f(x')p(x')dx'$ with the decision region denoted by $\mathcal{D}_\sigma = \{x \mid (f_\sigma(x))_1 \geq \frac{1}{c}\}$. The scope of this section is to investigate under what conditions (*w.r.t.* the classifier and smoothing factor σ) will the shrinking happen. On the whole, the shrinking effect depends highly on the geometry of the data distribution. However, considering the intractable numbers of possible decision region geometry, we will only discuss here two major classes of the geometries (*bounded* in Section 3.3.1 and *semi-bounded* in Section 3.3.2) for multidimensional data (*i.e.* $d > 1$). We supplement $d = 1$ discussions in the Appendix B.1.1. All the supporting proofs are deferred to the appendix.

3.3.1 Bounded decision region

In this section, we aim at proving the shrinking side-effects incurred by the smoothing filter when the decision region is bounded. Formally, we say a decision region is bounded and shrinks according to the following definition:

Definition 3.3.1 (Bounded Decision Regions)**.** If the decision region (disconnected or connected) of class 1 data is a bounded set in the Euclidean space (can be bounded by a ball of finite radius), then we call these decision regions bounded decision regions.

We denote the smallest ball that contains the original decision region of f by $S_{\mathcal{D}}$ ($\mathcal{D} \subseteq S_{\mathcal{D}}$). Similarly, we let the smallest ball that contains the smoothed decision region (the decision region of smoothed classifier) be $S_{\mathcal{D}_\sigma}$ ($\mathcal{D}_\sigma \subseteq S_{\mathcal{D}_\sigma}$).

Definition 3.3.2 (Shrinking of Bounded Decision Regions)**.** A bounded decision region is considered to have shurnk after applying smoothing filters if the radius R_σ of $S_{\mathcal{D}_\sigma}$ is strictly smaller than the radius R of $S_{\mathcal{D}}$, *i.e.* $R_\sigma < R$, where $S_{\mathcal{D}}$ and $S_{\mathcal{D}_\sigma}$ are the smallest balls containing the original decision region and the smoothed decision region, respectively.

For randomized smoothing, we observe that

Corollary 1. *The smallest ball $S_{\mathcal{D}_\sigma}$ containing the smoothed decision region is contained within the smoothed version of $S_{\mathcal{D}}$, i.e. $S_{\mathcal{D}_\sigma} \subseteq (S_{\mathcal{D}})_\sigma$.*

Proof. As we have $\mathcal{D} \subseteq S_{\mathcal{D}}$, from Lemma A.2 we get $\mathcal{D}_\sigma \subseteq (S_{\mathcal{D}})_\sigma$. Then by isotropy we have that $(S_{\mathcal{D}})_\sigma$ is also a ball centered at the same point as $S_{\mathcal{D}}$. As $S_{\mathcal{D}_\sigma}$ is the smallest ball containing $\mathcal{D}_\sigma$, we have that $S_{\mathcal{D}_\sigma} \subseteq (S_{\mathcal{D}})_\sigma$. □

Theorem 3. *A bounded decision region shrinks after applying Gaussian smoothing filters with large σ, i.e. if $\sigma > \frac{R\sqrt{c}}{\sqrt{2(d-1)}}$, then $R_\sigma < R$, where R and R_σ are the radii of $S_{\mathcal{D}}$ and $S_{\mathcal{D}_\sigma}$, the smallest balls bounding the original decision region and the smoothed decision region, respectively.*

Proof. Considering the ball $S_{\mathcal{D}}$, we see that from Corollary 1, $\mathcal{D}_\sigma \subseteq (S_{\mathcal{D}})_\sigma$. Thus, we see that by the definition of radius $R_{\mathcal{D}_\sigma} \leq R_{(S_{\mathcal{D}})_\sigma}$. It is sufficient to show that for

large σ, $R_{(S_{\mathcal{D}})_\sigma} < R_{S_{\mathcal{D}}}$. Then we observe that due to the isotropic nature of Gaussian smoothing, $(S_{\mathcal{D}})_\sigma$ is also a sphere concentric to $S_{\mathcal{D}}$. So, it is sufficient to show that for a point x at distance $R_{S_{\mathcal{D}}}$ from the center x_0 of the sphere, $f_\sigma(x)_1 < \frac{1}{c}$.
Without loss of generality consider $\mathcal{D}$ to be the origin-centered sphere of radius R and $x = [0, \ldots, 0, R]^T$. It is sufficient to show for large σ $f_\sigma(x)_1 < \frac{1}{c}$. By definition A.1.2, we have

$$\begin{aligned} f_\sigma(x)_1 &= \int_{x' \in \mathbb{R}^d} f(x')_1 p(x') dx' \\ &= \int_{\|x'\|_2 \leq R} (2\pi)^{-\frac{d}{2}} |\Sigma|^{-\frac{1}{2}} e^{-\frac{1}{2}(x'-x)^T \Sigma^{-1}(x'-x)} dx' \\ &= \int_{\|x'\|_2 \leq R} (2\pi\sigma^2)^{-\frac{d}{2}} e^{-\frac{(x'-x)^T(x'-x)}{2\sigma^2}} dx'. \end{aligned} \tag{3.1}$$

Then substituting the value of x, we get the equation.

$$\begin{aligned} f_\sigma(x)_1 &= \int_{\|x'\|_2 \leq R} (2\pi\sigma^2)^{-\frac{d}{2}} e^{\frac{\sum_{i=1}^{d-1} x_i'^2 + (x_d' - R)^2}{2\sigma^2}} dx' \\ &= \int_{-R}^{R} \int_{\sum_{k=1}^{d-1} {x_k'}^2 \leq R^2 - {x_d'}^2} (2\pi\sigma^2)^{-\frac{d}{2}} e^{-\frac{\sum_{k=1}^{d-1}(x_k' - x_k)^2}{2\sigma^2}} dx_1' \ldots dx_{d-1}' e^{-\frac{(x_d' - x_d)^2}{2\sigma^2}} dx_d' \\ &< \int_{-R}^{R} \int_{\sum_{k=1}^{d-1} {x_k'}^2 \leq R^2} (2\pi\sigma^2)^{-\frac{d}{2}} e^{-\frac{\sum_{k=1}^{d-1}(x_k' - x_k)^2}{2\sigma^2}} dx_1' \ldots dx_{d-1}' e^{-\frac{(x_d' - x_d)^2}{2\sigma^2}} dx_d' \\ &= \left(\int_{-R}^{R} (2\pi\sigma^2)^{-\frac{1}{2}} e^{-\frac{(x_d' - x_d)^2}{2\sigma^2}} dx_d'\right)\left(\int_{\sum_{k=1}^{d-1} {x_k'}^2 \leq R^2} (2\pi\sigma^2)^{-\frac{d-1}{2}} e^{-\frac{\sum_{k=1}^{d-1}(x_k' - x_k)^2}{2\sigma^2}} dx_1' \ldots dx_{d-1}'\right) \\ &= (\Phi(\frac{2R}{\sigma}) - \Phi(0)) \cdot Q(\frac{d-1}{2}, \frac{R^2}{2\sigma^2}) \\ &< \frac{1}{2} \cdot Q(\frac{d-1}{2}, \frac{R^2}{2\sigma^2}). \end{aligned}$$

Using Lemma A.3 we get that for $d \geq 3$, if $\frac{R^2}{2\sigma^2} \leq \frac{d-1}{c}$, then we have $\frac{1}{2} \cdot Q(\frac{d-1}{2}, \frac{R^2}{2\sigma^2}) < \frac{1}{c}$. Now, $\frac{R^2}{2\sigma^2} < \frac{d-1}{c}$ gives

$$\sigma > \frac{R\sqrt{c}}{\sqrt{2(d-1)}}.$$

□

Analysis of bounded decision regions with randomized smoothing. As we have proven that any bounded decision region shrinks after applying randomized

smoothing filters, we will investigate in this part of the chapter *how fast* the decision region (quantified by R_σ) shrinks/vanishes. From Corollary 1, we have that the smallest ball $S_{\mathcal{D}_\sigma}$ containing the smoothed decision region is contained within the smoothed version of $S_{\mathcal{D}}$. Therefore we only consider the worst case when we have a ball-like decision region. Without loss of generality, we consider a case when the decision region of class 1 data characterized by the network function is exactly $\{x \in \mathbb{R}^d \mid \|x\|_2 \leq R\}$.

Theorem 4 (Vanishing Rate in the Ball-like Decision Region Case)**.** *The decision region of class 1 data vanishes when smoothing factor* $\sigma_{van} > \frac{R\sqrt{c}}{\sqrt{d}}$.

Proof. Noticing that the surface area of a d-dimensional ball of radius r is proportional to r^{d-1}, we can therefore write out the probability of the point at the origin be classified as class 1 as

$$\begin{aligned}
q(R,d,\sigma) &= \frac{\int_0^R r^{d-1}(\frac{1}{2\pi\sigma^2})^{\frac{d}{2}} e^{-\frac{r^2}{2\sigma^2}} dr}{\int_0^\infty r^{d-1}(\frac{1}{2\pi\sigma^2})^{\frac{d}{2}} e^{-\frac{r^2}{2\sigma^2}} dr} \\
&= \frac{\int_0^R r^{d-1} e^{-\frac{r^2}{2\sigma^2}} dr}{\int_0^\infty r^{d-1} e^{-\frac{r^2}{2\sigma^2}} dr} \\
&\underline{\underline{t=\frac{r^2}{2\sigma^2}}} \ \frac{\int_0^{\frac{R^2}{2\sigma^2}} (2\sigma^2 t)^{\frac{d-1}{2}} e^{-t} \sigma^2 (2\sigma^2 t)^{-\frac{1}{2}} dt}{\int_0^\infty (2\sigma^2 t)^{\frac{d-1}{2}} e^{-t} \sigma^2 (2\sigma^2 t)^{-\frac{1}{2}} dt} \\
&= \frac{\int_0^{\frac{R^2}{2\sigma^2}} t^{\frac{d}{2}-1} e^{-t} dt}{\int_0^\infty t^{\frac{d}{2}-1} e^{-t} dt} \\
&= Q(\frac{d}{2}, \frac{R^2}{2\sigma^2}).
\end{aligned}$$

Now let $\sigma = \sqrt{\frac{c}{d}} R$ yields $q(R, d, \sqrt{\frac{c}{d}} R) = Q(\frac{d}{2}, \frac{d}{2c})$. By Lemma A.3, we then have $Q(\frac{d}{2}, \frac{d}{2c}) < \frac{1}{c}$, implying the decision region of class 1 data has already vanished and making $\sigma = \sqrt{\frac{c}{d}} R$ an upper bound of the vanishing smoothing factor. □

We validate Theorem 4 for binary classification ($c = 2$) by substituting R by $R = 1$ and plot the shrinking rate (the derivatives of R_σ with respect to σ) of the decision region as a function of the smoothing factor σ for different input data dimensions $d = \{3, 8, 20, 30, 40, 50\}$ in Figure 3-2. Notably, the x-axis in Figure 3-2 is the varying

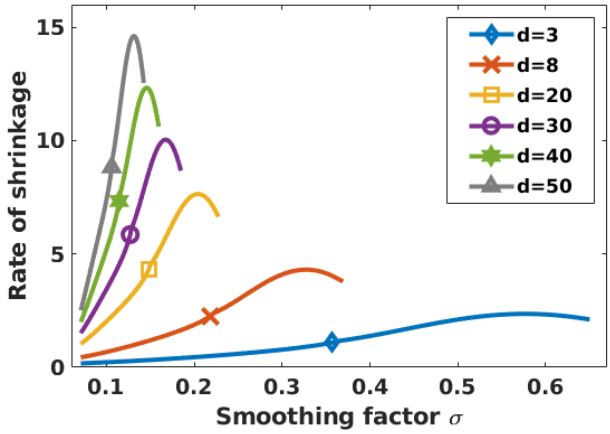


Figure 3-2: The shrinking rate of the decision region quantified by R_σ for different input data dimension d.

smoothing factor σ and the y-axis is the rate of the shrinkage concerning class 1 decision region. We then see that overall the region vanishes at smaller smoothing factor σ_{van} with the growing dimension. For example, the shrinking rate curve stops at smoothing factor $\sigma_{\text{van}} = 0.651$ when $d = 3$ but at smoothing factor $\sigma_{\text{van}} = 0.141$ when $d = 50$. We collect these vanishing smoothing factors with different data dimensions and compare with the theoretical lower bounds found in Theorem 4 in the appendix to demonstrate the tightness of our theoretical lower bound. In a multi-class case, the certifiability and prediction do not follow the same setting as in [31]. For the certifiability, the effective number of classes is 2 as [31] treats it as a one vs all setting. Therefore one would be unable to certify any radius with some smoothing factor $\sigma < \sigma_{\text{van}}$ in the multi-class case. We further elaborate on this point about certifiability in Section 3.3.3.

3.3.2 Semi-bounded decision region

In this section, we discuss the case when the decision region is semi-bounded and is not a half-space. Formally, we say a decision region is semi-bounded and shrinks according to the following definitions:

Definition 3.3.3 (Semi-bounded Decision Regions)**.** If a decision region is not bounded and there exists a half-space $\mathcal{H}$ (decided by a hyperplane) that contains the unbounded

decision region, then we call it semi-bounded decision region. We say a semi-bounded decision region is bounded in v-direction if there $\exists k \in \mathbb{R}/\infty$ such that for $\forall x \in \mathcal{D}$, $v^T x < k$.

An illustrative example of semi-bounded decision regions is shown as Figure 3-1, where we have 3 clusters of data points denoting three different classes' data and their decision regions. Observing the change in the decision region of class 1, we define "shrinking" as

Definition 3.3.4 (Shrinking of Semi-bounded Decision Regions)**.** A semi-bounded decision region bounded in v-direction is distinguished as shrinked along the direction after applying smoothing filters if the upper bound of projections of the decision region onto direction v shrinks, *i.e.* $\Upsilon^v_{\mathcal{D}_\sigma} < \Upsilon^v_{\mathcal{D}}$, where $\Upsilon^v_{\mathcal{D}} = \max_{x\in\mathcal{D}} v^T x, \Upsilon^v_{\mathcal{D}_\sigma} = \max_{x\in\mathcal{D}_\sigma} v^T x$.

With this definition of shrinking of semi-bounded decision regions, we demonstrate in the following that any "narrow" semi-bounded decision region bounded in v-dimension will shrink along the direction (*cf.* Figure 3-1(e-h)). We quantify the size of a decision region as follows:

Definition 3.3.5 (θ, v-Bounding Cone for a Decision Region)**.** A θ, v cone is defined as a right circular cone $\mathcal{C}$ with axis along $-v$ and aperture 2θ. Then we define the θ, v-bounding cone $\mathcal{C}^{\mathcal{D}}_{\theta,v}$ for $\mathcal{D}$ as the θ, v cone that has the smallest projection on v and contains $\mathcal{D}$, *i.e.*, $\mathcal{C}^{\mathcal{D}}_{\theta,v} = \arg\min_{\mathcal{D}\subseteq\mathcal{C}_{\theta,v}} \Upsilon^v_{\mathcal{C}_{\theta,v}}$.

Theorem 5. *A semi-bounded decision region that has a narrow bounding cone shrinks along v-direction after applying Gaussian smoothing filters with high σ, i.e. if the region admits a bounding cone $\mathcal{C}^{\mathcal{D}}_{\theta,v}$ with $\tan(\theta) < \sqrt{\frac{(d-1)}{2c\log(c-1)}}$, then for $\sigma > (\Upsilon^v_{\mathcal{C}^{\mathcal{D}}_{\theta,v}} - \Upsilon^v_{\mathcal{D}})\tan(\theta)\sqrt{\frac{c}{d-1}} \cdot \frac{2(d-1)}{(d-1)-2\tan^2(\theta)c\log(c-1)}$, $\Upsilon^v_{\mathcal{D}_\sigma} < \Upsilon^v_{\mathcal{D}}$.*

Proof. In this derivation we assume without loss of generality, $v = [0, \ldots, 0, 1]^T \in \mathbb{R}^d$ (It is always possible to orient the axis to make this happen). From Corollary A.1, we can see that $\mathcal{D}_\sigma \subseteq (\mathcal{C}^{\mathcal{D}}_{\theta,v})_\sigma$ which gives us $\Upsilon^v_{\mathcal{D}_\sigma} = \max_{x\in\mathcal{D}_\sigma} v^T x \leq \max_{x\in(\mathcal{C}^{\mathcal{D}}_{\theta,v})_\sigma} v^T x = \Upsilon^v_{(\mathcal{C}^{\mathcal{D}}_{\theta,v})_\sigma}$. Then to show that $\Upsilon^v_{\mathcal{D}_\sigma} < \Upsilon^v_{\mathcal{D}}$ it is sufficient to show that $\Upsilon^v_{(\mathcal{C}^{\mathcal{D}}_{\theta,v})_\sigma} < \Upsilon^v_{\mathcal{D}}$.

We observe that we only need to check the point x on the axis of the cone at distance $\Upsilon^v_{\mathcal{C}^{\mathcal{D}}_{\theta,v}} - \Upsilon^v_{\mathcal{D}}$ from the tip x_0 of the cone, *i.e.*, $x = x_0 - (\Upsilon^v_{\mathcal{C}^{\mathcal{D}}_{\theta,v}} - \Upsilon^v_{\mathcal{D}})v$. If x is not classified as Class 1 then by Lemma A.5, we have that

$$\begin{aligned}
\Upsilon^v_{(\mathcal{C}^{\mathcal{D}}_{\theta,v})_\sigma} < v^T x &= v^T(x_0 - (\Upsilon^v_{\mathcal{C}^{\mathcal{D}}_{\theta,v}} - \Upsilon^v_{\mathcal{D}})v) \\
&= v^T x_0 - (\Upsilon^v_{\mathcal{C}^{\mathcal{D}}_{\theta,v}} - \Upsilon^v_{\mathcal{D}})v^T v \\
&= \Upsilon^v_{\mathcal{C}^{\mathcal{D}}_{\theta,v}} - (\Upsilon^v_{\mathcal{C}^{\mathcal{D}}_{\theta,v}} - \Upsilon^v_{\mathcal{D}}) = \Upsilon^v_{\mathcal{D}}
\end{aligned}$$

From the above argument and the definition of the decision boundary we see that if $f_\sigma(x)_1 < \frac{1}{c}$, then $\Upsilon^v_{\mathcal{D}_\sigma} < \Upsilon^v_{\mathcal{D}}$. Without loss of generality we let x_0 be the origin. By definition A.1.2, we have

$$\begin{aligned}
f_\sigma(x)_1 &= \int_{x' \in \mathbb{R}^d} f(x')_1 p(x') dx' \\
&= \int_{x'_d + \|x'\| cos(\theta) \leq 0} (2\pi\sigma^2)^{-\frac{d}{2}} e^{-\frac{(x'-x)^T(x'-x)}{2\sigma^2}} dx' \\
&= (2\pi\sigma^2)^{-\frac{d}{2}} \int_{-\infty}^{0} \int_{\Sigma_{k=1}^{d-1} {x'_k}^2 \leq tan^2(\theta) {x'_d}^2} e^{-\frac{\sum_{k=1}^{d-1}(x'_k - x_k)^2}{2\sigma^2}} dx'_1 \dots dx'_{d-1} e^{-\frac{(x'_d - x_d)^2}{2\sigma^2}} dx'_d \\
&= (2\pi\sigma^2)^{-\frac{d}{2}} \int_{-\infty}^{0} \int_{\Sigma_{k=1}^{d-1} {x'_k}^2 \leq tan^2(\theta) {x'_d}^2} e^{-\frac{\sum_{k=1}^{d-1} {x'_k}^2}{2\sigma^2}} dx'_1 \dots dx'_{d-1} e^{-\frac{(x'_d - x_d)^2}{2\sigma^2}} dx'_d \\
&= (2\pi\sigma^2)^{-\frac{1}{2}} \int_{-\infty}^{0} q(|x'_d \tan(\theta)|, d-1, \sigma) e^{-\frac{(x'_d - x_d)^2}{2\sigma^2}} dx'_d \\
&= (2\pi\sigma^2)^{-\frac{1}{2}} \int_{-\infty}^{0} Q(\frac{d-1}{2}, \frac{tan^2(\theta) {x'_d}^2}{2\sigma^2}) e^{-\frac{(x'_d - x_d)^2}{2\sigma^2}} dx'_d
\end{aligned}$$

Substitute $X_d = \frac{x_d}{\sigma}, X'_d = \frac{x'_d}{\sigma}$,

$$f_\sigma(x)_1 = (2\pi)^{-\frac{1}{2}} \int_{-\infty}^{0} Q(\frac{d-1}{2}, \frac{tan^2(\theta) {X'_d}^2}{2}) e^{-\frac{(X'_d - X_d)^2}{2}} dX'_d$$

Let $M \leq \sqrt{\frac{d-1}{c\tan^2(\theta)}}, k = \frac{M}{X_d}$,

$$\begin{aligned}
f_\sigma(x)_1 &= (2\pi)^{-\frac{1}{2}} \int_M^0 Q(\frac{d-1}{2}, \frac{tan^2(\theta)X_d'^2}{2}) e^{-\frac{(X_d'-X_d)^2}{2}} dX_d' \\
&+ (2\pi)^{-\frac{1}{2}} \int_{-\infty}^M Q(\frac{d-1}{2}, \frac{tan^2(\theta)X_d'^2}{2}) e^{-\frac{(X_d'-X_d)^2}{2}} dX_d' \\
&< (\Phi(-X_d) - \Phi(M - X_d)) Q(\frac{d-1}{2}, \frac{\tan^2(\theta)M^2}{2}) + \Phi(M - X_d) \\
&< \frac{\Phi(-X_d) - \Phi(M - X_d)}{c} + \Phi(M - X_d) \\
&= \frac{1}{c} + \frac{(c-1)\Phi((k-1)X_d) - \Phi(X_d)}{c}.
\end{aligned}$$

Then we see that using Lemma A.6, we see that we see that if $e^{\frac{X_d^2((k-1)^2-1)}{2}} \geq c - 1$ then $(c-1)\Phi((k-1)X_d) \leq \Phi(X_d)$. So, we need to satisfy the inequalities for some k:

$$\sqrt{\frac{2\log(c-1)}{(k-1)^2 - 1}} \leq -X_d \leq \sqrt{\frac{d-1}{k^2 c\tan^2(\theta)}}.$$

This is only possible if for some k, we have $\sqrt{\frac{2\log(c-1)}{(k-1)^2-1}} \leq \sqrt{\frac{d-1}{k^2 c\tan^2(\theta)}}$ or $\tan(\theta) \leq \sqrt{\frac{d-1}{2c\log(c-1)}} \cdot \sqrt{1 - \frac{2}{k}}$. So, we need that

$$\tan(\theta) < \sqrt{\frac{d-1}{2c\log(c-1)}}.$$

Then we see that giving the cone is narrow enough, we have the required shrinking if we have X_d satisfies the inequalities for some k. So, we see that if we have $-X_d = \sqrt{\frac{d-1}{k^2 c\tan^2(\theta)}}$ for some k such that $\tan(\theta) \leq \sqrt{\frac{d-1}{2c\log(c-1)}} \cdot \sqrt{1 - \frac{2}{k}}$ is satisfied. So, we need that $\frac{-x_d}{\sigma} = \sqrt{\frac{d-1}{k^2 c\tan^2(\theta)}}$ for some suitable k. Thus we need $\sigma = -x_d \tan(\theta)\sqrt{\frac{c}{d-1}}k$ for some suitable k. Including the constraint on k and substituting the value for x_d, we get that shrinking always happens for

$$\sigma \geq (\Upsilon^v_{\mathcal{C}^{\mathcal{D}}_{\theta,v}} - \Upsilon^v_{\mathcal{D}}) \tan(\theta) \sqrt{\frac{c}{d-1}} \cdot \frac{2(d-1)}{(d-1) - 2\tan^2(\theta)c\log(c-1)}.$$

□

Concretely, the narrowness condition (the larger the easier to fulfill) of the cone for MNIST dataset [98] relaxes to $0.43\pi = 76.7°$, meaning that if any single class's decision region can be bounded by a θ, v cone with θ being less than $76.7°$, then shrinking effect happens. Correspondingly, this narrowness condition for CIFAR10 dataset [98] is $0.46\pi = 83.2°$ and $0.42\pi = 75.2°$ for ImageNet dataset [155]. Notably, for binary classification tasks ($c = 2$), according to Theorem 5, the condition for shrinking reduces to $\tan(\theta) < \infty$ that implies $\theta < \pi/2$. In other words, when there are only two classes, as long as the semi-decision region is not a half-space, it **will** shrink.

Analysis of the semi-bounded case with randomized smoothing. As in Section 3.3.1, we conduct the analysis using the worst-case ball-like bounded decision region, here we correspondingly consider a solid right circular cone along the v direction. The shrinkage in this case serves as a non-trivial lower bound. Without loss of generality, we consider a θ, v solid right circular cone $\{x \in \mathbb{R}^d \mid v^T x - \|v\|\|x\| cos(\theta) \leq 0\}$ as the decision region $\mathcal{D}$ of class 1 data, where $-v = [0, \ldots, 0, 1]^T \in \mathbb{R}^d$. Since the semi-bounded decision region is unbounded and will shrink but will not vanish, we emphasize in this section only on giving the shrinking rate with respect to the smoothing factor σ, the number of classes c, the angle θ, and the data dimension d with randomized smoothing. Two major theorems regarding the shrinking rate in the solid cone-like decision region are:

Theorem 6. *The shrinkage of class 1 decision region is proportional to the smoothing factor, i.e.* $\Upsilon^v_{\mathcal{D}} - \Upsilon^v_{\mathcal{D}_\sigma} \propto \sigma$.

Proof. In this case we assume a cone-like decision region which can be represented as $\mathcal{D} = \{x \in \mathbb{R}^d \mid v^T x + \|v\|\|x\| cos(\theta) \leq 0\}$ with $v = [0, \ldots, 0, 1]^T$ without loss of generality. By Lemma A.5, we see that in order to get bounds on $\Upsilon^v_{\mathcal{D}_\sigma}$, we only need to analyze the value of $f_\sigma(x)_1$ for points x along the axis of the cone. Then we see that for a general point $x = av$ along the axis of the cone, using the same ideas as in proof of Theorem 5, we have

$$
\begin{aligned}
f_\sigma(x)_1 &= \int_{x' \in \mathbb{R}^d} f(x')_1 p(x') dx' \\
&= (2\pi\sigma^2)^{-\frac{d}{2}} \int_{-\infty}^{0} \int_{\sum_{k=1}^{d-1} {x'_k}^2 \leq tan^2(\theta) {x'_d}^2} e^{-\frac{\sum_{k=1}^{d-1} {x'_k}^2}{2\sigma^2}} dx'_1 \dots dx'_{d-1} e^{-\frac{(x'_d - a)^2}{2\sigma^2}} dx'_d \\
&= (2\pi\sigma^2)^{-\frac{1}{2}} \int_{-\infty}^{0} Q(\frac{d-1}{2}, \frac{tan^2(\theta) {x'_d}^2}{2\sigma^2}) e^{-\frac{(x'_d - a)^2}{2\sigma^2}} dx'_d \\
&\text{Substitute } A = \frac{a}{\sigma}, x'_d = \frac{x'_d}{\sigma} \\
&= (2\pi)^{-\frac{1}{2}} \int_{-\infty}^{0} Q(\frac{d-1}{2}, \frac{tan^2(\theta) {x'_d}^2}{2}) e^{-\frac{(x'_d - A)^2}{2}} dx'_d \\
&= f_1(Av)_1 = f_1(\frac{1}{\sigma}x)_1.
\end{aligned}
$$

Using the equation above we see that for smoothing by a general σ,

$$
\begin{aligned}
\Upsilon_{\mathcal{D}_\sigma} &= \sup_{x | f_\sigma(x) \geq \frac{1}{c}} v^T x = \sup_{x | f_1(\frac{1}{\sigma}x) \geq \frac{1}{c}} v^T x = \sup_{x' | f_1(x') \geq \frac{1}{c}} v^T (\sigma x') \\
&= \sigma \sup_{x' | f_1(x') \geq \frac{1}{c}} v^T x' = \sigma \Upsilon_{\mathcal{D}_1}.
\end{aligned}
$$

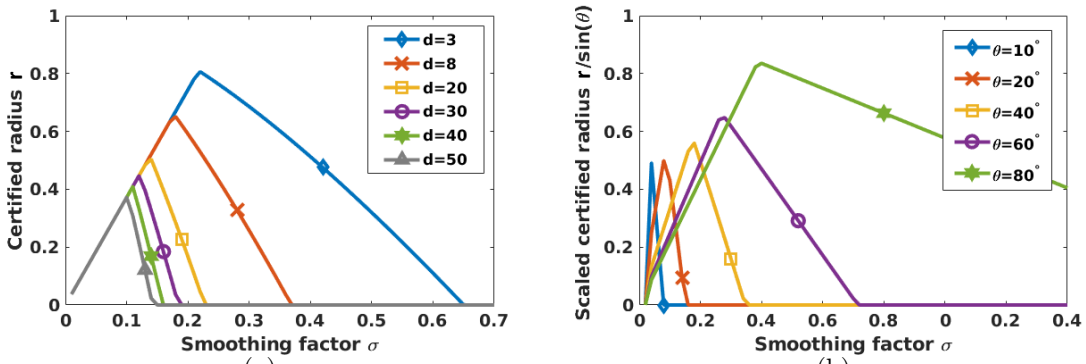
In this case we have $\Upsilon_{\mathcal{D}} = 0$ by construction, so $\Upsilon_{\mathcal{D}} - \Upsilon_{\mathcal{D}_\sigma} = 0 - \sigma \Upsilon_{\mathcal{D}_1} = \sigma \cdot (-\Upsilon_{\mathcal{D}_1}) \propto \sigma$. □

With the above Theorem 6, we can fix the smoothing factor to $\sigma = 1$ and further obtain a lower bound of the shrinking rate *w.r.t* c, θ, and d:

Theorem 7. *The shrinking rate of class 1 decision region is at least* $\sqrt{\frac{d-1}{c \tan^2(\theta)}} \cdot \frac{(d-1) - 2\tan^2(\theta) c \log(c-1)}{2(d-1)}$, *i.e.* $\frac{\Upsilon^v_{\mathcal{D}_\sigma} - \Upsilon^v_{\mathcal{D}_{\sigma+\delta}}}{\delta} > \sqrt{\frac{d-1}{c \tan^2(\theta)}} \cdot \frac{(d-1) - 2\tan^2(\theta) c \log(c-1)}{2(d-1)}$.

Proof. As in Theorem 6, we assume a cone at origin along $v = [0, \dots, 0, 1]^T$ given by $\mathcal{D} = \{x \in \mathbb{R}^d \mid v^T x + \|v\| \|x\| cos(\theta) \leq 0\}$. Following the same proof idea as Theorem 6, we see that the rate is given by the value $-\Upsilon_{\mathcal{D}_1}$. So, we try to get a bound on the value of $-\Upsilon_{\mathcal{D}_1}$. To establish a lower bound we show that for the point $x = av$, $f_1(x)_1 < \frac{1}{c}$. Then by Lemma A.5 we have $\Upsilon_{\mathcal{D}_1} < a$ or $-\Upsilon_{\mathcal{D}_1} > -a$.

Using the same procedure as in the proof of Theorem 5, we get that if x satisfies the



(a) (b)

Figure 3-3: (a) The certified radius r of the point at the origin for different input data dimension d; (b) The scaled certified radius $\frac{r}{\sin(\theta)}$ of a point on the axis v for cones with different apertures (2θ).

two inequalities

$$\sqrt{\frac{2\log(c-1)}{(k-1)^2-1}} \leq -v^T x \leq \sqrt{\frac{d-1}{k^2 c\tan^2(\theta)}}$$

for suitable real k, then we have $f_1(x)_1 < \frac{1}{c}$. So, we need $v^T x = -\sqrt{\frac{d-1}{k^2 c\tan^2(\theta)}}$ for some k such that $\sqrt{\frac{2\log(c-1)}{(k-1)^2-1}} \leq x \leq \sqrt{\frac{d-1}{k^2 c\tan^2(\theta)}}$. The constraint on k can be re-written as $k \geq \frac{2(d-1)}{(d-1)-2\tan^2(\theta)c\log(c-1)}$. Taking k to be lower bound, we get that for

$$-a = -v^T x = \sqrt{\frac{d-1}{c\tan^2(\theta)}} \cdot \frac{(d-1)-2\tan^2(\theta)c\log(c-1)}{2(d-1)}$$

$f_1(x)_1 \leq \frac{1}{c}$. So, we get that the rate is $-\Upsilon_{\mathcal{D}_1} \geq -a \geq \sqrt{\frac{d-1}{c\tan^2(\theta)}} \cdot \frac{(d-1)-2\tan^2(\theta)c\log(c-1)}{2(d-1)}$.

□

3.3.3 Remarks on certified radii

In the case of bounded decision region, the point at the origin has the highest probability to be classified as class 1 (see Lemma A.4). Therefore when it has less than 0.5 probability to be classified as class 1, the decision region vanishes and no point can be certified (certified radius $r = 0$). Specifically, Figure 3-3(a) describes the certified radius r of the point at the origin using equation 2.2 as a function of the smoothing

Table 3.4: The minimum and maximum class-wise accuracy (%) of CIFAR10 classifiers learned with data augmentation and inferred by the randomized smoothing prediction rule. The smaller the gap between the maximum and the minimum class-wise accuracies is, the better.

#Augmentation Points	1 (standard)		10		25		50	
class-wise acc.	min	max	min	max	min	max	min	max
$\sigma = 0.12$	67	94	76	96	78	96	68	97
0.25	55	90	68	92	65	93	48	84
0.50	46	84	51	84	52	81	0	87
1.00	22	73	28	74	27	72	3	64

factor σ and shows that the maximum certified radius (the peak) decreases with the increasing dimension. We include complete certified radius behavioral plots for different dimensions in the Appendix B.1.2. As training samples are normally scaled in practice, they lie within a ball of radius $R \leq \sqrt{d}/2$. According to Theorem 4, for this ball, the upper bound of σ_{van} is $1/\sqrt{2} \approx 0.707$. So in practice, if the decision region of any class lies within the volume spanned by the training samples, its certifiable region vanishes for $\sigma \geq 0.708$, regardless of the input-space dimension d.

In the case of semi-bounded decision region, the point on the axis has the highest probability to be classified as class 1, thus we study the certified radius of a point $x_0 = [0, \ldots, 0, 1]$ as a function of cone narrowness θ and smoothing factor σ. Acknowledging that the minimum distance from x_0 to θ, v cones is $\sin(\theta)$, we show in Figure 3-3(b) the scaled certified radius $r/\sin(\theta)$ when $d = 25$. One can then readily verify that overall the peak scaled certified radius decreases with θ, *e.g.* the scaled certified radius at x_0 can be as large as 0.84 when $\theta = 80°$, while it is at most 0.49 when $\theta = 10°$. Moreover, we point out that certified radii drop to zero when we keep increasing the smoothing factor σ - the "narrower" (smaller θ) the decision region is, the faster they drop to zero. We discuss the effect of input data dimension d on the certified radius in the Appendix B.1.3.

Interestingly, the certified radii increase with the growing smoothing factor σ but begin to decrease at certain point - larger certified radius can normally be obtained by larger smoothing factor σ according to equation 2.2 but the dominance is taken over by the vanishing decision region when the σ is enough-close to σ_{van}. This also explains

the eventual decrease in the average certified radius seen in Table 3.2. For small values of σ the average certified radius keeps increasing to a point ($\sigma_{thres} \in [0.50, 1.00]$) after which the effect of the vanishing decision region reduces the average certified radius.

3.4 Efficacy of data augmentation

As Section 3.3 proves that the biased prediction comes from the shrinking phenomenon of randomized smoothing, we want to hold a discussion herein investigating whether the state-of-the-art workflow for boosting randomized smoothing accuracy can solve this issue.

3.4.1 Counteracting shrinking effect of smoothing

Through the above arguments, we see that to counter-effect the shrinkage induced by randomized smoothing, one will want to obtain larger decision regions for geometrically compact classes. Assuming a well-balanced distribution of classes, compact classes have a larger number of points near the margin compared to more spread-out classes. As a result, data augmentation expands the compact classes a lot more compared to other classes, partially alleviating the shrinking issue caused by smoothing. As a result, we see that the experiments in Table 3.3 (without data augmentation) have a much bigger bias in prediction compared to the experiments in Table 3.4 column "1-standard", *e.g.* when $\sigma = 0.12$, Table 3.3 reads 0 versus 100 and Table 3.4 reads 67 versus 94.

However, it is important to note that the two effects do not exactly cancel each other out. Especially for high values of σ, the expansion caused by data augmentation can cause some of the more compact classes to dominate over all other classes, resulting in a highly biased classifier. Table 3.4 shows that the bias of the classifier consistently increases with increasing values of σ regardless of the number of augmenting points used. This signals two important observations: the need to limit the use of high values of smoothing factor σ and the need for a data geometry dependent augmentation scheme to properly counteract the shrinking effect caused by smoothing.

3.4.2 Heavy data augmentation

Besides showing the minimum and maximum class-wise accuracies of multiple CIFAR10 classifiers trained with standard data augmentation, we also give in Table 3.4 the corresponding accuracies for an enhanced version of data augmentation. Essentially, different from the standard data augmentation implementation, where only one point is used to estimate the expectation $\mathbb{E}_{z\sim\mathcal{D}_p}[l(f_{\text{train},\sigma}(x+z), h(x))]$ inside $\mathcal{R}_{\text{RS-train}}$, we evaluate the expectation using $\{10, 25, 50\}$ points, reducing the estimation bias. We denote this scheme as heavy data augmentation. Using a larger number of augmentation allows us to approximate the augmented distribution more closely and remove any unnecessary bias that is caused by using a bad approximation of the data augmentation. The results in Table 3.4 show that the bias is slightly reduced by using a larger number $\{10, 25\}$ of augmentation points but the problem still remains. Particularly, we see the relative improvement from increasing augmentation points becomes smaller with a larger smoothing factor σ. It is also worth noting that the gap in accuracies blows when we use up to 50 heavy data augmentation points, performing even worse than using the standard data augmentation. These observations signal it to be a more fundamental problem relating to the way we do data augmentation.

3.5 Conclusion

In this chapter, we provide a theoretical characterization showing that randomized smoothing during inference can lead to a drastic gap among class-wise accuracies, even when it is included in the training phase. In addition, we observe that the smoothing during inference is very sensitive to the distribution of the data and can have wildly-different effects on different classes depending on the data geometry. A similar analysis could be extended to other smoothing functions in addition to Gaussian smoothing. Crucially, our results point out the need for limiting the use of large values of σ, as well as the need for data-geometry dependent noise augmentation schemes.

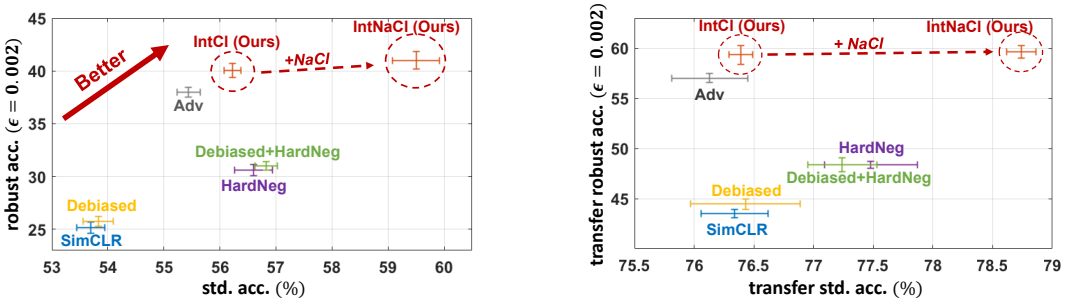
Chapter 4

Understanding and improving generic self-supervised representation learning

4.1 Introduction

Contrastive learning has drawn much attention and has become one of the most effective representation learning techniques recently. The contrastive paradigm [130, 203, 69, 23, 29, 62] constructs an objective for embeddings based on an assumed semantic similarity between positive pairs and dissimilarity between negative pairs, which stems from instance-level classification [41, 11, 203]. Specifically, the contrastive loss $\mathcal{L}_{\mathrm{CL}}$ [130, 23] is defined as $\mathbb{E}_{\substack{x\sim\mathcal{D},\\ x^+\sim\mathcal{D}_x^+,\\ x_i^-\sim\mathcal{D}_x^-}}\left[-\log\frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)}+\sum_{i=1}^{N} e^{f(x)^T f(x_i^-)}}\right]$ where, for an input data sample x, (x, x^+) denotes a positive pair and (x, x^-) denotes a negative pair. The function f is an encoder parameterized by a neural network and the number of negative pairs N is typically treated as a hyperparameter. Note that the contrastive loss can encode the inputs and keys by different encoders if one considers the use of memory bank or momentum contrast [203, 69, 26]. In this work, we will focus on the paradigm proposed in [191, 209, 23] which has demonstrated competitive results in representation learning.

When constructing loss $\mathcal{L}_{\mathrm{CL}}$, ideally, one draws x^+ from the data distribution $\mathcal{D}_x^+$ that characterizes the semantically-*similar* (*i.e.*, *positive*) samples to x; similarly, one



(a) CIFAR100 (b) CIFAR10

Figure 4-1: The performance of existing methods and our proposal (IntNaCl & IntCl) in terms of their standard accuracy (x-axis) and robust accuracy under FGSM attacks $\epsilon = 0.002$ (y-axis). The transfer performance refers to fine-tuning a linear layer for CIFAR10 with representation networks trained on CIFAR100.

wants to draw x^- from $\mathcal{D}_x^-$ that characterizes the semantically-*dissimilar* (*negative*) samples. However, the definition of semantically-*similar* and semantically-*dissimilar* is heavily contingent on downstream tasks: an image of a cat can be considered semantically similar to that of a dog if the downstream task is to distinguish between animal and non-animal classes. Without the knowledge of downstream tasks, $\mathcal{D}_x^+$ and $\mathcal{D}_x^-$ are hard to define. To provide a surrogate of measuring similarity, current mainstream contrastive learning algorithms [69, 23, 26, 62] typically build up $\mathcal{D}_x^+$ by considering data augmentation $\mathcal{D}_x^{\text{aug}}$ of a data sample x. In the meantime, $\mathcal{D}_x^-$ is approximated by the joint distribution $\mathcal{D}$ or $\mathcal{D}_{\setminus x}^{\text{aug}} := \cup_{x' \in \mathcal{D} \setminus \{x\}} \mathcal{D}_{x'}^{\text{aug}}$, and the resulting contrastive loss is known as $\mathcal{L}_{\text{SimCLR}}$ which was proposed in [23]:

(SimCLR loss $\mathcal{L}_{\text{SimCLR}}$)

$$\mathbb{E}_{\substack{x \sim \mathcal{D}, \\ x^+ \sim \mathcal{D}_x^{\text{aug}}, \\ x_i^- \sim \mathcal{D}_{\setminus x}^{\text{aug}}}} \left[-\log \frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)} + \sum_{i=1}^{N} e^{f(x)^T f(x_i^-)}} \right]. \tag{4.1}$$

Although this formulation seems to put no assumptions on the downstream task classes, we find that there are in fact implicit assumptions on the class probability prior of the downstream tasks. Specifically, we formally establish the connection between the Neighborhood Component Analysis (NCA) and the unsupervised contrastive learning

in this chapter for the first time (to our best knowledge). Inspired by this interesting relationship to NCA, we further propose two new contrastive losses (named NaCl) which outperform the existing paradigm. Furthermore, by inspecting the robust accuracy of several existing methods (e.g., Figure 4-1's y-axis, the classification accuracy when inputs are corrupted by crafted perturbations), one can see the insufficiency of existing methods in addressing robustness. Thus, we propose a new integrated contrastive framework (named IntNacl and IntCl) that accounts for *both* the standard accuracy and adversarial cases: our proposed method's performance remains in the desired upper-right region (circled) as shown in Figure 4-1.

4.1.1 Our contributions

- We establish the relationship between contrastive learning and NCA, and propose two new contrastive losses dubbed **NaCl** (Neighborhood analysis Contrastive loss). We provide theoretical analysis on NaCl and show better generalization bounds over the baselines;

- Building on top of NaCl, we propose a generic framework called Integrated contrastive learning (**IntCl** and **IntNaCl**) where we show that the spectrum of recently-proposed contrastive learning losses [29, 154, 75] can be included as special cases of our framework;

- We provide extensive experiments that demonstrate the effectiveness of IntNaCl in improving standard accuracy and robust accuracy. Specifically, IntNaCl improves upon literature [23, 29, 154, 75] by 3-6% and 4-16% in CIFAR100 standard and robust accuracy, and 2-3% and 3-17% in CIFAR10 standard and robust accuracy, respectively.

4.1.2 Related works

Contrastive learning. In the early work of [41], authors treat every individual image in a dataset as belonging to its own class and do multi-class classification tasks

under the setting. However, this regime will soon become intractable as the size of the dataset increases. To cope with this, [203] designs a memory bank for storing seen representations (keys) and utilize noise contrastive estimation [63, 121, 83, 130] for representation comparisons. [69] and [26] further improve upon [203] by storing keys inferred from a momentum encoder other than the representation encoder for x. To further reduce the computational cost, besides the practical tricks introduced in SimCLR [23] (e.g. stronger data augmentation scheme and projector heads), authors of SimCLR get rid of the memory bank and instead makes use of other samples from the same batch to form contrastive pairs.

In the rest of this chapter, we will focus on the setups of SimCLR and the related follow up work [29, 154, 75] due to computational efficiency. A temperature scaling hyperparameter t is normally used in contrastive learning to tune the radius of the hypersphere that representations lie in. For better readability, without loss of generality, we let $t = 1$ in all equations. We let $g_0(x, \{x_i^-\}^N)$ denote the negative term

$$\frac{1}{N}\sum_{i=1}^{N} e^{f(x)^T f(x_i^-)},$$

where the subscript i identifies the summation index and the superscript N identifies the summation limits. We omit the subscript i when the sample index is one dimensional (e.g. x_i^- has 1-D index, x_{ij}^- has 2-D index). Then $\mathcal{L}_{\text{SimCLR}}$ in equation 4.1 can be re-written as

(Re-written SimCLR loss $\mathcal{L}_{\text{SimCLR}}$)

$$\mathbb{E}_{\substack{x\sim\mathcal{D},\\ x^+\sim\mathcal{D}_x^{\text{aug}},\\ x_i^-\sim\mathcal{D}_{\setminus x}^{\text{aug}}}} \left[-\log \frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)} + N g_0(x, \{x_i^-\}^N)} \right]. \tag{4.2}$$

Designing *negative* pairs in contrastive learning. Several works [158, 29] have come to the awareness of the sampling bias of negative pairs in equation 4.2. Specifically, if the negative samples are sampled from $\mathcal{D}$, we will receive with $1/K$ probability a positive sample in a K-class classification task with balanced classes, hence biasing

the contrastive loss. To overcome this issue, [29] proposes a *de-biased* contrastive loss to mitigate the sampling bias by explicitly including the class probability prior on the downstream tasks (e.g., with probability 0.1, x_i^- contains a positive example in CIFAR10), and tune the prior τ^+ as a hyperparameter. We denote the loss from [29] as $\mathcal{L}_{\text{Debiased}}$ and the full equation is shown below:

$$\begin{aligned}&\text{(Debiased loss } \mathcal{L}_{\text{Debiased}}\text{)}\\ &\mathbb{E}_{\substack{x\sim\mathcal{D},\\ x^+\sim\mathcal{D}_x^{\text{aug}},\\ v_j\sim\mathcal{D}_x^{\text{aug}},\\ u_i\sim\mathcal{D}_{\setminus x}^{\text{aug}}}}\left[-\log\frac{e^{f(x)^Tf(x^+)}}{e^{f(x)^Tf(x^+)}+Ng_1(x,\{u_i\}^n,\{v_j\}^m)}\right],\end{aligned}\tag{4.3}$$

where the estimator $g_1(x,\{u_i\}^n,\{v_j\}^m)$ is defined by

$$\max\{\frac{\sum_{i=1}^n e^{f(x)^Tf(u_i)}}{(1-\tau^+)n}-\frac{\tau^+\sum_{j=1}^m e^{f(x)^Tf(v_j)}}{(1-\tau^+)m},e^{-1/t}\}$$

and n and m represents the numbers of sampled points in $\mathcal{D}_{\setminus x}^{\text{aug}}$ and $\mathcal{D}_x^{\text{aug}}$ for the re-weighted negative term, τ^+ is the class probability prior, and t is the temperature hyperparameter. Recently, [154] proposes to weigh sample pairs through the cosine distance in the estimator $g_1(x,\{u_i\}^n,\{v_j\}^m)$ based on $\mathcal{L}_{\text{Debiased}}$, and we denote their approach as $\mathcal{L}_{\text{Debiased+HardNeg}}$,

$$\begin{aligned}&\text{(Debiased+HardNeg loss } \mathcal{L}_{\text{Debiased+HardNeg}}\text{)}\\ &\mathbb{E}_{\substack{x\sim\mathcal{D},\\ x^+\sim\mathcal{D}_x^{\text{aug}},\\ v_j\sim\mathcal{D}_x^{\text{aug}},\\ u_i\sim\mathcal{D}_{\setminus x}^{\text{aug}}}}\left[-\log\frac{e^{f(x)^Tf(x^+)}}{e^{f(x)^Tf(x^+)}+Ng_2(x,\{u_i\}^n,\{v_j\}^m)}\right],\end{aligned}\tag{4.4}$$

where the estimator $g_2(x,\{u_i\}^n,\{v_j\}^m)$ is defined by

$$\max\{\frac{\sum_{i=1}^n \kappa_i^{\beta+1}}{(1-\tau^+)\sum_{i=1}^n \kappa_i^{\beta}}-\frac{\tau^+\sum_{j=1}^m e^{f(x)^Tf(v_j)}}{(1-\tau^+)m},e^{-1/t}\}$$

and $\kappa_i = e^{f(x)^Tf(u_i)}$. A typical choice of n and m are $n=N$ and $m=1$, and the hyperparameter τ^+ in g_2 is exactly the same as that in g_1 whereas the hyperpa-

rameter β controls the weighting mechanism. Specifically, when $\tau^+ = 0$, we denote $\mathcal{L}_{\text{Debiased+HardNeg}}$ as $\mathcal{L}_{\text{HardNeg}}$; when $\beta = 0$, equation 4.4 degenerates to equation 4.3 which is $\mathcal{L}_{\text{Debiased}}$.

Designing *positive* pairs in contrastive learning. Instead of modifying the negative pairs, another direction is to design the positive pairs [75, 89]. Specifically, authors of [75] define the concept of *adversarial examples* in the regime of representation learning as the positive sample x^{adv} that maximizes $\mathcal{L}_{\text{SimCLR}}$ in equation 4.2 within a pre-specified perturbation magnitude ϵ. The resulting loss function is denoted as $\mathcal{L}_{\text{Adv}}$:

$$\begin{aligned}&\text{(Adversarial loss } \mathcal{L}_{\text{Adv}}\text{)}\\ &\mathbb{E}_{\substack{x\sim\mathcal{D},\\ x^+\sim\mathcal{D}_x^{\text{aug}},\\ x_{i_1}^-\sim\mathcal{D}_{\setminus x}^{\text{aug}},\\ x_{i_2}^-\sim\mathcal{D}_{\setminus x}^{\text{adv}}}}\Bigg[-\log\frac{e^{f(x)^Tf(x^+)}}{e^{f(x)^Tf(x^+)}+Ng_0(x,\{x_{i_1}^-\}^N)}\\ &\qquad -\alpha\log\frac{e^{f(x)^Tf(x^{\text{adv}})}}{e^{f(x)^Tf(x^{\text{adv}})}+Ng_0(x,\{x_{i_2}^-\}^N)}\Bigg],\end{aligned} \tag{4.5}$$

where the $\mathcal{D}_{\setminus x}^{\text{adv}}$ is defined by $\cup_{x'\in\mathcal{D}\setminus\{x\}}x'\cup x'^{,\text{adv}}$. Notably, one can adjust the importance of the adversarial term by tuning α in equation 4.5.

4.2 Two new NCA-inspired contrastive losses and an integrated framework

In this section, we first derive a connection between Neighborhood Component Analysis (NCA) [59] and the unsupervised contrastive learning loss in Section 4.2.1. Inspired by our result in Section 4.2.1, we propose two new NCA-inspired contrastive losses in Section 4.2.2, which we refer to as **N**eighborhood **a**nalysis **C**ontrastive **l**oss (**NaCl**). To address a lack of robustness in existing contrastive losses, in Section 4.2.3, we propose a useful framework **IntNaCl** that integrates NaCl and a robustness-promoting loss. A summary of definitions is given as Table 4.1.

Table 4.1: A summary of definitions.

$\mathcal{L}_{\text{NCA}}(G^1, M)$	$\mathbb{E}_{x\sim\mathcal{D},x_j^+\sim\mathcal{D}_x^{\text{aug}},x_i^-\sim\mathcal{D}_{\setminus x}^{\text{aug}}}[-\log \frac{\sum_{j=1}^{M} e^{f(x)^T f(x_j^+)}}{\sum_{j=1}^{M} e^{f(x)^T f(x_j^+)}+NG^1(x,\{x_i^-\}^N)}]$
$\mathcal{L}_{\text{MIXNCA}}(G^1, M, \lambda)$	$\mathbb{E}_{x\sim\mathcal{D},x^+\sim\mathcal{D}_x^{\text{aug}},x_{i_1}^-,x_{i_2 j}^-,x_j^-\sim\mathcal{D}_{\setminus x}^{\text{aug}}}[-\log \frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)}+NG^1(x,\{x_{i_1}^-\}^N)}$ $-\frac{\lambda}{M-1}\sum_{j=1}^{M-1}\log\frac{e^{f(x)^T f(\lambda x^+ +(1-\lambda)x_j^-)}}{e^{f(x)^T f(\lambda x^+ +(1-\lambda)x_j^-)}+NG^1(x,\{x_{i_2 j}^-\}_{i_2}^N)}$ $-\frac{1-\lambda}{M-1}\sum_{j=1}^{M-1}\log\left(1-\frac{e^{f(x)^T f(\lambda x^+ +(1-\lambda)x_j^-)}}{e^{f(x)^T f(\lambda x^+ +(1-\lambda)x_j^-)}+NG^1(x,\{x_{i_2 j}^-\}_{i_2}^N)}\right)]$
$g_0(x,\{x_i^-\}_i^N)$	$\frac{1}{N}\sum_{i=1}^N e^{f(x)^T f(x_i^-)}$
$g_1(x,\{u_i\}^n,\{v_j\}^m)$	$\max\{\frac{1}{1-\tau^+}(\frac{1}{n}\sum_{i=1}^n e^{f(x)^T f(u_i)}-\tau^+\frac{1}{m}\sum_{j=1}^m e^{f(x)^T f(v_j)}), e^{-1/t}\}$
$g_2(x,\{u_i\}^n,\{v_j\}^m)$	$\max\{\frac{1}{1-\tau^+}(\frac{\sum_{i=1}^n e^{(\beta+1)f(x)^T f(u_i)}}{\sum_{i=1}^n e^{\beta f(x)^T f(u_i)}}-\tau^+\frac{1}{m}\sum_{j=1}^m e^{f(x)^T f(v_j)}), e^{-1/t}\}$
$\hat{w}(x)$	$-\log\frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)}+NG(x,\cdot)}$

4.2.1 Bridging from supervised NCA to unsupervised contrastive learning: a new finding

NCA is a supervised learning algorithm concerned with learning a quadratic distance metric with the matrix A such that the performance of nearest neighbor classification is maximized. Notice that the set of neighbors for a data point is a function of transformation A. However, it can remain unchanged as A changes within a certain range. Therefore the leave-one-out classification performance can be a piecewise-constant function of A and hence non-differentiable. To overcome this, the optimization problem is generally given using the concept of stochastic nearest neighbors. In the stochastic nearest neighbor setting, nearest neighbor selection is regarded as a random event, where the probability that point x_j is selected as the nearest neighbor for x_i is given as $p(x_j \mid x_i)$ with

$$p_{ij} := p(x_j \mid x_i) = \frac{e^{-\|Ax_i - Ax_j\|^2}}{\sum_{k\neq i} e^{-\|Ax_i - Ax_k\|^2}},\ j \neq i. \tag{4.6}$$

Let c_i denote the label of x_i, in the leave-one-out classification loss, the probability a point is classified correctly is given as $p_i = \sum_{j|c_j=c_i} p_{ij}$, where $\{j \mid c_j = c_i\}$ defines an index set in which all points x_j belong to the same class as point x_i. We use M to

denote the cardinality of this set. By the definition of c_i, the probability x_i's label is c_i is given as q_i, which is exactly 1[1]. Thus the optimization problem can be written as $\min_A \sum_{i=1}^n \ell(q_i, \sum_{j|c_j=c_i} p_{ij})$. This learning objective then naturally maximizes the expected accuracy of a 1-nearest neighbor classifier. Two popular choices for $\ell(\cdot)$ are the total variation distance and the KL divergence. In the seminal paper of [59], the authors showed both losses give similar results, thus we will focus on the KL divergence loss in this work. For $\ell(\cdot) = \mathrm{KL}$, the relative entropy from p to q is $D_{\mathrm{KL}}(q\|p) = \sum_i -q_i \log \frac{p_i}{q_i} = \sum_i -\log p_i$ when $q_i = 1$. By plugging in the definition of $p_i = \sum_{j|c_j=c_i} p_{ij}$ and equation 4.6, the NCA problem becomes

$$\min_A \sum_{i=1}^n -\log \left(\sum_{j|c_j=c_i} \frac{e^{-\|Ax_i - Ax_j\|^2}}{\sum_{k\neq i} e^{-\|Ax_i - Ax_k\|^2}} \right). \tag{4.7}$$

With the above formulation, we now show how to establish the connection of NCA to the contrastive learning loss. First, by assuming (a) positive pairs belong to the same class and (b) the transformation Ax is instead parametrized by a general function $\frac{f(x)}{\sqrt{2}} := \frac{h(x)}{\sqrt{2}\|h(x)\|}$, where h is a neural network, equation 4.7 becomes equation 4.8:

$$\min_f \sum_{i=1}^n -\log \left(\sum_{j=1}^M \frac{e^{-\frac{1}{2}\left\|f(x_i) - f(x_{ij}^+)\right\|^2}}{\sum_{k\neq i} e^{-\frac{1}{2}\|f(x_i) - f(x_k)\|^2}} \right). \tag{4.8}$$

[1]For every data point, p and q are defined differently with their supports being the class index. For every sample x, q_i is the ground truth probability of class labels and p_i is the prediction probability.

Then we can prove

$$\arg\min_f \sum_{i=1}^{n} -\log\left(\sum_{j=1}^{M}\frac{e^{-\frac{1}{2}\left\|f(x_i)-f(x_{ij}^+)\right\|^2}}{\sum_{k\neq i}e^{-\frac{1}{2}\|f(x_i)-f(x_k)\|^2}}\right)$$

$$=\arg\min_f \sum_{i=1}^{n} -\log\left(\sum_{j=1}^{M}\frac{e^{f(x_i)^Tf(x_{ij}^+)-\frac{1}{2}\|f(x_i)\|^2-\frac{1}{2}\left\|f(x_{ij}^+)\right\|^2}}{\sum_{k\neq i}e^{f(x_i)^Tf(x_k)-\frac{1}{2}\|f(x_i)\|^2-\frac{1}{2}\|f(x_k)\|^2}}\right) \tag{4.9}$$

$$=\arg\min_f \sum_{i=1}^{n} -\log\left(\sum_{j=1}^{M}\frac{e^{f(x_i)^Tf(x_{ij}^+)-1}}{\sum_{k\neq i}e^{f(x_i)^Tf(x_k)-1}}\right) \tag{4.10}$$

$$=\arg\min_f \sum_{i=1}^{n} -\log\left(\frac{\sum_{j=1}^{M}e^{f(x_i)^Tf(x_{ij}^+)}}{\sum_{k\neq i}e^{f(x_i)^Tf(x_k)}}\right)$$

$$=\arg\min_f \sum_{i=1}^{n} -\log\left(\frac{\sum_{j=1}^{M}e^{f(x_i)^Tf(x_{ij}^+)}}{\sum_{k\neq i,x_k\in\{x_{ij}^+\}}e^{f(x_i)^Tf(x_k)}+\sum_{k\neq i,x_k\notin\{x_{ij}^+\}}e^{f(x_i)^Tf(x_k)}}\right) \tag{4.11}$$

$$=\arg\min_f \mathbb{E}_{x\sim\mathcal{D}}\left[-\log\left(\frac{\sum_{j=1}^{M}e^{f(x)^Tf(x_j^+)}}{\sum_{j=1}^{M}e^{f(x)^Tf(x_j^+)}+\sum_{i=1}^{N}e^{f(x)^Tf(x_i^-)}}\right)\right] \tag{4.12}$$

$$=\arg\min_f \mathbb{E}_{x\sim\mathcal{D}}\left[-\log\left(\frac{\sum_{j=1}^{M}e^{f(x)^Tf(x_j^+)}}{\sum_{j=1}^{M}e^{f(x)^Tf(x_j^+)}+Ng_0(x,\{x_i^-\}^N)}\right)\right], \tag{4.13}$$

where we go from equation 4.9 to equation 4.10 based on the fact that $\|f(x)\|=1$, and from equation 4.11 to equation 4.12 assuming that set $\{x_k : k\neq i\}=\{x_j^+ : 1\leq j\leq M\}\cup\{x_i^- : 1\leq i\leq N\}$.

Notice that equation 4.13 is a more general contrastive loss where the contrastive loss $\mathcal{L}_{\text{SimCLR}}$ in [23] is a special case with $M=1, x^+\sim\mathcal{D}_x^{\text{aug}}$:

$$\min_f \mathbb{E}_{\substack{x\sim\mathcal{D},\\ x^+\sim\mathcal{D}_x^{\text{aug}},\\ x_i^-\sim\mathcal{D}_{\setminus x}^{\text{aug}}}}\left[-\log\left(\frac{e^{f(x)^Tf(x^+)}}{e^{f(x)^Tf(x^+)}+Ng_0(x,\{x_i^-\}^N)}\right)\right].$$

With the above analysis, two new contrastive losses are proposed based on equation 4.13

in the next Section 4.2.2. As a side note, as the computation of the loss grows quadratically with the size of the dataset, the current method [23] uses mini batches to construct positive/negative pairs in a data batch of size N to estimate the loss.

4.2.2 Neighborhood analysis contrastive loss (NaCl)

Based on the connection we have built in Section 4.2.1, we discover that the reduction from the NCA formulation to $\mathcal{L}_{\text{SimCLR}}$ assumes

1. the expected relative density of positives in the underlying data distribution is $1/N$;
2. the probability q_i induced by encoder network f is 1.

By relaxing the assumptions individually, in this section, we propose two new contrastive losses. Note that the two neighborhood analysis contrastive losses are designed from orthogonal perspectives, hence they are complementary to each other. We use $\mathcal{L}_{\text{NaCl}}$ to denote these two variant losses: $\mathcal{L}_{\text{NCA}}$ and $\mathcal{L}_{\text{MIXNCA}}$.

(I) Relaxing assumption 1: $\mathcal{L}_{\text{NCA}}$. When relating unsupervised SimCLR to supervised NCA, we view two samples in a positive pair as same-class samples. Since in SimCLR, the number of positive pairs $M = 1$, which means that $\{j \mid c_j = c_i\}$ only contains one element. This implies the relative density of positives in the underlying data distribution is $M/N = 1/N$, where N is the data batch size. However, as the expected relative density is task-dependent, it's more reasonable to treat the M/N ratio as a hyperparameter similar to the class probabilities τ^+ introduced by [29]. Therefore, we propose the more general contrastive loss $\mathcal{L}_{\text{NCA}}$ which could include more than one element or equivalently $M \neq 1$:

$$\begin{aligned}&(\text{NCA loss } \mathcal{L}_{\text{NCA}}(G = g_0, M))\\ &\mathbb{E}_{\substack{x\sim\mathcal{D},\\ x_j^+\sim\mathcal{D}_x^{\text{aug}},\\ x_i^-\sim\mathcal{D}_{\setminus x}^{\text{aug}}}}\left[-\log\frac{\sum_{j=1}^{M} e^{f(x)^T f(x_j^+)}}{\sum_{j=1}^{M} e^{f(x)^T f(x_j^+)} + N g_0(x, \{x_i^-\}^N)}\right].\end{aligned}$$

We further provide the generalization results as follows: if we let $\mathcal{F}$ be a function class, K be the number of classes, $\mathcal{L}_{\text{Sup}}$ be the cross entropy loss of any downstream K-class classification task, $\widehat{\mathcal{L}}_{\text{NCA}}(g_0, M)$ be the empirical NCA loss, T be the size of the dataset, and $\mathcal{R}_{\mathcal{S}}(\mathcal{F})$ be the empirical Rademacher complexity of $\mathcal{F}$ w.r.t. data sample $\mathcal{S}$, then

Theorem 8. *With probability at least $1-\delta$, for any $f \in \mathcal{F}$ and $N \geq K-1$,*

$$\mathcal{L}_{\text{Sup}}(\hat{f}) \leq \mathcal{L}_{\text{NCA}}(g_0, M)(f) + \mathcal{O}\left(\sqrt{\frac{1}{N}} + \frac{\lambda \mathcal{R}_{\mathcal{S}}(\mathcal{F})}{T} + B\sqrt{\frac{\log \frac{1}{\delta}}{T}}\right),$$

where $\hat{f} = \arg\min_{f \in \mathcal{F}} \widehat{\mathcal{L}}_{\text{NCA}}(g_0, M)(f)$, $\lambda = \frac{1}{M}$, and $B = \log N$.

We can see from the term λ that $\mathcal{L}_{\text{NCA}}(G = g_0, M)$ improves upon $\mathcal{L}_{\text{SimCLR}}$ by using a $M \neq 1$.

(II) Relaxing assumption 2: $\mathcal{L}_{\text{MIXNCA}}$. To reduce the reliance on the downstream prior, a practical relaxation can be made by allowing neighborhood samples to agree with each other with probability. This translates into relaxing the specification of $q_i = 1$ and consider a synthetic data point $x' = \lambda x_i + (1-\lambda)y, y \sim \mathcal{D}$ that belongs to a synthetic class $c_{\lambda,i}$. Assume the probability x_i's label is $c_{\lambda,i}$ is $q_{\lambda,i} = \lambda + (1-\lambda)[c_y = c_i]$, then $q_{\lambda,i}$ should match the probability $p_{\lambda,i} = \sum_{j|c_j = c_{\lambda,i}} p_{ij}$, where $\{j \mid c_j = c_{\lambda,i}\}$ is a singleton containing only the index of x', which yields

$$\text{(MIXNCA loss } \mathcal{L}_{\text{MIXNCA}}(G = g_0, M, \lambda))$$
$$\mathbb{E}_{\substack{x \sim \mathcal{D}, \\ x^+ \sim \mathcal{D}_x^{\text{aug}}, \\ x_{i_1}^-, x_{i_2 j}^-, x_j^- \sim \mathcal{D}_{\setminus x}^{\text{aug}}}} \left[-\log \frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)} + N g_0(x, \{x_{i_1}^-\}^N)} - \frac{\lambda}{M-1} \sum_{j=1}^{M-1} \log \Omega_j - \frac{1-\lambda}{M-1} \sum_{j=1}^{M-1} \log(1-\Omega_j) \right],$$

Table 4.2: The relationship between IntNaCl framework and the literature: existing works are special cases of $\mathcal{L}_{\text{IntNaCl}}$

	$\mathcal{L}_{\text{IntNaCl}}$	$\mathcal{L}_{\text{NaCl}}(G^1, M, \lambda)$				α	$\mathcal{L}_{\text{Robust}}(G^2, w))$	
		$\mathcal{L}_{\text{NaCl}}$	G^1	M	λ		G^2	w
Existing Work	$\mathcal{L}_{\text{SimCLR}}$ [23]	$\mathcal{L}_{\text{NCA}}/\mathcal{L}_{\text{MIXNCA}}$	g_0	1	-	0	-	-
	$\mathcal{L}_{\text{Debiased}}$ [29]	$\mathcal{L}_{\text{NCA}}/\mathcal{L}_{\text{MIXNCA}}$	g_1	1	-	0	-	-
	$\mathcal{L}_{\text{Debiased+HardNeg}}$ [154]	$\mathcal{L}_{\text{NCA}}/\mathcal{L}_{\text{MIXNCA}}$	g_2	1	-	0	-	-
	$\mathcal{L}_{\text{Adv}}$ [75]	$\mathcal{L}_{\text{NCA}}/\mathcal{L}_{\text{MIXNCA}}$	g_0	1	-	1	g_0	1
Our Method	$\mathcal{L}_{\text{IntCl}}$ in Fig. 4-1	$\mathcal{L}_{\text{NCA}}/\mathcal{L}_{\text{MIXNCA}}$	g_2	1	-	1	g_2	$\hat{w}(x)$
	$\mathcal{L}_{\text{IntNaCl}}$ in Fig. 4-1	$\mathcal{L}_{\text{MIXNCA}}$	g_2	5	0.5	1	g_2	$\hat{w}(x)$
	$\mathcal{L}_{\text{IntNaCl}}$ in Tab. 4.3	$\mathcal{L}_{\text{NCA}}/\mathcal{L}_{\text{MIXNCA}}$	g_0/g_2	1-5	0.5/0.9	0	-	-
	$\mathcal{L}_{\text{IntNaCl}}$ in Tab. 4.4	$\mathcal{L}_{\text{NCA}}/\mathcal{L}_{\text{MIXNCA}}$	g_2	1-5	0.5/0.7/0.9	1	g_2	$\hat{w}(x)$
	$\mathcal{L}_{\text{IntNaCl}}$ in Fig. 4-2	$\mathcal{L}_{\text{MIXNCA}}$	g_0/g_2	1-5	0.5-0.9	0	-	-
	$\mathcal{L}_{\text{IntNaCl}}$ in Tab. 4.5	$\mathcal{L}_{\text{NCA}}/\mathcal{L}_{\text{MIXNCA}}$	g_0/g_2	1/2/5	0.5/0.9	0/1	-/g_2	-/$\hat{w}(x)$/1

where $\Omega_j = \frac{e^{f(x)^T f(\lambda x^+ + (1-\lambda)x_j^-)}}{e^{f(x)^T f(\lambda x^+ + (1-\lambda)x_j^-)} + N g_0(x, \{x_{i_2 j}^-\}_{i_2}^N)}$. Interestingly, the construction of x' herein assembles the mixup [217] philosophy in supervised learning. Recent work [101, 184] have also considered augment the dataset by including synthetic data point and build domain-agnostic contrastive learning strategies, however, their loss is different from this work because they apply mixup on the data points x while we use mixup to produce diverse positive pairs.

4.2.3 Integrated contrastive learning framework

Building on top of NaCl, we can propose a useful framework **IntNaCl** that not only generalizes existing methods but also achieves good accuracy and robustness simultaneously. Before we introduce IntNaCl, we give an intermediate integrated loss as **IntCl**, which consists of two components – a standard loss and a robustness-promoting loss.

Motivated by $\mathcal{L}_{\text{Adv}}$ [75], we consider a robust-promoting loss defined by

$$\mathcal{L}_{\text{Robust}}(G, w) := \mathbb{E}\left[-\log \frac{e^{f(x)^T f(x^{\text{adv}})}}{e^{f(x)^T f(x^{\text{adv}})} + N G(x, \cdot)} w(x)\right],$$

where G can be chose from $\{g_0, g_1, g_2\}$, and $w(x)$ facilitates goal-specific weighting schemes. Note that $w(x)$ can be a general function and $\mathcal{L}_{\text{Adv}}$ [75] is a special case when $w(x) = 1$.

Adversarial weighting. Weighting sample loss based on their margins has been proven to be effective in the adversarial training under supervised settings [213]. Specifically, it is argued that training points that are closer to the decision boundaries should be given more weight in the supervised loss. While the margin of a sample is underdefined in unsupervised settings, we can give our weighting function as the value of the contrastive loss $\hat{w}(x) := -\log \frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)} + NG(x,\cdot)}$. Using this, we see that samples that are originally hard to be distinguished from other samples (*i.e.* small probability) are now assigned with bigger weights. Below, we propose a new integrated framework to involve the robustness term $\mathcal{L}_{\text{Robust}}(G, w)$ which can greatly help on promoting robustness in contrastive learning. In particular, we show that many existing contrastive learning losses are special cases of our proposed framework.

IntCl. For IntCl, the standard loss can be existing contrastive learning losses [23, 29, 154], which correspond to a form of

$$\begin{aligned}&\text{(IntCL loss } \mathcal{L}_{\text{IntCL}}\text{)}\\ &\mathbb{E}\left[-\log \frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)} + NG^1(x,\cdot)}\right] + \alpha \mathcal{L}_{\text{Robust}}(G^2, w),\end{aligned}$$

with G^1 and G^2 being g_0, g_1, and g_2. Unless otherwise specified, we use the adversarial weighting scheme introduced above throughout our experiments. Notice that $\mathcal{L}_{\text{IntCL}}$ reduces to $\mathcal{L}_{\text{Adv}}$ when $G^1 = G^2 = g_0$ and $w(x) \equiv 1$.

IntNaCl. To design a generic loss that accounts for robust accuracy while keeping clean accuracy, we utilize $\mathcal{L}_{\text{NaCl}}$ developed in Section 4.2.2 to strength the standard loss in $\mathcal{L}_{\text{IntCL}}$. We call this ultimate framework **Int**egrated **N**eighborhood **a**nalysis **C**ontrastive **l**oss (**IntNaCl**), which is given by

$$\mathcal{L}_{\text{IntNaCl}} := \mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) + \alpha \mathcal{L}_{\text{Robust}}(G^2, w), \tag{4.14}$$

where $\mathcal{L}_{\text{NaCl}}(G^1, M, \lambda)$ can be chose from $\{\mathcal{L}_{\text{NCA}}(G^1, M),\ \mathcal{L}_{\text{MIXNCA}}(G^1, M, \lambda)\}$. We remark that as $\mathcal{L}_{\text{NCA}}$ and $\mathcal{L}_{\text{MIXNCA}}$ all reduce to one same form when $M = 1$, the $\mathcal{L}_{\text{IntNaCl}}$ under $M = 1$ is exactly $\mathcal{L}_{\text{IntCl}}$. This general framework includes many of the existing works as special cases and we summarize these relationships in Table 4.2.

4.3 Experimental results

Implementation details. All the proposed methods are implemented based on open source repositories provided in the literature [23, 75, 154]. Five benchmarking contrastive losses are considered as baselines that include: $\mathcal{L}_{\text{SimCLR}}$ [23], $\mathcal{L}_{\text{Debiased}}$ [29], $\mathcal{L}_{\text{Debiased+HardNeg}}$ [154], $\mathcal{L}_{\text{Adv}}$ [75] (*i.e.* equation 4.2, equation 4.3, equation 4.4, equation 4.5). We train representations on resnet18 and include MLP projection heads [23]. A batch size of 256 is used for all CIFAR [93] experiments and a batch size of 128 is used for all tinyImagenet experiments. Unless otherwise specified, the representation network is trained for 100 epochs. We run five independent trials for each of the experiments and report the mean and standard deviation in the entries. We implement the proposed framework using PyTorch to enable the use of an NVIDIA GeForce RTX 2080 Super GPU and four NVIDIA Tesla V100 GPUs.

Evaluation protocol. We follow the standard evaluation protocal to report three major properties of representation learning methods: standard discriminative power, transferability, and adversarial robustness. To evaluate the standard discriminative power, we train representation networks on CIFAR100/tinyImagenet, freeze the network, and fine-tune a fully-connected layer that maps representations to outputs on CIFAR100/tinyImagenet, which is consistent with the standard linear evaluation protocol in the literature [23, 29, 62, 75, 88, 177, 154, 158, 89, 66]. To evaluate the transferability, we use the representation networks trained on CIFAR100, and only fine-tune a fully-connected layer that maps representations to outputs on CIFAR10. All the adversarial robustness evaluations are based on the implementation provided by [200].

Table 4.3: Performance comparisons of $\mathcal{L}_{\text{NaCl}}$ ($M \neq 1$) and i) *Left*: SimCLR [29] ($M = 1, G^1 = g_0$) and ii) *Right*: Debised+HardNeg [154] ($M = 1, G^1 = g_2$) when $\alpha = 0$. The best accuracy (%) within each loss type is in boldface (larger is better).

M	$\alpha=0,\ \mathcal{L}_{\text{NaCl}}(G^1,M,\lambda)=\mathcal{L}_{\text{NCA}}(g_0,M)$				$\alpha=0,\ \mathcal{L}_{\text{NaCl}}(G^1,M,\lambda)=\mathcal{L}_{\text{NCA}}(g_2,M)$			
	CIFAR100	CIFAR100 Adv.	CIFAR10	CIFAR10 Adv.	CIFAR100	CIFAR100 Adv.	CIFAR10	CIFAR10 Adv.
1	53.69±0.25	25.17±0.55	76.34±0.28	43.50±0.41	56.83±0.20	31.03±0.41	77.24±0.29	48.38±0.70
2	55.72±0.15	27.04±0.45	77.40±0.14	44.58±0.41	57.87±0.15	32.50±0.48	77.43±0.11	48.14±0.31
3	56.67±0.12	**28.41±0.24**	77.53±0.24	**45.21±0.89**	58.42±0.23	**33.19±0.60**	77.41±0.17	48.09±0.93
4	57.09±0.26	28.20±0.81	77.75±0.22	45.13±0.44	**58.86±0.18**	32.65±1.07	77.46±0.29	**48.43±0.94**
5	**57.32±0.17**	28.33±0.59	**77.93±0.40**	44.46±0.53	58.81±0.21	32.86±0.47	**77.58±0.23**	48.30±0.39
	$\alpha=0,\ \mathcal{L}_{\text{NaCl}}(G^1,M,\lambda)=\mathcal{L}_{\text{MIXNCA}}(g_0,M,0.9)$				$\alpha=0,\ \mathcal{L}_{\text{NaCl}}(G^1,M,\lambda)=\mathcal{L}_{\text{MIXNCA}}(g_2,M,0.5)$			
1	53.69±0.25	25.17±0.55	76.34±0.28	43.50±0.41	56.83±0.20	31.03±0.41	77.24±0.29	48.38±0.70
2	56.20±0.33	30.95±0.36	76.96±0.15	**48.85±0.75**	59.41±0.19	**32.22±0.35**	79.36±0.65	48.86±0.34
3	56.41±0.13	**30.98±0.90**	77.10±0.21	48.76±0.63	59.81±0.25	32.04±0.67	79.41±0.17	48.91±0.81
4	56.00±0.42	29.90±0.63	**77.11±0.40**	48.16±0.40	59.75±0.33	32.03±0.34	79.42±0.18	**49.05±0.71**
5	**56.63±0.31**	30.58±0.52	77.04±0.19	47.96±0.46	**59.85±0.30**	32.06±0.72	**79.45±0.20**	48.32±0.70

Table 4.4: Performance comparisons of $\mathcal{L}_{\text{IntNaCl}}$ ($M \neq 1$) and $\mathcal{L}_{\text{IntCL}}$ ($M = 1$) when $\alpha = 1, G^1 = G^2 = g_2, w = \hat{w}(x)$. The best accuracy (%) within each loss type is in boldface (larger is better).

M	$\alpha\neq0,\ \mathcal{L}_{\text{NaCl}}(G^1,M,\lambda)=\mathcal{L}_{\text{NCA}}(g_2,M)$				$\alpha\neq0,\ \mathcal{L}_{\text{NaCl}}(G^1,M,\lambda)=\mathcal{L}_{\text{MIXNCA}}(g_2,M,0.5)$			
	CIFAR100	CIFAR100 Adv.	CIFAR10	CIFAR10 Adv.	CIFAR100	CIFAR100 Adv.	CIFAR10	CIFAR10 Adv.
1	56.22±0.15	40.05±0.67	76.39±0.10	**59.33±0.94**	56.22±0.15	40.05±0.67	76.39±0.10	**59.33±0.94**
2	56.71±0.11	39.80±0.57	76.55±0.27	58.44±0.31	58.97±0.19	40.25±0.52	78.61±0.20	58.41±0.59
3	57.13±0.26	40.53±0.29	**76.67±0.22**	58.47±0.31	59.26±0.18	40.96±0.58	**78.83±0.22**	59.20±1.25
4	57.06±0.19	40.85±0.31	76.34±0.22	58.91±0.62	59.32±0.21	40.82±0.54	78.83±0.27	59.03±0.52
5	**57.46±0.04**	**41.00±0.86**	76.60±0.37	57.98±0.47	**59.43±0.23**	**41.01±0.34**	78.80±0.21	**59.51±0.93**
	$\alpha\neq0,\ \mathcal{L}_{\text{NaCl}}(G^1,M,\lambda)=\mathcal{L}_{\text{MIXNCA}}(g_2,M,0.7)$				$\alpha\neq0,\ \mathcal{L}_{\text{NaCl}}(G^1,M,\lambda)=\mathcal{L}_{\text{MIXNCA}}(g_2,M,0.9)$			
1	56.22±0.15	40.05±0.67	76.39±0.10	59.33±0.94	56.22±0.15	40.05±0.67	76.39±0.10	59.33±0.94
2	58.00±0.18	40.35±0.34	77.73±0.24	59.40±1.27	56.54±0.33	40.85±0.13	76.81±0.22	60.40±0.46
3	58.23±0.18	40.94±0.75	77.91±0.25	**59.57±0.81**	56.69±0.11	41.23±0.66	**76.98±0.22**	60.13±0.56
4	58.20±0.25	40.95±0.45	77.89±0.20	59.49±0.49	56.43±0.26	**41.56±0.56**	76.97±0.20	**61.21±0.49**
5	**58.37±0.14**	**41.15±0.48**	**78.27±0.26**	59.17±0.94	**56.86±0.11**	41.09±0.31	76.91±0.21	60.09±0.39

Experiment outline. Since the performance of the integrated method $\mathcal{L}_{\text{IntNaCl}}$ is attributed to multiple components in the formulation (equation 4.14), we do ablation studies in the following sections to study their effectiveness individually. In Section 4.3.1, we evaluate the effect of $\mathcal{L}_{\text{NaCl}}$; in Section 4.3.2, we evaluate the effect of $\mathcal{L}_{\text{Robust}}$; in Section 4.3.3, we evaluate the effect of M, λ, and w.

4.3.1 The effect of $\mathcal{L}_{\text{NaCl}}$

By evaluating the effect of $\mathcal{L}_{\text{NaCl}}$, we want to evaluate the performance difference of our framework $\mathcal{L}_{\text{IntNaCl}}$ when $M \geq 1$ and $M = 1$. In order to see that, we consider 2 cases: (1) set $\alpha = 0$ in equation 4.14 and compare $\mathcal{L}_{\text{NaCl}}(G^1, M \neq 1, \lambda)$ with existing work $\mathcal{L}_{\text{NaCl}}(G^1, M = 1, \lambda)$, or (2) set $\alpha = 1$ and compare $\mathcal{L}_{\text{IntNaCl}}$ and $\mathcal{L}_{\text{IntCl}}$.

Case (1) $\alpha = 0$**.** In Table 4.3, after setting $\alpha = 0$, we experiment with $G^1 = g_0, g_2$.

Table 4.5: Performance comparisons of $\mathcal{L}_{\text{NaCl}}$ and $\mathcal{L}_{\text{IntNaCl}}$ with baselines on TinyImagenet. The best accuracy (%) within each loss type is in boldface (larger is better).

$\alpha = 0$	$\mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{NCA}}(g_0, M)$		$\mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{NCA}}(g_2, M)$	
M	TinyImagenet	TinyImagenet Adv.	TinyImagenet	TinyImagenet Adv.
1	39.66±0.15	24.80±0.07	41.26±0.14	27.34±0.77
2	**40.71±0.26**	**26.29±0.51**	**41.99±0.23**	**28.14±0.13**
	$\mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_0, M, 0.9)$		$\mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_2, M, 0.5)$	
1	39.66±0.15	24.80±0.07	41.26±0.14	27.34±0.77
2	**40.23±0.37**	**26.47±0.24**	**43.91±0.20**	**28.29±0.33**
$\alpha = 1$	$\mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_2, M, 0.5)$ $\mathcal{L}_{\text{Robust}}(G^2, w)) = \mathcal{L}_{\text{Robust}}(g_2, \hat{w}(x)))$		$\mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_2, M, 0.5)$ $\mathcal{L}_{\text{Robust}}(G^2, w)) = \mathcal{L}_{\text{Robust}}(g_2, 1))$	
1	42.56±0.13	31.18±0.51	42.24±0.14	31.55±0.38
2	44.69±0.20	**32.65±0.52**	44.37±0.08	32.20±0.23
5	**45.31±0.22**	32.43±0.33	**44.77±0.11**	**32.47±0.42**

Table 4.6: Combining $\mathcal{L}_{\text{NCA}}(g_2, 5)$ and $\mathcal{L}_{\text{MIXNCA}}(g_2, 5, 0.5)$.

Accuracy (%)	CIFAR100	CIFAR100 Adv.	CIFAR10	CIFAR10 Adv.
$\mathcal{L}_{\text{NCA}}$	58.81±0.21	32.86±0.47	77.58±0.23	48.30±0.39
$\mathcal{L}_{\text{MIXNCA}}$	59.85±0.30	32.06±0.72	79.45±0.20	48.32±0.70
Combined	59.66±0.14	**33.64±0.31**	78.94±0.07	**51.19±0.44**

By referring to Table 4.2, our baseline becomes exactly SimCLR [23] when $G^1 = g_0$, and becomes Debiased+HardNeg [154] when $G^1 = g_2$. From Table 4.3, one can see that when $M \neq 1$, $\mathcal{L}_{\text{NCA}}$ and $\mathcal{L}_{\text{MIXNCA}}$ can both improve upon the baselines($M = 1$) in all metrics (standard/robust/transfer accuracy). When $G^1 = g_0$, $\mathcal{L}_{\text{NCA}}$'s improvement over SimCLR also exemplifies our Theorem 8. Due to page limits, we only select one λ when $\mathcal{L}_{\text{NaCl}} = \mathcal{L}_{\text{MIXNCA}}$ and report results together with the results of $\mathcal{L}_{\text{NaCl}} = \mathcal{L}_{\text{NCA}}$. Full tables can be found in the Appendix B.2. We further verify the performance on TinyImagent and give results in Table 4.5. Notice that now when $G^1 = g_0$, we are using a batch size of $N = 128$ for 200-class TinyImagent task. Therefore, the requirement of $N \geq K - 1$ in Theorem 8 is not fulfilled. However, we can still see improvements when going from $M = 1$ to $M = 2$. Additionally, we combine $\mathcal{L}_{\text{NCA}}$ and $\mathcal{L}_{\text{MIXNCA}}$ in training and give their results in Table 4.6. We see that the robustness performance can be further boosted by 1-3% with the combined loss while keeping similar standard accuracy to $\mathcal{L}_{\text{MIXNCA}}$.

Case (2) $\alpha = 1$**.** In Table 4.4, after setting $\alpha = 1$, we experiment with $G^1 = G^2 = g_2$ since g_2 generally yields better performance in Table 4.3. When $\mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_2, M, \lambda)$, we give the results for $\lambda = 0.5, 0.7, 0.9$ to show an interesting effect:

while $\mathcal{L}_{\text{MIXNCA}}(g_2, M, \lambda = 0.5)$ benefits a lot going from $M = 1$ to $M = 5$ (standard accuracy increases from 56.22% to 59.43%), the improvement is comparatively smaller with $\mathcal{L}_{\text{MIXNCA}}(g_2, M, 0.9)$ (standard accuracy increases from 56.22% to 56.86%). In Figure 4-1, we plot the robust accuracy defined under FGSM attacks [61] along the y-axis. Ideally, one desires a representation network that pushes the performance to the upper-right corner in the 2D accuracy grid (standard-robust accuracy plot). We highlight the results of $\mathcal{L}_{\text{IntNaCl}}$ and $\mathcal{L}_{\text{IntCL}}$ in circles, through which we see that while $\mathcal{L}_{\text{IntCL}}$ can already train representations that are decently robust without sacrificing the standard accuracy on CIFAR100, the standard accuracy on CIFAR10 is inferior to some baselines (HardNeg and Debiased+HardNeg). Comparatively, $\mathcal{L}_{\text{IntNaCl}}$ demonstrates high transfer standard accuracy and wins over the baselines by a large margin on both datasets, proving the ability of learning representation networks that also transfer robustness property. For TinyImagent, we only show the results when $\mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_2, M, 0.5)$ since g_2 generally achieves higher accuracy and combines well with $\mathcal{L}_{\text{MIXNCA}}$. Importantly, with the help of $\mathcal{L}_{\text{NaCl}}$ module, the performance can be boosted from 42.56% to 45.31% while maintaining good robust accuracy 32.43%.

4.3.2 The effect of $\mathcal{L}_{\text{Robust}}$

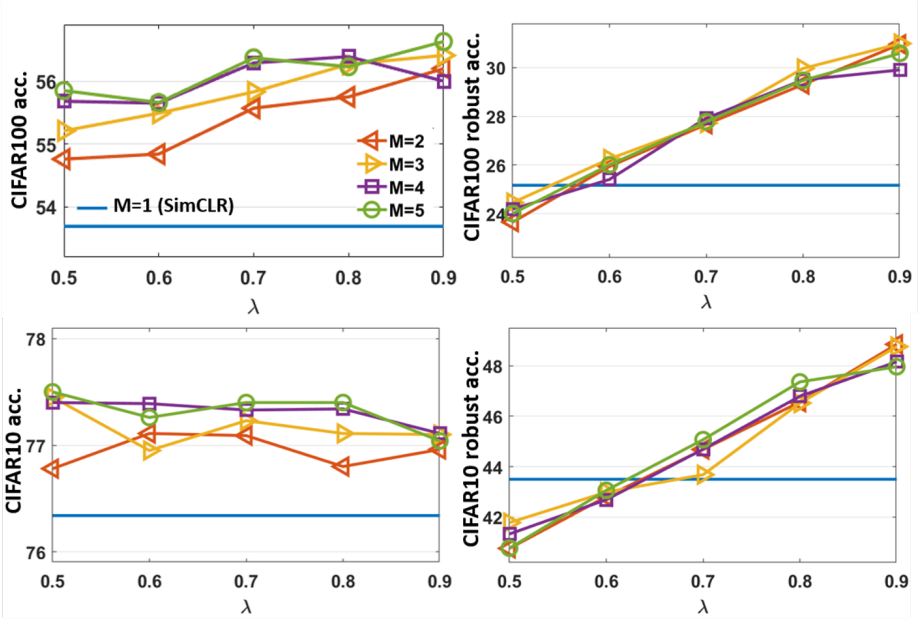
By evaluating the effect of $\mathcal{L}_{\text{Robust}}$, we want to see the performance difference of our framework $\mathcal{L}_{\text{IntNaCl}}$ when $\alpha \neq 0$ and $\alpha = 0$. Therefore, we consider 2 cases: (1) set $M = 1$ in equation 4.14 and compare $\mathcal{L}_{\text{IntCl}}$ with existing work $\mathcal{L}_{\text{NaCl}}(G^1, M = 1, \lambda)$, or (2) set $M \neq 1$ and compare $\mathcal{L}_{\text{IntNaCl}}$ and $\mathcal{L}_{\text{NaCl}}(G^1, M \neq 1, \lambda)$.

Case (1) $M = 1$**.** Notice that $\mathcal{L}_{\text{IntCL}}$ differs from standard contrastive losses by including the term $\mathcal{L}_{\text{Robust}}$. Therefore, one can easily evaluate the effect of $\mathcal{L}_{\text{Robust}}$ by inspecting the performance difference between $\mathcal{L}_{\text{IntCl}}$ and the baselines in Figure 4-1. Specifically, we let $G^1 = g_2$ for $\mathcal{L}_{\text{IntCl}}$ in Figure 4-1, hence a direct baseline is Debiased+HardNeg. By adding a robustness-promoting term, the robust accuracy can be boosted from 31.03% to 40.05% and transfer robust accuracy from 48.38% to 59.33%, which is a significant improvement.

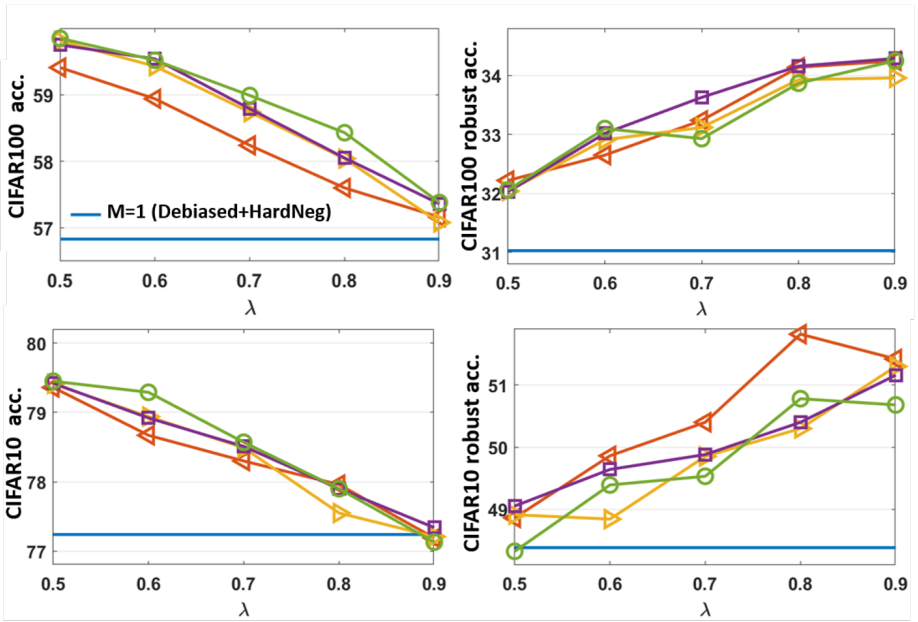
Case (2) $M \neq 1$**.** The effect of $\mathcal{L}_{\text{Robust}}$ is also demonstrated through the robust accuracy "jump" from Table 4.3 to Table 4.4. For example, we point out that in Table 4.3, $\mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{NCA}}(g_2, 3)$ gives the maximum robust accuracy of 33.19%, while the robust accuracy obtained with the same $\mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{NCA}}(g_2, 3)$ and additional $\mathcal{L}_{\text{Robust}}$ increases to 40.53% in Table 4.4. The robust accuracy boost on TinyImagent with the help of $\mathcal{L}_{\text{Robust}}$ is also visible: when $\mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_2, 2, 0.5)$, the robust accuracy increases from 28.29% to 32.65%.

4.3.3 The effect of M, λ, and $w(x)$

To evaluate the effect of M, we can see from Table 4.3 and Table 4.4 that the performance is generally increasing as M increases. However, this effect seems to be less visible for robust accuracy and transfer robust accuracy. In practice, $M = 5$ does not require exactly 5 times training time since the number of training parameter remains the same. In our experiment, we observe that $M = 5$ requires roughly 3 times training time compared with the baseline $M = 1$. To evaluate the effect of λ, we include in Figure 4-2 the standard and robust accuracy on CIFAR100 and CIFAR10 as functions of λ. Intriguingly, we see that the accuracy curves mainly show trends of increasing in Figure 4-2(a). Comparatively, the standard accuracy on CIFAR100 and CIFAR10 shows trends of decreasing in Figure 4-2(b). One possible explanation is by the original baselines' room for improvement. Since Debiased+HardNeg is a much stronger baseline than SimCLR, it is closer to the robustness-accuracy trade-off. However, we note that the overall performance of NaCl on Debiased+HardNeg is still better than NaCl on SimCLR regardless of the robustness-accuracy trade-off. In the last row of Table 4.5, we list the results when $\mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_2, M, 0.5)$ but different $\mathcal{L}_{\text{Robust}}(G^2, w)$). Specifically, on the left we show the case when $w = \hat{w}(x)$ and on the right we show the case when $w = 1$. One can then see that by using a goal-specific weighting scheme, the performance can be further boosted.



(a) NaCl on SimCLR [23], *i.e.* $\alpha = 0, \mathcal{L}_{\text{NaCl}} = \mathcal{L}_{\text{MIXNCA}}, G^1 = g_0$ in equation 4.14



(b) NaCl on Debiased+HardNeg [154], *i.e.* $\alpha = 0, \mathcal{L}_{\text{NaCl}} = \mathcal{L}_{\text{MIXNCA}}, G^1 = g_2$ in equation 4.14

Figure 4-2: The standard and robust accuracy (%) on CIFAR100 and CIFAR10 as functions of λ in Eq. equation 4.14 when $\alpha = 0, \mathcal{L}_{\text{NaCl}} = \mathcal{L}_{\text{MIXNCA}}$.

#epoch	100	200	400	600	800	1000	1200	1400	1600	1800	2000
$\mathcal{L}_{\text{SimCLR}}$	53.69	57.45	60.06	60.96	61.27	61.90	61.94	62.53	62.44	62.10	62.06
$\mathcal{L}_{\text{NCA}}(g_0, 2)$	55.72	59.31	61.19	61.66	62.49	61.95	62.06	62.39	62.39	62.52	62.54
$\mathcal{L}_{\text{MIXNCA}}(g_0, 2, 0.9)$	56.20	58.98	61.81	62.43	62.46	63.48	63.48	64.13	64.14	64.21	64.31
$\mathcal{L}_{\text{Debiased+HardNeg}}$	56.83	59.35	61.77	62.74	62.68	63.12	63.22	63.08	62.86	62.90	63.38
$\mathcal{L}_{\text{NCA}}(g_2, 2)$	57.87	60.06	62.36	62.58	62.86	63.07	63.29	63.65	63.13	63.73	63.20
$\mathcal{L}_{\text{MIXNCA}}(g_2, 2, 0.5)$	59.41	62.14	64.06	65.59	65.53	66.29	66.64	67.14	66.94	67.53	67.85

Table 4.7: The CIFAR100 linear evaluation results (%) after different numbers of training epochs.

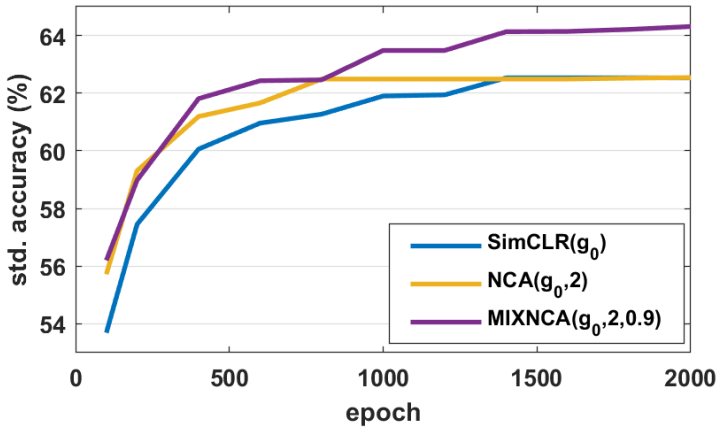
4.3.4 Extended runtime

As training the representation with more epochs can also expose the data to more augmentations, we carry out an additional experiments to compare the efficiency and ultimate accuracy of $\mathcal{L}_{\text{NaCl}}$, $\mathcal{L}_{\text{SimCLR}}$, and $\mathcal{L}_{\text{Debiased+HardNeg}}$. In Table 4.7, we give the standard accuracy of NaCl on SimCLR and NaCl on Debiased+HardNeg at different epochs. Same as before, we only select one λ when $\mathcal{L}_{\text{NaCl}} = \mathcal{L}_{\text{MIXNCA}}$ and report its results together with those of $\mathcal{L}_{\text{NaCl}} = \mathcal{L}_{\text{NCA}}$. In Figure 4-3, we plot the best standard accuracy achieved as a function of training epochs. Specially, [66] has reported a $\mathcal{L}_{\text{SimCLR}}$ CIFAR100 accuracy of 54.74% after 200 epochs, compared to $\mathcal{L}_{\text{NCA}}(g_0, 2)$'s 55.72% after 100 epochs. In our reproduction of the $\mathcal{L}_{\text{SimCLR}}$ 200-epoch result[2], we have witnessed an accuracy of 57.45% however at the cost of 1.34X training time (*cf.* 200 epochs with $\mathcal{L}_{\text{SimCLR}}$ takes 211 mins vs. 100 epochs with $\mathcal{L}_{\text{NCA}}(g_0, 2)$ takes 158 mins). Overall, we see that NaCl methods demonstrate better efficiency when applying on SimCLR and better ultimate accuracy when applying on Debiased+HardNeg.

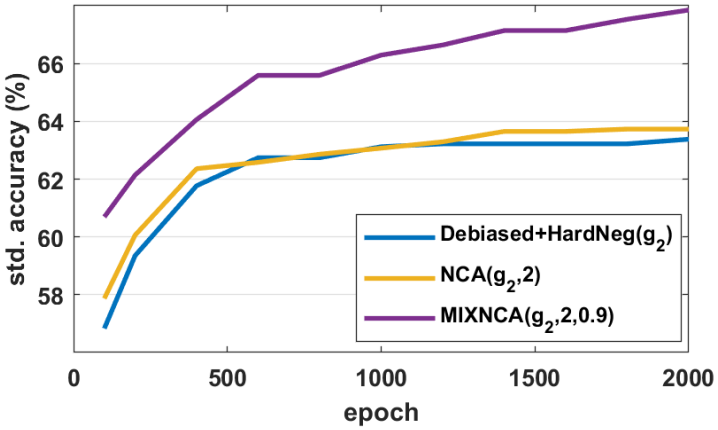
4.4 Conclusion

In this chapter, we discover the relationship between contrastive loss and Neighborhood Component Analysis (NCA), which motivates us to generalize the existing contrastive loss to a set of Neighborhood analysis Contrastive losses (NaCl). We further propose a generic and integrated contrastive learning framework (IntNaCl) based on NaCl,

[2]We let the dataloader shuffle the whole dataset to form new batches after every epoch, so by doubling the training epoch, one will effectively expose the network to more diverse negative pairs.



(a) NaCl on SimCLR [23]



(b) NaCl on Debiased+HardNeg [154]

Figure 4-3: The standard accuracy (%) on CIFAR100 with extended runtime.

which learns representations that score high in both standard accuracy and adversarial accuracy in downstream tasks. With the integrated framework, we can boost the standard accuracy by 6% and the robust accuracy by 17%.

Chapter 5

Evaluating robustness-accuracy of large vision models using synthetic data

5.1 Introduction

In recent years, the use of large pretrained neural networks for efficient fine-tuning on downstream tasks has prevailed in many domains such as vision, language, and speech. Instead of designing task-dependent neural network architectures for different downstream tasks, the current methodology focuses on the principle of task-agnostic pretraining and task-specific finetuning. This methodology uses a neural network pretrained on a large-scale broad dataset to extract generic representations of the input data, which we call *pretrained representations* for simplicity. The pretrained representations are then used as a foundation [12] to solve downstream tasks. Prevalent ways include training a linear head (*i.e.*, linear probing) on the representations with the labels provided by a downstream dataset, or simply employing zero-shot inference.

When gauging the usefulness of a pretrained model, it is a convention to conduct evaluations on selected public datasets. For example, ViT [40] reports accuracy on 25 tasks, CLIP [141] on 27 datasets, and PLEX [181] on over 40 datasets to systematically

evaluate different reliability dimensions on both vision and language domains. However, this convention has several drawbacks. For example, the evaluation process evidently poses significant computational overhead on the model trainer and raises data privacy concerns, setting a high bar for new model designs and large-scale AI governance. More importantly, the evaluation result is dependent on specific evaluation datasets. Thus the nominal evaluation score can be inconclusive if the evaluation data are biased or under-representative. For instance, ViT-L/16 is reportedly performing better than ViT-B/16 on 23 out of 27 linear probing tasks according to [141, Table 10], but worse than ViT-B/16 on FoodSeg103 [202, Table 8], X-ray images [128, Table 4-8], and magnetic resonance imaging [182, Table 2-3] tasks. Fundamentally, a poor probing result might come from either (1) evaluation data bias, (2) true model deficiency, or both. In this chapter, we attempt to disentangle the effect of the two and focus on designing well-posed sanity checks for the latter. We utilize synthetic data generated from class-conditional data prior, whose optimal classification strategy is known, and compare the optimal strategy with representations' linear separability. For example, Fisher's linear discriminant rule [81, 137] decides the optimal strategy for Gaussian distribution. If the data can be separated with 90% accuracy in the raw input space and 60% in the representation space, then the pretrained model has an intrinsic deficiency. Building on that, the trending practice of pretraining and fine-tuning also signifies immediate damage to all adapted applications if the foundation model has hidden risks [12], such as lacking robustness to adversarial examples[1]. Luckily, similar to Fisher's linear discriminant rule for the optimal standard accuracy, [36] has characterized the optimal classification strategy in the presence of input perturbations. Our sanity check can thereby evaluate the robustness of pretrained models by considering the same synthetic conditional Gaussian data prior.

Besides being great candidates for establishing well-posed problems, the idea of probing foundation models with synthetic conditional Gaussians is also motivated by the longstanding practice of Gaussian modeling in signal processing [68], data

[1]These types of risks may not be informed by the standard accuracy as they do not correlate well [168]

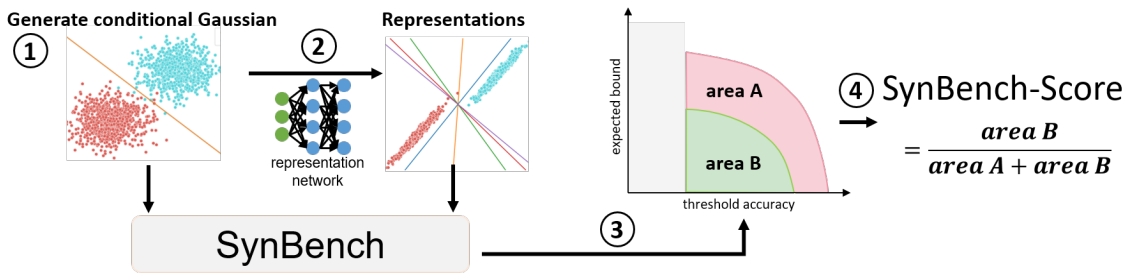


Figure 5-1: Overview of SynBench. Step 1: generate class-conditional Gaussian and form the inputs to the pretrained model; Step 2: gather rendered representations; Step 3: measure the expected robustness bound under a range of threshold accuracy for both input synthetic data and their representations according to **eqn. (5.3)** and obtain the expected bound-threshold accuracy plot; Step 4: calculate SynBench score by the relative area under the curve of the representations (area B) to the inputs (area A + area B) in the expected bound-threshold accuracy plot. The closer the ratio is to 1, the better the quality of pretrained representations is, in terms of the robustness-accuracy characterization.

mining [71], machine learning [90, 183, 221], and other engineering fields. For example, Gaussian mixtures have found applications in modeling noise, magnetic field inhomogeneities, biological variations of tissues in magnetic resonance imaging [145], and computerized tomography [157]. The facts that Gaussian mixture models often lead to mathematically tractable problems [118, 150, 113] and the abundance of analytical tools available for Gaussian models [84, 138, 81, 36] also inspire our study on how Gaussian mixtures can be leveraged for evaluating pretrained image models. Further discussions regarding our choice of Gaussian models can be found in the Section 5.4.2.

An ideal pretrained model should entail both good accuracy and robustness, and the level of goodness is desired to be measurable in a task/data-agnostic manner. In this chapter, we propose *SynBench* to precisely address this requirement. Specifically, SynBench establishes a theoretical reference characterizing the robustness-accuracy trade-off of the synthetic data based on the Bayes optimal linear classifiers. Then, SynBench obtains the representations of the same synthetic data from the pretrained model and compares them to the reference. Finally, we define the ratio of area-under-the-curves in robustness-accuracy plots, *SynBench-Score*, as a quantifiable metric of the pretrained representation quality. The entire procedure of SynBench is illustrated

in Figure 5-1. We list possible use case of SynBench in the Section 5.4.1.

SynBench features the following key advantages:

1. *Soundness*: We formalize the fundamental trade-off in robustness and accuracy of the considered conditional Gaussian model and use this characterization as a reference to analyze the quality of pretrained representations in a completely real-data-free scenario.

2. *Task-independence*: Since the pretraining of large models is independent of the downstream datasets and tasks (e.g., through self-supervised or unsupervised training on broad data at scale), the use of synthetic data in SynBench provides a task-agnostic approach to evaluating pretrained representations without the knowledge of downstream tasks and datasets.

3. *Completeness and privacy*: The flexibility of generating synthetic data (e.g., by adopting a different data sampling procedure) offers a good proxy towards a more comprehensive evaluation of pretrained representations before fine-tuning on downstream datasets, especially in the scenario when the available datasets are not representative of the entire downstream datasets. Moreover, the use of synthetic data enables complete control and simulation over data size and distribution, protects data privacy, and facilitated model auditing and governance.

5.1.1 Our contributions

- We propose SynBench, a novel evaluation framework for pretrained image models that uses data synthesized from a data prior. The evaluation process is independent of the downstream image classification datasets/tasks.

- Evaluated with several pretrained image models for image classification, our experimental results show that SynBench-Score matches well the model performance when finetuned on several downstream datasets. For example, SynBench-Score suggests that the Imagenet21k pretrained network (*ViT-B/16-in21k*) improves with finetuning on Imagenet1k (*ViT-B/16*), echoing with the higher

linear probing accuracy of *ViT-B/16* on real-life datasets. The Pearson correlation coefficient between SynBench-Score and the average downstream task accuracy suggests strong correlation (above 0.9).

- We show that SynBench can be used to guide hyperparameter selection in robust linear probing to mitigate the robustness-accuracy trade-off when fine-tuned on downstream datasets. For example, conducting ϵ-robust linear probing with ϵ selected by SynBench-Score gives *ViT-B/16* 0.1% and 2.7% increase on CIFAR10 standard and robust accuracy, and 0.7% and 2.5% increase on TinyImagenet standard and robust accuracy.

5.1.2 Related works

In the past few years, much focus in the machine learning community has been shifted to training representation networks capable of extracting features for a variety of downstream tasks with minimal fine-tuning. Nowadays, many common vision tasks are achieved with the assistance of good backbones, e.g. classifications [211, 201, 51, 206, 40, 21], object detection [149, 110], segmentation [20, 205], etc. Among the popular backbones, vision transformers (ViT) [40] and convolutional models (e.g. ResNet [70]) have attracted enormous interest. We will exemplify the use of SynBench using several pretrained ViTs and ResNets.

Since pretrained models are used as a foundation for different downstream tasks, it is central to transfer learning [123, 139], and also tightly related to model generalization [140, 18]. To benchmark the performance of a pretrained model, it is a convention to apply the pretrained model for a number of popular tasks and conduct linear probing on the representations [23, 40, 21, 25]. Besides accuracy-based probing methods, evaluation methods have been proposed based on information theory and minimum description length [8, 185], surplus description length [196], maximum evidence [210], Fisher discriminant analysis [162], among others. These metrics are reliant on the label information of the downstream tasks and are hence task-specific.

Lately, more fundamental questions related to pretrained models are brought up

[12, 181, 219, 164]. [12] raised practical concerns about the homogenization incentivized by the scale of the pretraining. Although homogenization might help in achieving competitive performance for some downstream tasks, the defects are also inherited by all these downstreams. On that account, a more careful study of the fundamentals of pretrained models is of paramount importance. [181] explored the reliability of pretrained models by devising 10 types of tasks on 40 datasets. It is further pointed out by [219] in 9 benchmarks that pretrained models may not be robust to subpopulation or group shift. The adversarial robustness is benchmarked by [161, 134].

To enable quantifying representation quality in the pretraining stage, SynBench differs from the above frameworks as it does not need knowledge of any real-world downstream data. Moreover, SynBench has full control of the evaluation set via synthetic data generation. With the assumed synthetic data distribution, we can theoretically characterize the reference robustness-accuracy trade-off. Therefore, SynBench provides a standardized quality metric with theoretical groundings and evaluates for representations induced by pretrained models at a low cost.

5.2 SynBench: methodology and evaluation

Without the knowledge of the downstream tasks and data, we aim to develop a task-agnostic framework to evaluate some fundamental behaviors of the representation network. In this chapter, we inspect and quantify how representation networks preserve the robustness and accuracy enjoyed by the original synthesized data. On the whole, we measure the idealized robustness-accuracy trade-off using synthetic data. By propagating the Gaussian realizations through different representation networks, we can also compare the robustness-accuracy trade-off for representations. We start this section by giving the preliminaries on the synthetic data of interest.

5.2.1 Synthetic data

We consider binary classification problems with data pair (x, y) generated from the mixture of two Gaussian distributions $P_{\mu_1,\mu_2,\Sigma}$, such that $x|y = 1 \sim \mathcal{N}(\mu_1, \Sigma), x|y =$

$-1 \sim \mathcal{N}(\mu_2, \Sigma)$, or equivalently,

$$x - \frac{\mu_1 + \mu_2}{2} | y \sim \mathcal{N}(y\tilde{\mu}, \Sigma), \tag{5.1}$$

where $y \in \mathcal{C} = \{+1, -1\}$, $P(y = +1) = \tau$, $P(y = -1) = 1-\tau$, and $\tilde{\mu} = \frac{\mu_1 - \mu_2}{2}$. We focus on the class-balanced case ($\tau = \frac{1}{2}$) and defer the imbalanced case to Appendix A.3.2. When sampling from this idealized distribution, we eliminate the factor of data bias and can test the accuracy and robustness degradation in an ideal setting.

Let $\| \cdot \|_p$ denote the ℓ_p norm of a vector for any $p \geq 1$. For a given classifier f and input x with $f(x) = y$, where y is the predicted label, it is not rational for the classifier to respond differently to $x + \delta$ than to x for a small perturbation level measured by $\|\delta\|_p$, *i.e.* inconsistent top-1 prediction [171, 61]. Therefore, the level of (adversarial) robustness for a classifier can be measured by the minimum magnitude of perturbation that causes misclassification, *i.e.* $\|\Delta\|_p := \min_{\delta: f(x+\delta) \neq f(x)} \|\delta\|_p$. For a generic function f, solving the optimization problem exactly is hard [85, 166]. Luckily, one can readily solve for the optimization if f is affine [122].

5.2.2 Main theorem

In what follows, we will leverage this point and focus on the linear classifier that minimizes robust classification error. An ideal candidate classifier for the class conditional Gaussian (equation 5.1) is specified by the robust Bayes optimal classifier [4, 39]. Specifically, it is stated that the optimal robust classifier (with a robust margin ϵ) for data generated from equation 5.1 is a linear classifier. We derive the following result as a direct application of the fact. To simplify the exposition, we focus on the ℓ_2 norm in the remainder of this chapter. We refer the readers to Appendix A.3.1 for general ℓ_p-norm results. We use "bound" to denote the minimal perturbation of a sample. We first formally state our theorem that serves as the foundation of our SynBench framework.

Theorem 9. *For any sample x, the optimal robust classifier f_ϵ for $P_{\mu_1,\mu_2,\Sigma}$ gives*

(i) the bound (decision margin) $\|\Delta\|_2 = \frac{|(x - \frac{\mu_1+\mu_2}{2})^T \Sigma^{-1}(\tilde{\mu} - z_\Sigma(\tilde{\mu}))|}{\|\Sigma^{-1}(\tilde{\mu} - z_\Sigma(\tilde{\mu}))\|_2}$,

(ii) the scaled bound $\|\bar{\Delta}\|_2 = \frac{|(x-\frac{\mu_1+\mu_2}{2})^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))|}{|\tilde{\mu}^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))|}$.

For a sample $x \sim P_{\mu_1,\mu_2,\Sigma}$*, it further gives*

(iii) the standard accuracy $a = \Phi(\frac{\tilde{\mu}^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))}{\|\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))\|_\Sigma})$,

(iv) the expected scaled bound of correct samples
$\mathbb{E}\left[\|\bar{\Delta}\|_2 \mid f_\epsilon(x) = y\right] = \frac{1}{\sqrt{2\pi}}\frac{1}{a\Phi^{-1}(a)}e^{-\frac{1}{2}\left(\Phi^{-1}(a)\right)^2} + 1$,

where z_Σ *is the solution of the convex problem* $\arg\min_{\|z\|_2\leq\epsilon}(\tilde{\mu}-z)^T\Sigma^{-1}(\tilde{\mu}-z)$ *and* Φ *denotes the CDF of the standard normal distribution.*

Proof. (i) Following [4, 36], the Bayes optimal robust classifier for the general non-symmetric conditional Gaussians $P_{\mu_1,\mu_2,\Sigma}$ specified in equation 5.1 is

$$f_\epsilon(x) = sign\left\{\left(x - \frac{\mu_1+\mu_2}{2}\right)^T \Sigma^{-1}\left(\tilde{\mu} - z_\Sigma(\tilde{\mu})\right)\right\}, \tag{5.2}$$

where $sign(\cdot)$ is the typical sign function and z_Σ is the solution of the convex problem $\arg\min_{\|z\|_2\leq\epsilon}(\tilde{\mu}-z)^T\Sigma^{-1}(\tilde{\mu}-z)$. The corresponding decision boundary is at $\left((x+\delta) - \frac{\mu_1+\mu_2}{2}\right)^T \Sigma^{-1}\left(\tilde{\mu} - z_\Sigma(\tilde{\mu})\right) = 0$,

$$\begin{aligned}
\implies \quad \Delta \quad &= \arg\min \|\delta\|_2 \quad \text{s.t.} \quad \delta^T\Sigma^{-1}\left(\tilde{\mu} - z_\Sigma(\tilde{\mu})\right) = -\left(x - \frac{\mu_1+\mu_2}{2}\right)^T \Sigma^{-1}\left(\tilde{\mu} - z_\Sigma(\tilde{\mu})\right) \\
\implies \quad \|\Delta\|_2 &= \frac{|(x-\frac{\mu_1+\mu_2}{2})^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))|}{\|\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))\|_2}.
\end{aligned}$$

(ii) Since the bound $\|\Delta\|_2$ is subject to the positions of two Gaussians, we scale the bound by the distance from Gaussian centers to the classifier, $\frac{|\tilde{\mu}^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))|}{\|\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))\|_2}$ and obtain

$$\begin{aligned}
\|\bar{\Delta}\|_2 &= \frac{|(x-\frac{\mu_1+\mu_2}{2})^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))|}{\|\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))\|_2}\frac{\|\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))\|_2}{|\tilde{\mu}^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))|} \\
&= \frac{|(x-\frac{\mu_1+\mu_2}{2})^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))|}{|\tilde{\mu}^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))|}.
\end{aligned}$$

(iii) For sample $x \sim P_{\mu_1,\mu_2,\Sigma}$, consider the Bayes optimal robust classifier in equa-

tion 5.2, we can calculate the analytical standard accuracy by

$$
\begin{aligned}
&\mathbb{P}(y=1)\mathbb{P}\left[f_\epsilon(x)=1 \mid y=1\right]+\mathbb{P}(y=-1)\mathbb{P}\left[f_\epsilon(x)=-1 \mid y=-1\right] \\
=&\mathbb{P}\left[f_\epsilon(x)=1 \mid y=1\right] \\
=&\mathbb{P}\left[(x-\frac{\mu_1+\mu_2}{2})^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))>0 \mid y=1\right] \\
=&\mathbb{P}\left[(\tilde{\mu}+w)^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))>0\right], \quad w\sim\mathcal{N}(0,\Sigma) \\
=&\mathbb{P}\left[w^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))>-\tilde{\mu}^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))\right], \quad w\sim\mathcal{N}(0,\Sigma) \\
=&\mathbb{P}\left[\frac{w^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))}{\|\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))\|_\Sigma}>-\frac{\tilde{\mu}^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))}{\|\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))\|_\Sigma}\right], \quad \frac{w^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))}{\|\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))\|_\Sigma}\sim\mathcal{N}(0,1) \\
=&\Phi(\frac{\tilde{\mu}^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))}{\|\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))\|_\Sigma}).
\end{aligned}
$$

(iv) For sample $x\sim P_{\mu_1,\mu_2,\Sigma}$, let a denote the accuracy, t denote $x-\frac{\mu_1+\mu_2}{2}$, and w denote $\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))$. From (iii), we have that the standard accuracy of conditional Gaussian samples with the Bayes optimal (robust) classifier is $\Phi(\frac{\tilde{\mu}^Tw}{\|w\|_\Sigma})$, so $\frac{\tilde{\mu}^Tw}{\|w\|_\Sigma}=\Phi^{-1}(a)$. Since for binary classification, we only care about accuracy from 0.5 to 1, so we should have $\tilde{\mu}^Tw>0$.

Now consider the classifier in equation 5.2 and the corresponding scaled bound from (ii),

$$
\|\bar{\Delta}\|_2=\frac{|(x-\frac{\mu_1+\mu_2}{2})^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))|}{|\tilde{\mu}^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))|}=\frac{|t^Tw|}{|\tilde{\mu}^Tw|}=\frac{|t^Tw|}{\tilde{\mu}^Tw}.
$$

Since $t|y\sim\mathcal{N}(y\tilde{\mu},\Sigma)$, we have $t^Tw|y\sim\mathcal{N}(y\tilde{\mu}^Tw,w^T\Sigma^Tw)$. When we only want to get the expected scaled bound of the correctly-classified samples, we have that

$$
\begin{aligned}
\mathbb{E}\left[\|\bar{\Delta}\|_2 \mid f_\epsilon(x)=y\right]&=\frac{1}{\tilde{\mu}^Tw}\mathbb{E}\left[|t^Tw| \mid f_\epsilon(x)=y\right] \\
&=\frac{1}{2\tilde{\mu}^Tw}\mathbb{E}\left[|t^Tw| \mid f_\epsilon(x)=y=1\right]+\frac{1}{2\tilde{\mu}^Tw}\mathbb{E}\left[|t^Tw| \mid f_\epsilon(x)=y=-1\right] \\
&=\frac{1}{2\tilde{\mu}^Tw}\mathbb{E}\left[t^Tw \mid y=1,t^Tw\geq 0\right]+\frac{1}{2\tilde{\mu}^Tw}\mathbb{E}\left[-t^Tw \mid y=-1,t^Tw<0\right].
\end{aligned}
$$

Recall that $t^Tw|y\sim\mathcal{N}(y\tilde{\mu}^Tw,w^T\Sigma^Tw)$, then by the mean of truncated normal

distribution, it is true that

$$\begin{aligned}
\mathbb{E}\left[t^T w \mid y=1, t^T w \geq 0\right] &= \tilde{\mu}^T w + \sqrt{w^T \Sigma^T w} \frac{\phi(\frac{0-\tilde{\mu}^T w}{\sqrt{w^T \Sigma^T w}})}{1-\Phi(\frac{0-\tilde{\mu}^T w}{\sqrt{w^T \Sigma^T w}})} \\
&= \tilde{\mu}^T w + \sqrt{w^T \Sigma^T w} \frac{\phi(-\frac{\tilde{\mu}^T w}{\sqrt{w^T \Sigma^T w}})}{1-\Phi(-\frac{\tilde{\mu}^T w}{\sqrt{w^T \Sigma^T w}})} \\
&= \tilde{\mu}^T w + \sqrt{w^T \Sigma^T w} \frac{1}{\sqrt{2\pi}\Phi(\frac{\tilde{\mu}^T w}{\sqrt{w^T \Sigma^T w}})} e^{-\frac{1}{2}\left(\frac{\tilde{\mu}^T w}{\sqrt{w^T \Sigma^T w}}\right)^2} \\
\mathbb{E}\left[-t^T w \mid y=-1, t^T w < 0\right] &= -\mathbb{E}\left[t^T w \mid y=-1, t^T w < 0\right] \\
&= -\left(-\tilde{\mu}^T w - \sqrt{w^T \Sigma^T w} \frac{\phi(\frac{0+\tilde{\mu}^T w}{\sqrt{w^T \Sigma^T w}})}{\Phi(\frac{0+\tilde{\mu}^T w}{\sqrt{w^T \Sigma^T w}})}\right) \\
&= \tilde{\mu}^T w + \sqrt{w^T \Sigma^T w} \frac{1}{\sqrt{2\pi}\Phi(\frac{\tilde{\mu}^T w}{\sqrt{w^T \Sigma^T w}})} e^{-\frac{1}{2}\left(\frac{\tilde{\mu}^T w}{\sqrt{w^T \Sigma^T w}}\right)^2}.
\end{aligned}$$

Therefore

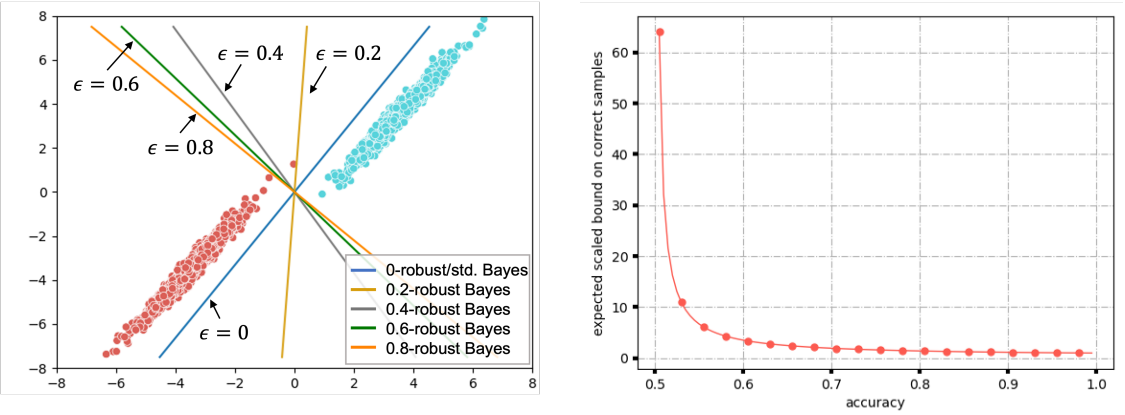
$$\begin{aligned}
\mathbb{E}\left[\|\bar{\Delta}\|_2 \mid f_\epsilon(x)=y\right] &= \frac{1}{\tilde{\mu}^T w}\left(\tilde{\mu}^T w + \sqrt{w^T \Sigma^T w} \frac{1}{\sqrt{2\pi}\Phi(\frac{\tilde{\mu}^T w}{\sqrt{w^T \Sigma^T w}})} e^{-\frac{1}{2}\left(\frac{\tilde{\mu}^T w}{\sqrt{w^T \Sigma^T w}}\right)^2}\right) \\
&= 1 + \frac{\sqrt{w^T \Sigma^T w}}{\tilde{\mu}^T w} \frac{1}{\sqrt{2\pi}\Phi(\frac{\tilde{\mu}^T w}{\sqrt{w^T \Sigma^T w}})} e^{-\frac{1}{2}\left(\frac{\tilde{\mu}^T w}{\sqrt{w^T \Sigma^T w}}\right)^2}.
\end{aligned}$$

By replacing $\frac{\tilde{\mu}^T w}{\sqrt{w^T \Sigma^T w}}$ by $\Phi^{-1}(a)$, we got

$$\mathbb{E}\left[\|\bar{\Delta}\|_2 \mid f_\epsilon(x)=y\right] = \frac{1}{\sqrt{2\pi}} \frac{1}{a\Phi^{-1}(a)} e^{-\frac{1}{2}\left(\Phi^{-1}(a)\right)^2} + 1.$$

□

We note that for samples drawn from $P_{\mu_1,\mu_2,\Sigma}$, $\Sigma = \sigma^2 I_d$, all ϵ-robust Bayes optimal classifier overlap with each other. For a general covariance Σ, the ϵ of an ϵ-robust Bayes classifier specifies the desired size of margin and demonstrates the robustness accuracy trade-off. We give an illustrative 2D class-conditional Gaussian example in Figure 5-2(a), where different ϵ-robust Bayes classifiers give different overall margins at the cost of accuracy. As ϵ increases, the robust Bayes optimal classifier rotates



(a) 2D Gaussian case

(b) Rob.-Acc. trade-off

Figure 5-2: Illustration of robustness-accuracy trade-off suggested by ϵ-robust Bayes optimal classifiers. Figure (a) depicts a class-conditional 2D Gaussian case with decision boundaries drawn by ϵ-robust Bayes optimal classifiers of varying ϵ values. Figure (b) draws the theoretically characterized robustness-accuracy trade-off given in Theorem 9(iv).

counterclockwise, leading to increased misclassifications, but also overall enlarged margins.

5.2.3 Objective

For a given representation network parameterized by θ, we are interested in evaluating the expected bounds on synthetic data and their representations, under a thresholding accuracy a_t. That is, $\mathbb{E}_{\mu\sim\mathbb{P}_\mu,\Sigma\sim\mathbb{P}_\Sigma,x-\bar{\mu}|y\sim\mathcal{N}(y\mu,\Sigma)}\left[\|\bar{\Delta}\|_2 \mid f_\epsilon(x)=y, a>a_t\right]$ for $\bar{\Delta}=\bar{\Delta}_x$ and $\bar{\Delta}_z$, where $\mathbb{P}_\mu$ and $\mathbb{P}_\Sigma$ characterize the probability density function of the synthetic data manifold of interest, $\bar{\mu}$ is a translation vector allowing non-symmetric class conditional Gaussian, and $\bar{\Delta}_x$ and $\bar{\Delta}_z$ denote the bounds on synthetic data and representations respectively. Here, without the prior of applications, we assume $\mu = s\cdot 1_d/\sqrt{d}$, where s denotes a random variable that follows uniform distribution and $1_d/\sqrt{d}$ is the normalized all-ones vector. For simplicity, we let $\Sigma = I_d$. Formally,

we define the accuracy-constrained expected bound $E_{\theta,\epsilon}(a_t)$ as

$$\begin{aligned}
E_{\theta,\epsilon}(a_t) =& \mathbb{E}_{s\sim\mathcal{U},x-\bar{\mu}|y\sim\mathcal{N}(\mu,\Sigma)}\left[\|\bar{\Delta}\|_2 \mid f_\epsilon(x)=y, a>a_t, \mu=s\cdot 1_d/\sqrt{d}, \Sigma=I_d\right] \\
=& \mathbb{E}_{s,x}\left[\|\bar{\Delta}\|_2 \mid f_\epsilon(x)=y, a(s,\epsilon)>a_t\right] \\
=& \sum_i \mathbb{E}_x\left[\|\bar{\Delta}\|_2 \mid f_\epsilon(x)=y, a(s_i,\epsilon)>a_t\right]\mathbb{P}(s=s_i) \\
=& \frac{1}{n}\sum_i \mathbb{E}_x\left[\|\bar{\Delta}\|_2 \mid f_\epsilon(x)=y, a(s_i,\epsilon)>a_t\right] \\
=& \frac{1}{n}\sum_i \mathbb{E}_x\left[\|\bar{\Delta}\|_2 \mid f_\epsilon(x)=y\right]\mathbb{1}_{\mathrm{a(s_i,\epsilon)>a_t}}. \qquad (5.3)
\end{aligned}$$

where $\mathbb{1}_{\mathrm{a(s_i,\epsilon)>a_t}}$ is the indicator function specifying the s_i, ϵ-dependent accuracy a that surpasses the threshold accuracy a_t.

In the following sections, we will illustrate how to calculate the inner expectation term $\mathbb{E}_x\left[\|\bar{\Delta}\|_2 \mid f_\epsilon(x)=y\right]$ for both the raw data (synthetic data) and representations.

Raw data. For raw data synthesized from $P_{\mu_1,\mu_2,\Sigma}$ according to equation 5.1, the inner expectation term is given by Theorem 9(iv) $\mathbb{E}\left[\|\bar{\Delta_x}\|_2 \mid f_\epsilon(x)=y\right] = \frac{1}{\sqrt{2\pi}}\frac{1}{a\Phi^{-1}(a)}e^{-\frac{1}{2}\left(\Phi^{-1}(a)\right)^2}+1$, where a denotes the standard accuracy. The subscript x in the expected scaled bound $\mathbb{E}\left[\|\bar{\Delta_x}\|_2 \mid f_\epsilon(x)=y\right]$ indicates the raw data space, to distinguish from the scaled bound to be derived for representations. We highlight that Theorem 9(iv) directly shows a robustness-accuracy trade-off. We plot the expected scaled bound as a function of accuracy in Figure 5-2(b), which holds true when the data follow equation 5.1 exactly. In SynBench, we treat this theoretically-derived robustness-accuracy trade-off as the reference, enabling a fair comparison among representations induced by different pretrained models.

Representations. Given a pretrained network, we gather the representations of the Gaussian realizations and quantify the bound induced by robust Bayes optimal classifier in the representation space. When deriving the robust Bayes optimal classifier, we model the representations by a general conditional Gaussian $z|y=1 \sim \mathcal{N}(\mu_1,\Sigma), z|y=-1 \sim \mathcal{N}(\mu_2,\Sigma)$. By Theorem 9(ii), we consider the optimal robust classifier for the modeled conditional Gaussian in the representation space to calculate

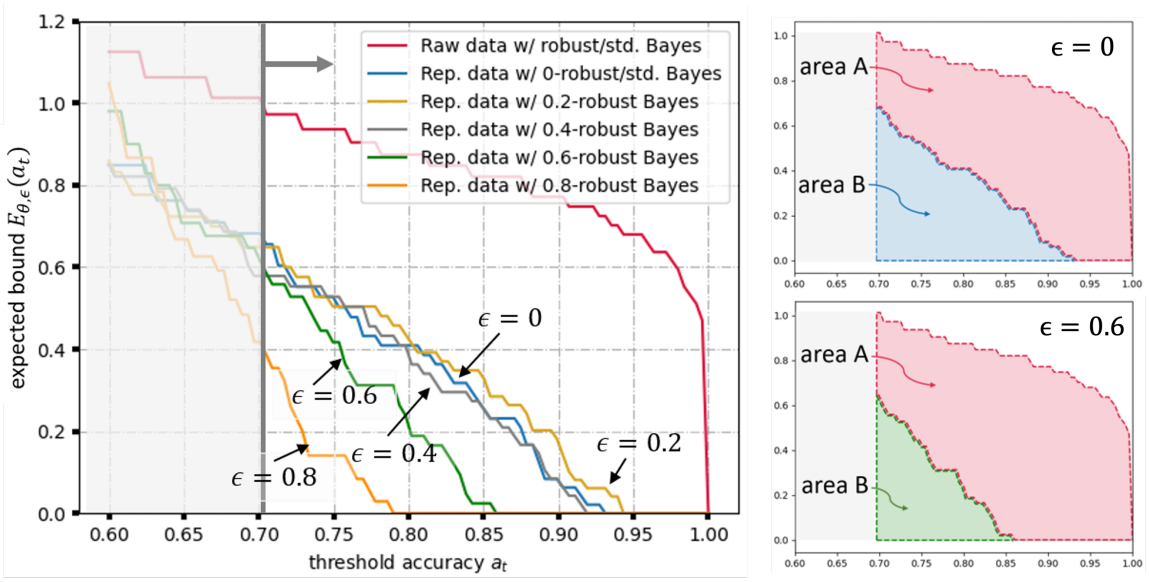


Figure 5-3: An example of the robustness-accuracy quantification of representations for ViT-B/16. (Left) The expected bound-threshold accuracy plot for the input raw data ($E(a_t)$) and representations ($E_{\theta,\epsilon}(a_t)$) with $\epsilon = 0 \sim 0.8$. (Right) To calculate the SynBench-Score for $\epsilon = 0$ (top) and $\epsilon = 0.6$ (bottom), we use the definition SynBench-Score$(\theta, \epsilon, a_t) = \frac{\text{area B}}{\text{area A+area B}}$ (refer to equation 5.4), which gives SynBench-Score$(\theta_{\text{ViT-B/16}}, 0, 0.7) = 0.33$ and SynBench-Score$(\theta_{\text{ViT-B/16}}, 0.6, 0.7) = 0.20$.

the scaled bound $\|\bar{\Delta}_z\|_2 = \frac{|(z - \frac{\mu_1+\mu_2}{2})^T \Sigma^{-1}(\tilde{\mu} - z_\Sigma(\tilde{\mu}))|}{|\tilde{\mu}^T \Sigma^{-1}(\tilde{\mu} - z_\Sigma(\tilde{\mu}))|}$ for correctly-classified samples and the inner expectation is estimated empirically. It should be noted that now the Bayes optimal classifier does not necessarily coincide with the robust Bayes optimal classifier even when we synthesized the dataset with an identity matrix covariance in the input space.

5.2.4 Robustness-accuracy quantification

Recall that we aim to calculate

$$E_{\theta,\epsilon}(a_t) = \sum_i \mathbb{E}_{x|y\sim\mathcal{N}(ys_i\cdot 1_d/\sqrt{d}, I_d)} \left[\|\bar{\Delta}\|_2 \mid f_\epsilon(x) = y \right] \mathbb{1}_{\text{a}(\text{s}_\text{i},\epsilon)>\text{a}_\text{t}} p(s_i)$$

for both raw data and the representations (*i.e.* $\|\bar{\Delta}_x\|$ and $\|\bar{\Delta}_z\|$). We treat the expected bounds of the raw data under a threshold accuracy as the reference. Given

Algorithm 1 Evaluating Pretrained Image Representations using Synthetic Data (*SynBench*)

Input A representation network $g_\theta : \mathbb{R}^d \to \mathbb{R}^{d'}$, threshold accuracy a_T, (optional) the probability density function of the synthetic data manifold $\mathbb{P}_\mu$ and $\mathbb{P}_\Sigma$.
Output: SynBench-score that quantifies the robustness-accuracy performance.

if $\mathbb{P}_\mu$ and $\mathbb{P}_\Sigma$ are specified **then**
  $\mu \sim \mathbb{P}_\mu, \Sigma \sim \mathbb{P}_\Sigma$.
else
  $\mu = s \cdot 1_d/\sqrt{d}$, $s \sim \mathcal{U}\{0.1, 5\}$, and $\Sigma = I_d$.
end if
Draw n synthetic data hyper-parameters $\{(\mu_k, \Sigma_k)\}_{k=1}^n$.
for $k \leftarrow 1$ to n **do**
  Generate class-conditional Gaussian data (x^{train}, y^{train}) and test set (x^{test}, y^{test}) following $x - \bar{\mu}|y \sim \mathcal{N}(y\mu_k, \Sigma_k)$ and $\bar{\mu} = 0.5 \cdot 1_d/\sqrt{d}$.
  Calculate a_k^{input}, the theoretical accuracy for input data, following Thm 9(iii).
  Calculate b_k^{input} (denotes $\mathbb{E}\left[\|\bar{\Delta}_x\|_2 \mid f_\epsilon(x) = y\right]$), the expected scaled bound of correct samples for input data, following Thm 9(iv).
  Gather representations for class 1 training samples $z_1^{train,i} = g_\theta(x^{train,i})$ if $y^{train,i} = 1$, representations for class 2 training samples $z_2^{train,j} = g_\theta(x^{train,j})$ if $y^{train,j} = -1$, and $z^{test} = g_\theta(x^{test})$.
  Estimate class-conditional Gaussian in the representation space by $\mu_1' = \frac{\sum_{i=1}^{n_1} z_1^{train,i}}{n_1}$, $\mu_2' = \frac{\sum_{j=1}^{n_2} z_2^{train,j}}{n_2}$, $\Sigma' = \frac{\sum_{i=1}^{n_1}(z_1^{train,i}-\mu_1')(z_1^{train,i}-\mu_1')^T + \sum_{j=1}^{n_2}(z_2^{train,j}-\mu_2')(z_2^{train,j}-\mu_2')^T}{n_1+n_2-1}$.
  Derive Bayes optimal classifier f_ϵ' for class-conditional Gaussian distribution $z|y=1 \sim \mathcal{N}(\mu_1', \Sigma'), z|y=-1 \sim \mathcal{N}(\mu_2', \Sigma')$.
  Calculate a_k^{repre}, the accuracy of f_ϵ' for representations z^{test}, empirically.
  Calculate the scaled bound of correct samples for representations following Thm 9(ii), $\|\bar{\Delta}_z\|_2 = \frac{|(z^{test} - \frac{\mu_1'+\mu_2'}{2})^T \Sigma'^{-1}(\tilde{\mu} - z_{\Sigma'}(\tilde{\mu}))|}{|\tilde{\mu}^T \Sigma'^{-1}(\tilde{\mu} - z_{\Sigma'}(\tilde{\mu}))|}$ where $\tilde{\mu} = \frac{\mu_1' - \mu_2'}{2}$.
  Estimate b_k^{repre}, the expected scaled bound of correct samples for representations empirically, by the arithmetic mean.
end for
Calculate $E(a_t)$ for input data with $\{a_k^{\text{input}}, b_k^{\text{input}}\}_{k=1}^n$ according to equation 5.3.
Calculate $E_{\theta,\epsilon}(a_t)$ for representations with $\{a_k^{\text{repre}}, b_k^{\text{repre}}\}_{k=1}^n$ according to equation 5.3.
Calculate SynBench-Score$(\theta, \epsilon, a_T) = \frac{\int_{a_T}^1 E_{\theta,\epsilon}(a_t) da_t}{\int_{a_T}^1 E(a_t) da_t}$.

a representation network, we compare the expected bounds of the representations rendered by representation networks with the reference.

In our implementation, we take $s \sim \mathcal{U}\{0.1, 5\}$ under the guidance of Theorem 9(iii).

Specifically, as Theorem 9(iii) gives an analytical expected accuracy for class conditional Gaussian, we can obtain the desired range of s by giving the accuracy. Since we are interested in having the reference as a class conditional Gaussian that yields accuracy from 55% to almost 100%, we set the starting and ending s by the fact that $\Phi(0.1) \approx 0.55$ and $\Phi(5) \approx 1.0$. We reiterate that with more accurate modeling of the data manifold of interest, SynBench can give a more precise capture of the pretrained representation performance. We will demonstrate this point in Section 5.3.4.

When the data is perfect Gaussian (e.g. input synthetic data), we calculate $E_{\theta,\epsilon}(a_t)$ as detailed in Section 5.2.3. We note that $\bar{\Delta}_x$ is independent of pretrained network parameters θ, and all the ϵ-robust classifiers f_ϵ in the input space overlap with each other when $\Sigma = I_d$. We hereby denote the desired metric on the input synthetic data by $E(a_t)$, to distinguish from that on the representations $E_{\theta,\epsilon}(a_t)$. For representations, we calculate $E_{\theta,\epsilon}(a_t)$ following Section 5.2.3 and the expectation is estimated empirically. We show an example of the probing results in Figure 5-3.

To integrate over all the desired threshold accuracy, we use the area under the curve (AUC) and give the ratio to the reference by

$$\text{SynBench-Score}(\theta, \epsilon, a_T) = \frac{\int_{a_T}^{1} E_{\theta,\epsilon}(a_t) da_t}{\int_{a_T}^{1} E(a_t) da_t}, \tag{5.4}$$

which correspond to the relative area $\frac{\text{area B}}{\text{area A + area B}}$ in Figure 5-3. Values of SynBench-Score closer to 1 imply better probing performance on pretrained representations. To summarize, SynBench framework generates a sequence of proxy tasks with different difficulty levels (monitored by s). With each proxy task, we can obtain an accuracy and an expected bound (Section 5.2.3). With gathered pairs of accuracy and expected bound, we filter ones whose accuracy is below a threshold accuracy (x-axis), and calculate the accuracy-constrained expected bound to reflect the robustness level (y-axis). With this, the AUC will counter for the discriminative power of the foundation model given an idealized distribution, as well as the robustness level. We refer readers to Algorithm 1 for the pseudo-code.

5.3 Experimental results

In Section 5.3.1, we give the setup of our experiments. We exemplify the use of SynBench in making efficient comparisons of pretrained representations in Section 5.3.2. We compare SynBench with baseline methods and demonstrate the supremacy of SynBench-Score in giving consistent model suggestions and high correlation with performance on possible downstream tasks. In Section 5.3.3, we study how SynBench can be used to select robust linear probing hyper-parameters. In Section 5.3.4, we show how to model the covariance matrix Σ used for synthesizing Gaussian samples given prior knowledge of the downstream data distribution.

5.3.1 Experiment setups

In the following sections, we will calculate SynBench-Scores for pretrained models and make pair-wise comparisons. For example, ViT-B/16 is a fine-tuned pretrained model from ViT-B/16-in21k. By checking their SynBench-Scores, we could understand how the fine-tuning procedure helps or worsens the performance.

Setups. In order to systematically understand how each network attribute affects the robustness-accuracy performance, it is desirable to control the variates. We list and compare 10 pretrained vision transformers (ViTs) [40, 25, 19] and ResNets [24] in Table 5.1.

Baselines. Although to the best of our knowledge, there is no real-data-free evaluation method for pretrained representations, we refer to recent work [196, 210, 162] and report the validation accuracy (Val loss), minimum description length (MDL), surplus description length (SDL), logarithm of maximum evidence (LogME) and self-challenging Fisher discriminant analysis (SFDA), following the official implementation from the literature on our synthetic proxy task as baselines [196, 162]. In essence, we expect these real-data-free evaluations for pretrained models can give meaningful performance assessments of possible downstream tasks. For this purpose, we take an average of the accuracy in 27 downstream tasks (*cf.* [141], Table 10) as

Table 5.1: Model descriptions. The performance of models might be nuanced by scheduler, curriculum, and training episodes, which are not captured in the table.

Model	Arch.	pretraining	fine-tuning	patch	# parameters (M)
ViT-Ti/16	ViT-Tiny	Imgn21k	Imgn1k	16	5.7
ViT-B/16	ViT-Base	Imgn21k	Imgn1k	16	86.6
ViT-B/16-in21k	ViT-Base	Imgn21k	No	16	86.6
ViT-L/16	ViT-Large	Imgn21k	Imgn1k	16	304.3
ViT-S/16-DINO	ViT-Small	self-Imgn1k	No	16	21.7
ViT-S/8-DINO	ViT-Small	self-Imgn1k	No	8	21.7
ViT-B/16-DINO	ViT-Base	self-Imgn1k	No	16	85.8
ViT-B/8-DINO	ViT-Base	self-Imgn1k	No	8	85.8
Resnet50-SimCLRv2	Resnet50	self-Imgn1k	No	-	144.4
Resnet101-SimCLRv2	Resnet101	self-Imgn1k	No	-	261.2
Variation:					
Model size	ViT-{Ti,B,L}/16, ViT-{S,B}/16-DINO, ViT-{S,B}/8-DINO, Resnet{50,101}-SimCLRv2				
Finetuning	ViT-B/16{,-in21k}				
ViT patch size	ViT-S/{16,8}-DINO, ViT-B/{16,8}-DINO				

in the literature [40, 141, 107, 50, 211] to give a sense of the general performance on possible downstream tasks, and report the Pearson correlation coefficients with SynBench-Scores. Building on top of these, we also show the consistency of SynBench suggestions given different numbers of synthetic realizations compared to the baselines.

Besides the SynBench-Score, we will also report the standard accuracy (SA) and robust accuracy against adversarial perturbations (RA) for studying robustness-accuracy performance.

Runtime analysis. The runtime of SynBench depends on the number of outcomes of the discrete uniform distribution $\mathcal{U}\{0.1, 5\}$ and the data inference time through the pretrained model. For one outcome (one robustness-accuracy relationship), it costs 59 seconds to generate 2048 Gaussian samples, 37 and 81 seconds to obtain the SynBench-Score for ViT-B/16 and ViT-L/16 on one GeForce RTX 2080 super.

Correspondingly, to obtain one robustness-accuracy relationship with task-specific methods requires us to perform adversarial attacks on multiple possible datasets. Here, we ignore the time to train the linear probing layer. For one single dataset, e.g. CIFAR10, AutoAttack uses 72320 and 332288 seconds to evaluate 2048 samples on ViT-B/16 and ViT-L/16 on one GeForce RTX 2080 super; PGD attack uses 1280 and 4608 seconds to evaluate 2048 samples on ViT-B/16 and ViT-L/16 on one GeForce

Table 5.2: The SynBench-Score of pretrained representations and the standard/robust accuracy (SA/RA) (%) of their linear probing classifier on class-conditional Gaussian data.

Models	SynBench-Score ($\epsilon = 0$)	SA	RA
ViT-Ti/16	0.01	76.0	50.8
ViT-B/16	0.33	96.4	52.9
ViT-B/16-in21k	0.20	92.1	51.3
ViT-L/16	0.26	96.1	52.9
ViT-S/16-DINO	0.48	97.9	55.5
ViT-B/16-DINO	0.55	99.3	50.4
ViT-S/8-DINO	0.40	95.8	51.1
ViT-B/8-DINO	0.50	98.8	49.6
Res50-SimCLRv2	0.66	99.8	50.1
Res101-SimCLRv2	0.60	99.4	51.6

RTX 2080 super.

For other task-agnostic metrics (MDL, SDL, ϵSC), obtaining them for ViT-B/16 costs 6807 seconds and ViT-L/16 costs 7373 seconds on one Tesla V100. However, it should be noted that these metrics do not indicate robustness performance.

To provide a comprehensive evaluation, we give SynBench-Score(θ, ϵ, a_t) with a_t ranging from 0.7 to 0.9, and ϵ from 0 to 0.8. $a_t \neq 0.7$ and some ϵ results are deferred to the appendix.

5.3.2 SynBench analysis of pretrained representations

Comparing model attributes. We list the SynBench-Score of the 10 pretrained representations with their standard and robust accuracy on the class-conditional Gaussian proxy task in Table 5.2. The robust accuracy is obtained by ℓ_2 PGD attack [115] with attack strength 0.2.

By referring to rows "ViT-B/16" and "ViT-B/16-in21k", we see that SynBench will suggest ViT-B/16 over ViT-B/16-in21k, implying that the fine-tuning is beneficial on ViT-B/16-in21k - both networks are pretrained on Imagenet 21k with supervision, whereas ViT-B/16 is further finetuned on Imagenet 1k. We can also use SynBench to evaluate the effect of model sizes. Specifically, we refer to rows "ViT-Ti/16", "ViT-B/16", "ViT-L/16", and see that ViT-B/16 and ViT-L/16 score much higher than ViT-Ti/16, suggesting larger models have better capacities for robustness and accuracy. It is noticeable that ViT-B/16 is generally on par with ViT-L/16 when

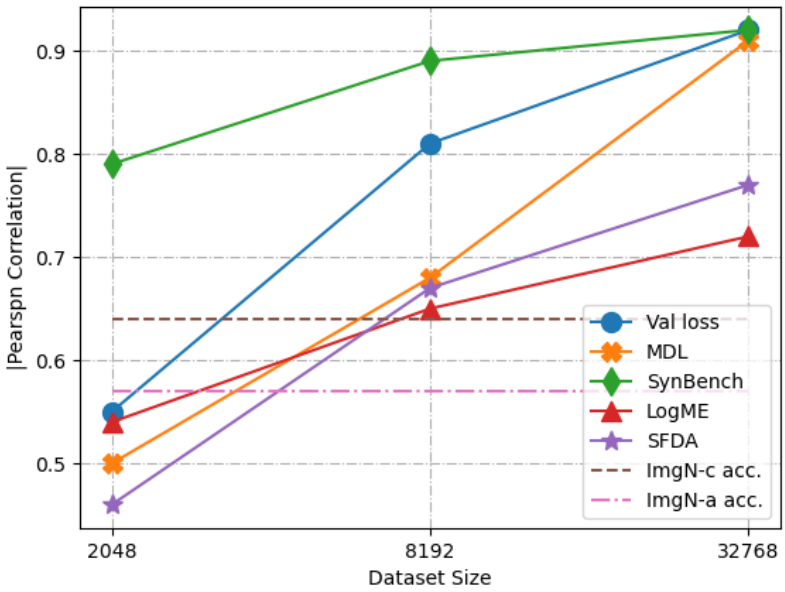


Figure 5-4: Pearson correlation between task-agnostic metrics (Val loss, MDL, SynBench, LogME, SFDA) and task-specific metrics (the average accuracy on 27 real-life tasks) as functions of the dataset size. Two dashed lines characterize the correlation by transfer datasets' accuracy.

we vary ϵ (*cf.* Appendix Table B.4). Similar conclusions can also be drawn by referring to self-supervised pretrained representations, rows "ViT-S/-DINO" and "ViT-B/-DINO". Moreover, if we check rows "ViT-B/16" and "ViT-B/16-DINO", we compare two pretrained models of the same architecture but trained under different regimes, either supervised or self-supervised. Between these two models, SynBench favors self-supervised trained "ViT-B/16-DINO", echoing with the inductive bias of self-supervised contrastive learning discovered in recent literature [65].

SynBench shows better correlation with real-data probing accuracy and robustness. We run baseline evaluations as described in Section 5.3.1 for the synthetic classification task on pretrained models with dataset size n being $2048, 8192, 32768$ and list their results in Appendix Table B.5. Throughout our experiments, we use 2048 test samples in the synthetic dataset. For Val loss, MDL, and SDL, ϵSC, the smaller the better; for LogME, SFDA, SynBench, the bigger the better. In Figure 5-4, we illustrate how the correlation between task-agnostic evaluation metrics and real-life data tasks varies with the dataset size n. Specifically, we calculate the Pearson

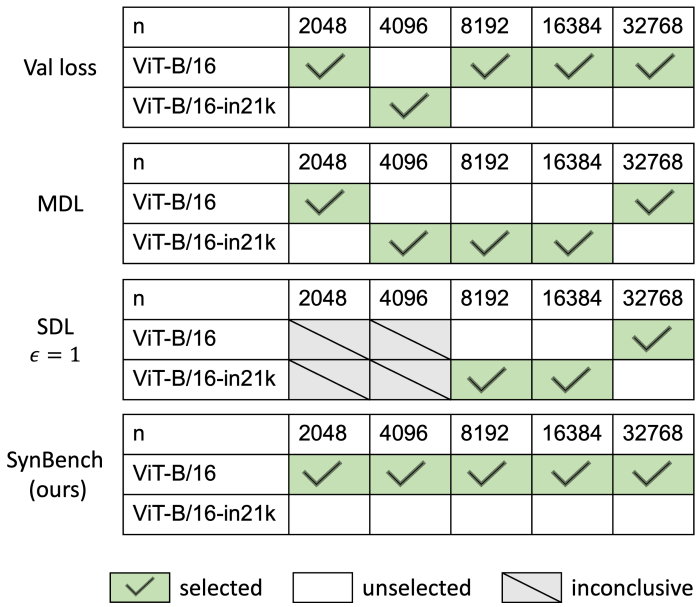


Figure 5-5: Comparison of model selections using task-agnostic benchmarks. We denote the model predicted to have better performance by "selected". Only SynBench gives consistent selections across varying data sample sizes. Refer to Appendix Table B.6 for more details.

correlation coefficients between the average accuracy in downstream tasks to scores given by Val loss, MDL, SDL, ϵSC, LogME, SFDA, and SynBench (SDL and ϵSC are excluded from the figure since they fail to give concrete numbers for small dataset sizes). With 2k synthetic samples, SynBench gives 0.79, whereas Val loss, MDL, LogME, and SFDA range between 0.46 and 0.55; with 8k synthetic samples, SynBench gives 0.89, whereas Val loss, MDL, LogME, and SFDA range between 0.65 and 0.81, surpassing the correlation by vanilla out-of-distribution accuracy (ImageNet-c's 0.64 and ImageNet-a's 0.57); with over 30k synthetic samples, Val loss, MDL, and SynBench all indicate very strong correlation (> 0.9) with real-life data accuracy, confirming the feasibility of probing pretrained representations in a task-agnostic yet effective way. To validate the capability of SynBench in informing model robustness, we further conduct CW attack [14], on CIFAR10 test set and calculate its correlation with SynBench. With 2k, 8k, and 30k synthetic samples, SynBench is also able to demonstrate moderate correlation with coefficient ranging from 0.74 to 0.84.

SynBench gives more consistent suggestions than baselines. We run a

Table 5.3: TinyImagenet standard and robust accuracy (%) changes (δSA and δRA) using ϵ-robust linear probing (ϵ-robust prob.). We see that ϵ-robust prob. with $\epsilon = \arg\max_\epsilon$SynBench-Score gives the best robust accuracy.

Models		TinyImagenet			
		$\epsilon = 0$	$\epsilon = 0.1$	$\epsilon = 0.2$	$\epsilon = 0.3$
ViT-Ti/16	SynBench-Score(ϵ)	**0.01**	**0.01**	0	0
	ϵ-robust prob. δSA	0	+0.3	-1.5	-1.9
	ϵ-robust prob. δRA	0	+1.1	+0.4	+2.2
ViT-B/16	SynBench-Score(ϵ)	0.33	0.36	**0.37**	0.35
	ϵ-robust prob. δSA	0	0	+0.7	+0.6
	ϵ-robust prob. δRA	0	-1.0	+2.5	+2.4
ViT-B/16-in21k	SynBench-Score(ϵ)	0.20	0.22	**0.23**	0.21
	ϵ-robust prob. δSA	0	+0.3	+0.3	+0.2
	ϵ-robust prob. δRA	0	+1.3	+2.0	+2.0
ViT-L/16	SynBench-Score(ϵ)	0.26	0.30	**0.33**	0.32
	ϵ-robust prob. δSA	0	-0.1	-0.2	-0.3
	ϵ-robust prob. δRA	0	+4.2	+6.6	+0.7

finer grid on the dataset size $n \in \{2048, 4096, 8192, 16384, 32768\}$ and compare the consistency of each metrics. Since LogME and SFDA showed worse correlation in the previous experiment, we exclude the two and only report the results on Val loss, MDL, and SynBench. We also include SDL to highlight its struggle with small sample size. In Figure 5-5, we give an example of the model selections between ViT-B/16 and ViT-B/16-in21k. Detailed numbers are reported in Appendix Table B.6. It is worth noting that SynBench consistently recommends ViT-B/16 over ViT-B/16-in21k, while other methods change with n. Besides better correlation and consistency, the runtime analysis also confirms 50× speedup over baselines using SynBench.

5.3.3 SynBench-guided ϵ-robust linear probing

When performing linear probing on downstream datasets, one can implement ϵ-robust linear probing [49] for better robustness. Concretely, let θ be the pretrained representation network and θ_c be the probing layer parameters, ϵ-robust linear probing solves $\min_{\theta_c} \max_{\delta:\|\delta\|_2\leq\epsilon} \mathbb{E}_{(x,y)\in\mathcal{D}} \ell_{\text{Cross-entropy}}(f_{\theta_c} \circ f_\theta(x+\delta), y)$. Here, we will show that the SynBench-guided ϵ-robust linear probing provides better insight into robustness-accuracy trade-off.

In Table 5.2, we only give SynBench-Scores with $\epsilon = 0$. We refer readers to Appendix Table B.4 for the full table with different ϵ. We cite 4 pretrained representations'

Table 5.4: Task-specific linear probing standard accuracy and robust accuracy (%).

Models	CIFAR10		SVHN		TinyImageNet	
	SA	RA	SA	RA	SA	RA
ViT-Ti/16	81.9	1.1	48.0	0.7	42.93	3.36
ViT-B/16	95.0	32.1	65.4	5.2	74.65	33.67
ViT-L/16	98.0	57.0	68.9	8.4	86.58	55.0

Table 5.5: Distances from synthetic data to CIFAR10, SVHN, and TinyImageNet.

Dataset	Distance	Gaussian-I	Gaussian-H
CIFAR10	FID	438	399
	MD	86142	67508
SVHN	FID	406	370
	MD	71527	57604
TinyImageNet	FID	403	361
	MD	76706	59979

SynBench-Score in Table 5.3 and observe that, for each model, SynBench-score is not necessarily monotonic in ϵ (peaks are boldfaced). For example, the SynBench-Score for ViT-B/16 peaks at $\epsilon = 0.2$, which indicates standard linear probing (*i.e.*, $\epsilon = 0$) may not be the most effective way to probe pretrained representations in terms of robustness-accuracy performance. This interesting indication is consistent with recent findings [49].

We hereby implement ϵ-robust linear probing and verify that $\epsilon = \arg\max_{\epsilon}$SynBench-Score can indeed find the best robustness-accuracy trade-off according to Table 5.3. For instance, SynBench-Score peaks at $\epsilon = 0.2$ for ViT-B/16 and correspondingly 0.2-robust linear probing on ViT-B/16 representations improves TinyImagenet standard and robust accuracy by the most (+0.7% and +2.5%). We defer CIFAR10 results to the Appendix Table B.8. The robust accuracy herein is obtained by AutoAttack [34].

5.3.4 The effect of data prior

In Section 5.2.4, it is stated that a more precise capture of the pretrained representation performance can be given if one has some prior knowledge of the downstream data distribution. In this section, we show this point by studying three specific downstream tasks, CIFAR10, SVHN, and TinyImageNet classifications, and give an example of the devised covariance matrix for SynBench synthetic Gaussians. In Table 5.4, we give

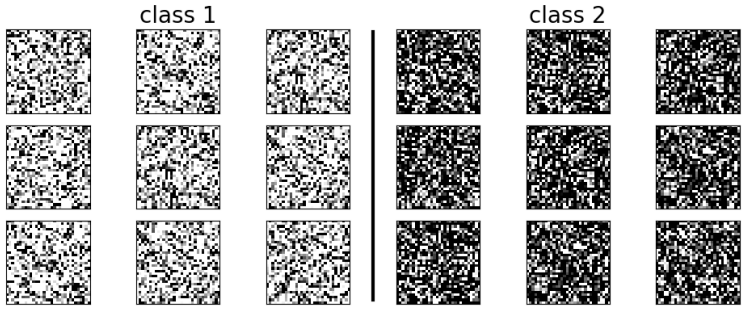
Table 5.6: SynBench-Scores on synthetic data with heptadiagonal covariance (Gaussian-H).

Models	$\epsilon = 0$	$\epsilon = 0.2$	$\epsilon = 0.4$	$\epsilon = 0.6$	$\epsilon = 0.8$
ViT-Ti/16	0	0	0	0	0
ViT-B/16	0.18	0.24	0.20	0.10	0.01
ViT-L/16	0.18	0.28	0.28	0.23	0.12

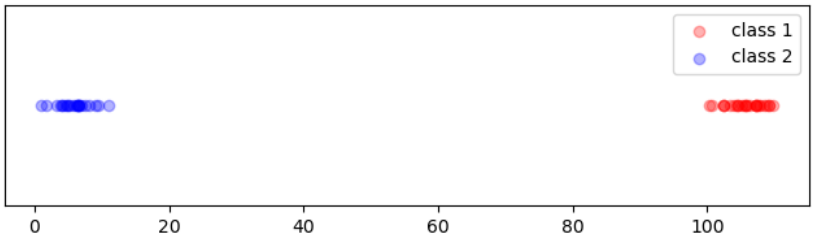
the standard and robust accuracy on CIFAR10, SVHN, and TinyImageNet (robust accuracy obtained by AutoAttack). Comparing the rows "ViT-B/16" and "ViT-L/16", it is observed that ViT-L/16 is in fact performing better than ViT-B/16 on these three downstream tasks, whereas SynBench-Score with identity covariance suggests the opposite (*cf.* Table 5.2). To uncover the reason behind the inconsistency, we calculate the distance between the synthetic Gaussian used throughout the experiments till now (dubbed Gaussian-I) and these datasets in Table 5.5. Recall that Gaussian-I, $P_{\mu_1,\mu_2,\Sigma}$, has $\mu_1 = -\mu_2 = s_i \cdot 1_d/\sqrt{d}$ and $\Sigma = I_d$. An easy modification on the covariance matrix Σ leads us to Gaussian-H, $P_{\mu_1,\mu_2,\Sigma}$ with $\mu_1 = -\mu_2 = s_i \cdot 1_d/\sqrt{d}$ and Σ be a channel-wise band matrix covariance. Gaussian-H captures the case when the R,G,B channel entries are externally independent (hence overall a block-diagonal covariance matrix with each of the 3 blocks being $224^2 \times 224^2$), and internally correlated based on locality (each block is a heptadiagonal matrix where only the main diagonal, and the first three diagonals above and below it have nonzero entries). Note that Gaussian-H is closer to the three datasets compared to Gaussian-I with respect to Fréchet inception distance (FID) [73] and Mahalanobis distance (MD) [116] according to Table 5.5. Based on Gaussian-H, SynBench now recommends ViT-L/16 over ViT-B/16 according to Table 5.6. We defer more results with Gaussian-H covariate synthetic data to Appendix Table B.9-B.11. This result shows that SynBench can incorporate complex data structures and downstream data characteristics into the process of synthetic data generation.

5.3.5 Synthetic data generation and separability

The synthetic data can be generated pixel by pixel if the covariance matrix is a diagonal matrix. In the case when the covariance is not a diagonal, we need to draw the whole image at once from the multivariate normal with generic covariance matrix.



(a) Synthetic data samples



(b) Projections of samples on $\mu_1 - \mu_2$

Figure 5-6: 18 synthetic data samples and their projections on the direction $\mu_1 - \mu_2$.

We include 18 synthetic data samples in Figure 5-6(a), showing 9 samples for each of the two classes. These examples are drawn from class-conditional Gaussians with scale $s = 25$ (*cf.* Section 5.2.3) and of size 32×32. Class-1 samples are on the left, and Class-2 samples are on the right. We can see that Class-1 samples are generally brighter than Class-2 samples. This is because Class-1 samples are drawn from the Gaussian with larger mean in the magnitude.

Furthermore, we demonstrate the separability of two class samples by projecting samples down along the direction of two Gaussian mean difference, in order to showcase their hidden discriminate patterns. That is, for vectorized sample x, Gaussian mean μ_1 and μ_2, we do the calculation $x^T(\mu_1 - \mu_2)$ and plot them on a line in Figure 5-6(b). From the plot, one can see that the samples from the two classes can be separated

easily.

5.3.6 Correlation breakdowns and robustness to out-of-distribution and challenging tasks

In this analysis, we calculate how SynBench score correlates with downstream performance per data set in the following Table 5.7.

Table 5.7: The correlation between SynBench-score and individual downstream task, and the Frechet Inception Distance (FID) scores from ImageNet21k to individual downstream task.

Datasets	Food101	CIFAR10	CIFAR100	birdsnap	SUN397	StanfordCars	Aircraft
FID to ImageNet21k	100.81	115.47	96.22	102.39	54.78	154.81	206.47
SynBench	**0.01**	-0.30	-0.50	-0.33	-0.32	**0.90**	0.87
Val loss	-0.31	0.07	0.24	0.03	0.03	-0.82	-0.70
MDL	-0.18	**0.19**	**0.37**	**0.17**	**0.16**	-0.84	-0.77
LogME	-0.48	-0.70	-0.83	-0.74	-0.74	0.85	**0.95**
SFDA	-0.41	-0.66	-0.77	-0.67	-0.69	0.88	**0.95**
Datasets	VOC2007	DTD	Pets	Caltech101	Flowers	MNIST	FER2013
FID to ImageNet21k	52.30	98.37	104.15	53.51	112.64	301.28	175.75
SynBench	**0.64**	0.86	**0.40**	**0.09**	-0.64	0.56	**0.81**
Val loss	-0.80	-0.66	-0.63	0.02	0.37	-0.33	-0.85
MDL	-0.76	-0.75	-0.54	-0.01	**0.49**	-0.41	-0.82
LogME	0.22	**0.98**	-0.13	-0.01	-0.92	**0.85**	0.55
SFDA	0.24	0.96	-0.07	-0.07	-0.87	0.84	0.60
Datasets	STL10	EuroSAT	RESISC45	GTSRB	KITTI	Country211	PCAM
FID to ImageNet21k	71.19	142.62	104.80	156.81	163.92	36.72	235.63
SynBench	-0.40	0.77	0.91	0.59	0.40	**0.96**	**0.90**
Val loss	0.11	-0.54	-0.76	-0.34	-0.14	-0.96	-0.99
MDL	**0.23**	-0.64	-0.82	-0.43	-0.25	-0.97	-0.96
LogME	-0.80	**0.97**	**0.96**	**0.85**	**0.81**	0.69	0.59
SFDA	-0.75	0.93	**0.96**	0.82	0.77	0.70	0.64
Datasets	UCF101	Kinetics700	CLEVR	HatefulMemes	SST	ImageNet	AVG acc.
FID to ImageNet21k	79.40	time out	194.64	86.64	368.13	17.78	
SynBench	**0.81**	**0.64**	0.72	-0.59	0.35	**0.30**	**0.92**
Val loss	-0.93	-0.82	-0.48	0.34	-0.22	-0.56	-0.92
MDL	-0.87	-0.74	-0.59	**0.47**	-0.32	-0.45	-0.91
LogME	0.45	0.17	**0.97**	-0.88	**0.41**	-0.22	0.72
SFDA	0.51	0.24	0.94	-0.83	0.34	-0.15	0.77

Subset of OOD tasks We analyze SynBench score's correlation to the subset of OOD tasks. In the following Table 5.7, we computed the Frechet Inception Distance (FID) scores from ImageNet21k to the downstream tasks, and used them as the indicator of how OOD are the tasks. We then computed SynBench-score correlation with tasks that have FID scores larger than a threshold {50,100,150,200}. We do want to note that not all models in our analysis are pretrained with ImageNet21k; however,

Table 5.8: The correlation between SynBench-score and the average accuracy of FID-thresholded downstream tasks.

FID	> 0 (all tasks)	> 50	>100	>150	> 200
SynBench Correlation	0.92	0.93	0.93	0.82	0.92

since ImageNet21k has become a go-to pretraining dataset, we assume samples therein are in-distribution.

From Table 5.8, we see that if we don't apply filter on FID (or equivelantly let threshold be 0), the initial correlation was 0.92. As we gradually increase the threshold to 50, 100, 150, and even 200, the correlation stays above 0.8, indeed suggesting SynBench's robustness to OOD tasks.

Subset of more challenging tasks We futher analyze SynBench score's correlation to the subset of more challenging tasks. When we check how SynBench can serve as a performance metric of pretrained models, we used the average accuracy of 27 downstream tasks as the proxy of the general performance. Among the 27 tasks, there are indeed datasets that are large and complex, inclduing ImageNet. In the following Table 5.9, we highlight 3 subsets of tasks that represent more challenging datsets in different dimensions (number of classes, data types, task types).

1. For datasets that have more than 100 classes (Food101, Birdsnap, SUN397, StanfordCars, Aircraft, Caltech101, Flowers, Country211, UCF101, Kinetics700, ImageNet), SynBench-score correlates with their average performance with correlation of 0.56, compared with the best baseline (SFDA) of 0.19.

2. For video datasets (UCF101 and Kinetics 700), SynBench-score correlates with their average performance with correlation of 0.72, compared with the best baseline (SFDA) of 0.36.

3. For the visual reasoning and question-answering dataset, CLEVR,, SynBench-score correlates with its performance with correlation of 0.72, while LogME and SFDA demonstrate even stronger correlation (> 0.9).

Overall, SynBench shows robust performance across these break-down groups.

Table 5.9: The correlation between SynBench-score and subsets of downstream tasks.

Large/complex datasets	datasets w/ #classes>100	video datasets (UCF101 and Kinetics 700)	visual reasoning/QA dataset	dataset average
SynBench	**0.56**	**0.72**	0.72	**0.80**
Val loss	-0.75	-0.88	-0.48	-0.91
MDL	-0.66	-0.81	-0.59	-0.85
LogME	0.11	0.30	**0.97**	0.45
SFDA	0.19	0.36	0.94	0.51

Table 5.10: The average ranking of correlations with downstream tasks SynBench and other baselines.

Correlation	ranking
SynBench	**2.11± 0.976**
Val loss	3.68± 1.166
MDL	3.57± 1.613
LogME	3.00± 1.488
SFDA	2.64± 1.076

Correlation ranking Now, some might wonder, why SynBench negatively-correlated with parts of datasets in Table 5.7. To answer this, we hint that if there exist a metric that highly correlates with the linear probing performance on every single downstream task, it would imply that the linear probing performance on every single downstream task also correlates highly with each other— which is not the case in reality. Therefore, we are seeking a metric that can inform on the potential overall performance. In Table 5.10, we provide the average ranking of correlations with downstream tasks by SynBench and other baselines as a more robust and intuitive measure. It is clear that SynBench is able to give the overall best correlation with each individual downstream.

5.4 Discussions

5.4.1 Usage

We view SynBench as a "necessary" and "minimum" model test in the sense that, with perfect data sampled from an ideal distribution, any undesirable deteriorated behavior (such as weakened robustness) reveals the weaknesses of the representation model that could possibly lead to vulnerabilities in real-life downstream tasks. Therefore, in designing this minimum test, it is important that the task has a theoretical ideal

(and optimal) solution (*i.e.* the trade-off preserved by class conditional Gaussians, Theorem 9 iv).

Here are some possible scenarios to use our developed tool:

- model auditing: use SynBench to generate diverse psuedo tasks (e.g., with diffrent difficulty levels) and compare them with theoretically optimial results, for a comprehensive evaluation on the capability of a pre-trained model

- hyperparameter tuning: as shown in Sec. 5.3.3, SynBench can be used for hyperparameter selection in robust linear probing, which leads to improved performance in the considered downstream tasks.

- model selection (without using downstream data): without the knowledge of downstream applications, one can use SynBench to rank the quality of pre-trained representations (e.g., the example shown in Figure 5-4). It is also possible to incorporate some known statistics of the downstream dataset into guided synthetic data generaltion and evaluation in SynBench, as discussed in Sec. 5.3.4.

- model training: while updating a model in the pre-training state, one can use SynBench to ensure the model performance (in terms of SynBench-Score) is aligned.

5.4.2 Gaussian models

Besides the fact that Gaussian models make great well-posed problems for pretrained models, the idea of evaluating foundation models on synthetic Gaussian datasets also stems from two observations previously made in the literature. (1) [223] showed that simple Gaussian Mixture Models (GMMs) learned from pixels of natural image patches can successfully be used to model the statistics of natural images, which include contrast, textures at different scales and orientations, and boundaries of objects in the reference. Specifically, since our target is pretrained vision models, the capabilities of perceiving contrasts and edges etc are centric. Besides image patches,

there are some discussions in the literature about how images patches connects to whole images [222, 79]. (2) Nevertheless, general GMMs do not yield themselves for analytic derivation of accuracy-robustness trade-offs. Luckily, some recent works on Gaussian universality [135, Theorem C.1, Fig 6] have showed that for the overparameterized setting general linear models for GMMs and Gaussians both show similar training and generalization errors even when the underlying labels are strongly correlated with the data structure. Moreover, Gaussian models can be readily used to analytically derive expression for efficiently measuring accuracy-robustness trade-offs. Although the models used in real life came from richer model classes, foundation models do lie strongly in the overparameterized regime. We design our testing framework using similar Gaussian models and test their effectiveness empirically in understanding performance of foundation models on downstream tasks.

5.4.3 Pretrain data versus synthetic data

Conducting evaluation with pre-train data can be infeasible/inappropriate due to three reasons. First of all, with the increasing use of self-supervision during the pretraining, the pre-train data can be unlabeled. Secondly, even in the case when the application scenerio is model training and the pre-train data is labeled, the evaluation scores based on the pre-train data can be inconclusive if the evaluation data are biased or under-representative (e.g. pretrained models tend to overfit to the pre-train data). Lastly, from the perspective of the model auditing, the data used for model pretraining can simply be private or inaccessible (e.g., Web-scale raw data).

In these scenarios, one can use SynBench to generate diverse pseudo tasks and non-private synthetic data for conducting comprehensive evaluation of a pre-trained model. By comparing to an idealized data distribution and the corresponding theoretically-optimial reference, SynBench-Score (as illustrated in Figure 5-1) can quantify the quality of representations, in the sense that the area under the curve (AUC) ratio closer to 1 means better representations.

5.4.4 Limitations

Linear probing. SynBench analysis focuses on linear probing performance, which is a popular, low-complexity evaluation protocol widely used in the community [23, 69], especially for large neural networks (foundation models). Other assessment tools of pretrained models, such as LogME [210], is also evaluated by the correlation coefficient between their metric and linear probing accuracy. For tasks other than classification, we do observe in some literature that SynBench-Score might still be informative, e.g. ViT-L/16 is reportedly performing worse than ViT-B/16 with MLA decoder in a food segmentation task from [202], DINO ViT-B performs better than DINO ViT-S in DAVIS 2017 Video object segmentation, and DINO ViT-S/16 performs better than DINO ViT-S/8 according to Jaccard similarity on PASCAL VOC12 dataset from [19]. For fine-tuned pretrain representations, ViT-L/16 loses to ViT-B/16 on finetuned medical tasks with, e.g., X-ray images [128, Table 4-8], and magnetic resonance imaging [182, Table 2-3]. Although we are unable to fully justify the relationship between SynBench-Score and non-classification tasks, we believe that if non-classification tasks such as object detection/regression can be translated into classification tasks, SynBench can be extended to those tasks.

Gaussian models. "Can we trust the data representations from a pretrained image model, if it fails to have reasonable performance on simple synthetic datasets?" This is the motivation for our work. When designing the task-agnostic and data-free framework, we narrow our scope for a more "well-posed" problem, by using an idealized data distribution with tractable separability, lifting the need for real-life data. This enables interesting application scenerio such as model auditing, selection, training, and alignment. Therefore, ideologically, SynBench allows any idealized data distribution, provided that the optimal performance (e.g. accuracy-robustness as in our case) can be characterized. At the current stage, the practicality of SynBench owes to the idealized Gaussian distribution, whose optimal robust Bayes classifier is known.

Synthetic tests. Since SynBench is a task-agnostic and data-free framework, it relies on synthetic data drawn from idealized data distribution with optimal performance. Albeit these synthetic data may inevitably miss intricate details of downstream tasks and data, this framework still provides an easy first check in representation quality.

5.5 Conclusion

In this chapter, we explored how to extend the well-studied Gaussian data modeling techniques to systematically study the representation quality of pretrained image models, by proposing a **task-agnostic** and data-free framework, *SynBench*. With our synthetic Gaussian analysis, the robustness-accuracy relationship becomes tractable and naturally yields a theoretically-derived robustness-accuracy trade-off, which serves as the reference for pretrained representations. We validated the usefulness of SynBench on several pretrained image models in giving insightful comparisons of model attributes. We demonstrated its high correlation with real-life tasks and showed its consistent model selections. We envision the SynBench framework to be further extended to other trustworthiness dimensions (e.g., privacy and fairness) and other domains, to shed light on task-agnostic benchmarking designs that are simple and synthetic.

Chapter 6

Evaluating robustness-accuracy of large language models using synthetic data

6.1 Introduction

In recent years, language models (LMs) have emerged, showcasing remarkable capabilities across a wide range of natural language processing (NLP) applications [136, 38, 208, 143, 142, 176, 76]. While new opportunities present themselves with foundation models, they also bring forth potential risks and challenges [12, 9, 174, 6]. For example, despite the unprecedented publicity of LMs and beliefs in their emergent abilities [193], some also argued the emergent abilities of LMs are a mirage [159] and a change in metric choice can lead to a different conclusion. Recently, researchers have also expressed concerns about the potential for LMs to be trained on test sets [108, 58, 131]. Even worse, private or held-out unpublished test sets may as well be vulnerable to data leakage through querying the LMs via APIs for evaluation purposes. Extraction attacks [15, 16], membership inference attacks [74, 175, 120], and generative embedding inversion attack [105], caused by unintended memorization [15, 163] further deepened our concerns about test set contamination.

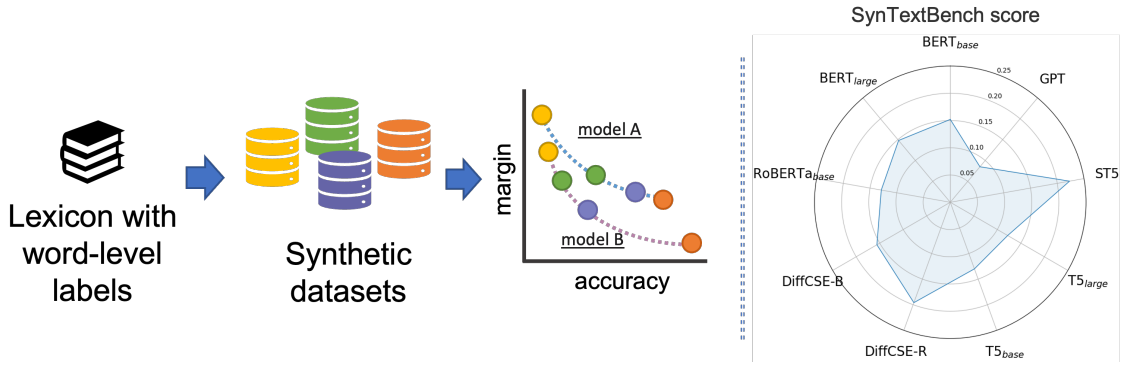


Figure 6-1: Overview of SynTextBench. SynTextBench generates a set of synthetic datasets from any given lexicon with word-level labels. We test the given LM on sentence-level tasks with these datasets and obtain robustness-accuracy characterization under a range of steerable task difficulties. For each LM, we can plot the robustness-accuracy trade-off curve and make model comparisons.

To address the caveat of test set contamination, in this chapter, we aim to propose a new testbed for evaluating LMs with synthetic data. We link the design of the synthetic test set to two fundamental skills infants must master during language acquisition: identifying words and understanding linguistic structures [52]. One intuitive approach is to generate labeled synthetic sentences using an existing generative LM and then evaluate LMs with the constructed test sets. By this, the generated sentences would harness language structural heuristics learned by the LM, and a decent probing result also requires the ability to distinguish words and their associated meanings, such as semantics. However, this workflow does not permit the active manipulation of synthetic task difficulties due to the limited level of interpretability [216] and intrinsic bias of specific LMs [1]. Motivated by the limitation, we explore an alternative route by entirely eliminating the reliance on LMs for test set generation. Specifically, we leverage existing sentiment lexicons, such as SentiWordNet 3.0 [3], to generate working word lists based on the word (or synset) level labels. We build positive, negative, and neutral word lists from the lexicon, and construct sentences following the nesting parentheses [132], which mimics the recursion structural hypothesis about the narrow language faculty in humans [67] and the dependency tree structure in natural language [28]. By maneuvering the mixing percentage of binary words (positive/negative words) and

neutral words, we create a configurable testbed for evaluating the performance of LMs on different levels of difficulty and complexity. Finally, we benchmark and quantify the ability of each LM on sentence classification tasks by comparing their performance on a set of our synthetic datasets with varying difficulty levels.

We dub our evaluation framework using synthetic data by *SynTextBench* and present the workflow in Figure 6-1, where we focus on benchmarking LM sentence embeddings in terms of their accuracy and robustness. By accuracy, we are interested in analyzing the linear separability of sentence representations rendered by different pretrained LMs. We note that in learning sentence embeddings, the go-to metrics are cosine distance or linear probing accuracy, both of which imply separability. By robustness, we refer to the decision margin on these sentence embeddings with respect to the optimal classification strategy. We derive both measures using only the constructed synthetic datasets, which allow for contamination-free benchmarking of LMs. SynTextBench is designed as an extendable framework for the evaluation of language sentence representations that covers a range of controllable task difficulties.

6.1.1 Our contributions

- We introduce *SynTextBench*, a novel theoretically-grounded framework to generate steerable synthetic datasets towards a holistic evaluation of LMs. The use of synthetic datasets alleviates the risk of test-data leakage and offers new tools for LM testing and auditing.

- SynTextBench provides a configurable lightweight testbed and a quantifiable metric for evaluating the robustness and accuracy of LMs on different levels of difficulty and complexity for sentence classification tasks, with no restrictions on the model architecture.

- We conduct experiments with several state-of-the-art LMs on our testbed and report their performance and behavior. SynTextBench, as a real-data-free evaluation method, shows high correlation with robustness-accuracy performance evaluated on real data. Further study demonstrates its capability of making

quick attribution comparisons such as analyzing fine-tuning effects for LMs.

6.1.2 Related works

In evaluating the performance of LMs, the current de facto evaluation paradigm is to utilize widely-used NLP benchmarks such as the General Language Understanding Evaluation (GLUE [188]/SuperGLUE [187]) benchmark, the Stanford Question Answering Dataset (SQuAD v1.1 [147]/v2.0 [146]), the Situations With Adversarial Generations (SWAG [212]) dataset, the ReAding Comprehension from Examinations (RACE [97]) dataset, the Evaluation Toolkit for Universal Sentence Representations (SentEval [32]), BIG-Bench [167], etc. In many cases, these NLP benchmarks are supersets of datasets, e.g., GLUE is a collection of 9 datasets for evaluating natural language understanding systems, and SentEval is a collection of 7 Semantic Textual Similarity (STS) tasks and 7 transfer datasets that have partial overlap with GLUE. The heavy reliance on real-world tasks can be exemplified by broad literature. For example, Bert [38] was evaluated on GLUE, SQuAD v1.1/2.0, SWAG; Roberta [112] was evaluated on GLUE, SQuAD v1.1/2.0, RACE; and T5 [143] was evaluated on GLUE/SuperGLUE, SQuAD, CNN/Daily Mail abstractive summarization and WMT translation. HELM [109] proposes a holistic evaluation framework for LMs that measures 7 metrics on 42 scenarios. However, when confronting the challenge of test-data leakage, to the best of our knowledge, there is no real-data-free evaluation method for NLP pretrained representations. In Chapter 5, we reported the validation loss (Val loss), minimum description length (MDL) [8, 185], surplus description length (SDL) and ϵ-sample complexity (ϵSC) [196] on class-conditional Gaussian distribution data as an effort to build task-agnostic evaluation baselines for pretrained representations in computer vision. This chapter differs from the previous chapter in that we focus on the domain of natural language processing and we do not assume the data inputs are sampled from an idealized distribution. Instead, we create synthetic sentences and proxy tasks based on a lexical resource for LM evaluation.

6.2 Methodology

6.2.1 Why using synthetic datasets for LM evaluation?

To reduce the reliance on real-world data, we propose to build synthetic NLP tasks by generating synthetic sentences as model inputs at test time. This way, we no longer need to exchange sensitive private data or label-annotated data as test sets with LM APIs. In making a steerable and transparent evaluation framework for LMs, we first detail the desiderata of proxy tasks and the evaluation metric.

- Task substance: Tasks should test a pretrained LM's ability to encode sentence representations that preserve class separability when evaluated by a linear classifier.
- Task difficulty: Tasks' difficulty should be configurable to allow for comprehensive analysis, *i.e.*, one can generate tasks of various levels of difficulty.
- Task feasibility: Tasks should be feasible to solve, *i.e.*, the sentences should be distinguishable to a certain degree by an algorithm that works on the raw sentences input.
- Task independence: Tasks should be independent of the LM to be evaluated, in order to avoid biased evaluation, *i.e.*, neither sentences nor labels should be given by an LM.
- Task equity: Tasks should be able to be generated by anyone and affordable for anyone without requiring any private data or favoring any party with more resources.
- Metric informativeness: The designed framework should give a quantifiable metric that has a clear implication (e.g., the larger the better) and correlates well with the real performance.

With these in mind, it is straightforward to see why we should not opt for synthetic datasets generated by any LM: (1) task difficulty would not be configurable (see more

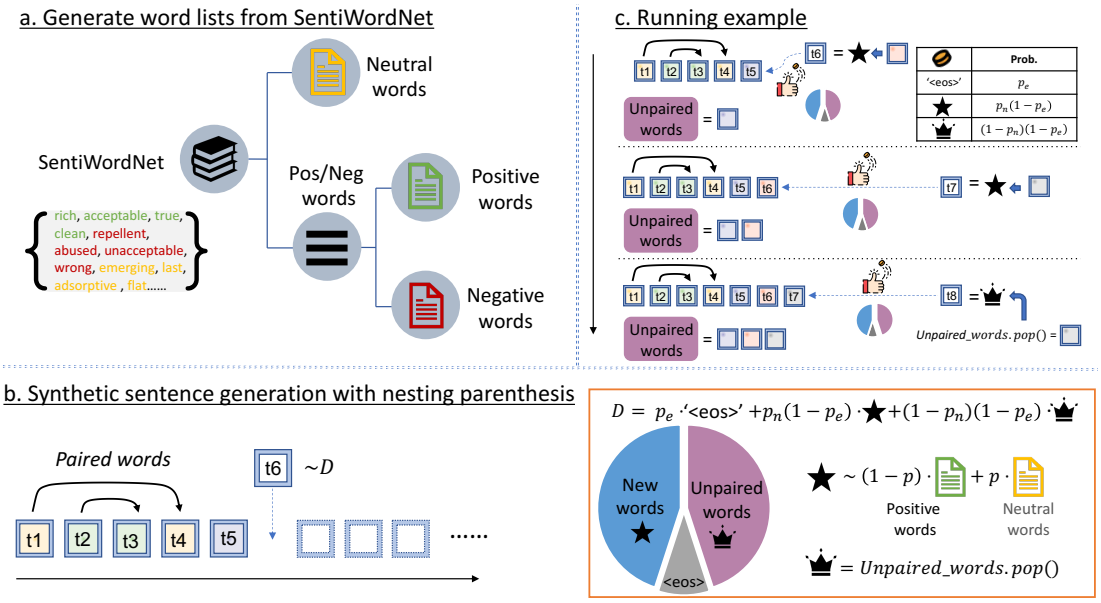


Figure 6-2: Overview of the sentence generation procedure. In block a, we generate word lists from SentiWordNet 3.0. In block b, we generate each sentence token following nesting parentheses and mixing distribution D. In block c, we show a running example of sequentially generating t_6, t_7, t_8.

evidence in Section 6.4.1), (2) the evaluation might be biased and favor the LM that generates the synthetic sentences or labels due to the intrinsic bias of each LM, and (3) any auditor without access to proprietary LMs or datasets cannot run independent evaluation.

In the following, we explain how we leverage sentiment lexicons, such as SentiWordNet 3.0, to create building blocks for our framework. Then, we put together building blocks and generate synthetic inputs to LMs by observing a nesting structure. We adjust the mixing ratio of ingredients in the recipe to simulate tasks of different difficulties. We depict this procedure in Figure 6-2. Finally, we will introduce our evaluation workflow and how we arrive at a quantifiable metric.

6.2.2 Constructing synthetic datasets and tasks

Word List. Building a synthetic task requires us to define the synthetic inputs to be used. Here, we utilize sentiment lexicons with word-level labeling. SentiWordNet labels the synsets of WORDNET [119] according to the notions of "positivity", "negativity",

Table 6.1: Examples of synsets in SentiWordNet 3.0.

SynsetTerms	PosScore	NegScore	SynsetTerms	PosScore	NegScore
able#1	0.125	0	unable#1	0	0.75
acroscopic#1	0	0	unquestioning#2	0.5	0.5
living#3	0.5	0.125	concrete#1	0.625	0.25
accurate#1	0.5	0	straight#5	0	0
unfaithful#4	0	0.5	active#5	0.5	0.125

and "neutrality". Each of the entries in SentiWordNet has PosScore and NegScore denoting the positivity and negativity score, and ObjScore is calculated by 1 - (PosScore + NegScore), denoting the neutrality score. When categorizing these words, we remove the sense number associated with the words and group words into individual word list based on the following criteria: for a word w,

- if PosScore $>$ NegScore, we categorize w into the positive word list;
- if PosScore $<$ NegScore, we categorize w into the negative word list;
- if PosScore $=$ NegScore $= 0$, we categorize w into the neutral word list.

We give running examples in the following for better understanding: We drop columns POS, ID, GLOSS in the examples for easier illustration. By performing the procedure on synsets in Table 6.1, we obtain a positive word list {able, living, accurate, concrete, active}, a negative word list {unfaithful, unable}, a neutral word list {acroscopic, straight}.

In practice, we perform the procedure on SentiWordNet 3.0 and gather a positive word list with 23147 words, a negative word list with 26440 words, and a neutral word list with 154993 words. The same procedures can be applied to any sentiment lexicons with word-level labeling, which will result in different word lists. To this end, we created the word lists from SentiWordNet 3.0 as depicted in Figure 6-2(a).

Sentence structure. A recent literature [132] explored the power of music and Java code in training models that transfer to NLP tasks. It further stated that, not only music and Jave code, non-linguistic artificial parentheses languages can also train LMs that yield substantial gains compared to random data when testing on natural language [27, 153, 133]. Motivated by this, we follow one of the abstract structures, nesting parentheses, when generating the synthetic sentences in our proxy tasks. The

inclusion of the parenthesis is to guarantee we test for the linguistic structures, whose importance is repeatedly advocated in literature from both machine learning and cognitive science [52, 198, 117]. Specifically, the nesting parenthesis involves paired tokens and a recursive structure. For example, by referring to Figure 6-2(b), one sees that t_1 and t_4 are paired words, while t_2 and t_3 are another paired words. In our example, the words are hierarchically nested, meaning the token to be paired with t_2, which is t_3 in our case, should appear before the pairing token with t_1. In other words, it observes a "last in first out" data structure, and the arcs in Figure 6-2(b) do not cross.

Sentence generation and difficulty level. With the created word list from above, we will now explain how to do sentence generation following the structure introduced. Let us revisit the case in Figure 6-2(b). Assume we want to generate a positive sentence (label $y = 1$), and we already generated the first five tokens $t_1 : t_5$ in the sentence with colors denoting the picked word. Now, to decide the next token, we sample t_6 from a mixing distribution D, where

$$D = p_e \cdot \text{`<eos>'} + p_n(1 - p_e) \cdot \text{last\_unpaired\_word} + (1 - p_n)(1 - p_e) \cdot D_{\text{new}}. \quad (6.1)$$

To interpret distribution D, we realize that there are essentially 3 possible outcomes for the incoming t_6 token: (1) it can be the end of sentence indicator '<eos>', (2) it can be the popped token from the stack that stores the unpaired words, *i.e.*, the last unpaired word, (3) it can be a new word. If it is to pick a new word, this word will be sampled from the distribution of new words D_{new}, which directly depends on the label y of the sentence to be generated and the desired task difficulty. For a positive sentence ($y = 1$), $D_{\text{new}|y=1}$ is described by the probability density function (PDF) $p \cdot f_{\text{NEU}}(x) + (1 - p) \cdot f_{\text{POS}}(x)$, where p specifies the percentage of neutral words in a synthetic sentence, f_{NEU} gives the PDF of neutral words, and f_{POS} gives the PDF of positive words. Similarly, if we are to generate a negative sentence ($y = -1$), we have $D_{\text{new}|y=-1}$ described by $p \cdot f_{\text{NEU}}(x) + (1 - p) \cdot f_{\text{NEG}}(x)$, where f_{NEG} gives the PDF of negative words.

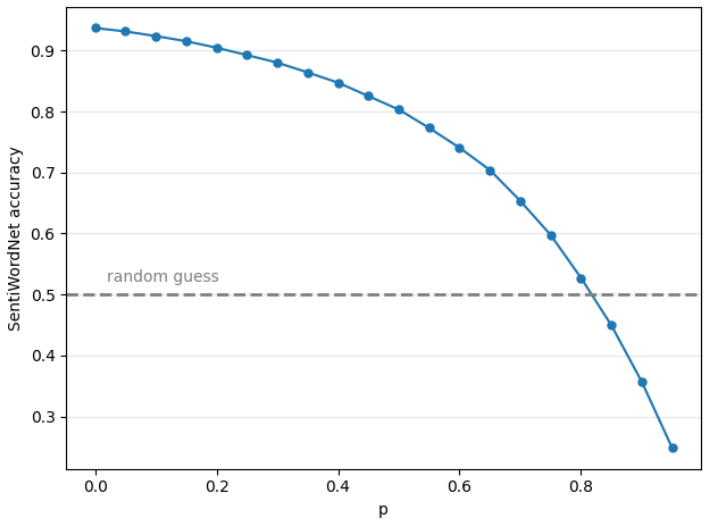


Figure 6-3: The reference accuracy given by SentiWordNet sentiment analysis. With an increasing mixing ratio p, the task becomes harder and the reference accuracy also shows a decreasing trend.

In Figure 6-2(c), we show a running example of the sentence generation process, where we flip a coin with 3 outcomes each time to decide on a new token. When the realization is "new words" (like in t_6 and t_7), this word will also be pushed to the stack "*Unpaired_words*" that stores unpaired words. When we are deciding t_8, we draw "unpaired words" and hence t_8 is determined by *Unpaired_words.pop()*. In essence, with the generated sentence, its label is determined by construction, which guarantees the **task independence** since the label is not given by an LM. It also allows configurable **task difficulty** by adjusting the percentage p of neutral words in a synthetic sentence. That is, it is easier to predict the sentiment of sentences consisting of 90% positive words and 10% neutral words than that of sentences constructed all by neutral words. On the whole, by fixing a mixing ratio p, together with the fixed p_e and p_n given in the above, one synthetic dataset will be constructed as well as a resulting proxy sentiment classification task. By varying the mixing ratio p, a set of tasks with diverse difficulties can be created. In Figure 6-3, we prove the **task feasibility** by demonstrating the separability of generated synthetic datasets by SentiWordNet sentiment analysis algorithm [37]. With an increasing mixing ratio p, while the task becomes harder, we show there at least exists an algorithm that can separate the data

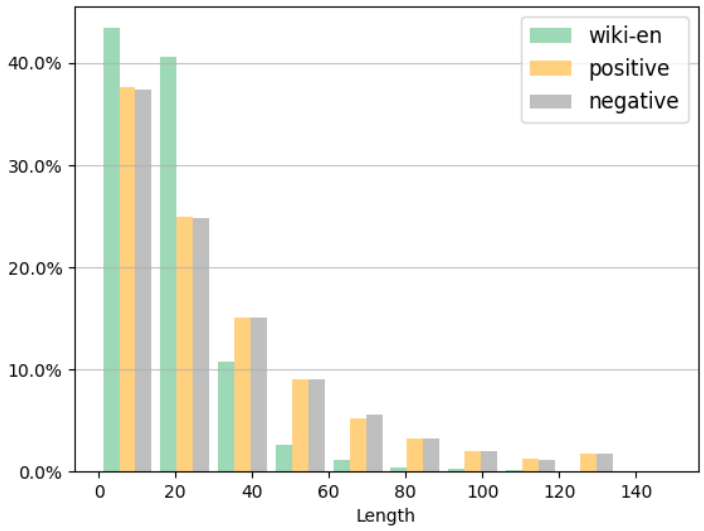


Figure 6-4: The histograms of sentence lengths in the English Wikipedia corpus (stop words removed) and the constructed synthetic corpus (positive/negative sentences).

to a certain degree, showcasing a lower bound on the optimal classification strategy. By our workflow of constructing synthetic datasets and tasks, we also guarantee **task equity** since the generation process requires no access to any LM or private data, and can be readily replicated by anyone with limited resources. Furthermore, we note that the construction of synthetic datasets and tasks described herein is also extendable to other lexicons and tasks by swapping the lexicon used for extracting word lists.

Lastly, we note that during the construction of synthetic sentences, the probability p_e associated with the special token '<eos>' is determined by its frequency in the English Wikipedia corpus. For the remaining mass $1 - p_e$, p_n portion is assigned to new words, with its value picked following [132], which is $p_n = 0.5$. Additionally, when there are no unpaired words in the stack (e.g., when drawing the starting token of the sentence, or when all the unpaired words are popped), we assign its probability $p_n(1-p_e)$ to new words. We show the length profile of our synthetic data in Figure 6-4.

Discussions. The inclusion of parentheses in our sentence structure guarantees we test for the linguistic structures but at the same time makes non-grammatical test sets. While grammar might be crucial in some NLP tasks that requires more advanced reasoning. For sentiment analysis, we believe it should not have a strong

dependency on grammar (we exclude the scenario of negation which can be detected by a rule-based method). For example, the reviews "love love fantastic", "love fantastic love" and their word permutations should all be predicted as positive, regardless of their grammar. We support this intuition by additional experiment where we noticed that 86% of the labels given by Huggingface sentiment analysis pipeline on product reviews classification [77] remain the same after removing 284 stop words (*cf.* Appendix B.4.3) from the sentences and hence making them non-grammatical. We leave more details and sentence examples to the discussions in Section 6.4.2.

6.2.3 Robustness-accuracy evaluation

Given an LM g, let x, y be the input sentence and its label, z be the sentence embeddings $z = g(x) \in \mathbb{R}^n$, we are interested in evaluating the accuracy of the sentence embedding classifiers f, and the average distance Δ from sentence embeddings to the linear classifiers (*i.e.*, decision margins). We let z_1 be $\{z : z = g(x), y = 1\}$ and z_{-1} be $\{z : z = g(x), y = -1\}$.

Preparing sentence embeddings. Recall that Bert-flow [104] and Bert-whitening [169] transformed the sentence embeddings into an isotropic Gaussian distribution to remedy the anisotropic behavior in the sentence embedding vector space. We thereby also perform whitening on sentence representations before we draw the decision rule on the embeddings. Transforming a set of sentence embeddings of a class into an isotropic Gaussian involves two steps: (1) model the mean b_y and covariance Σ_y of original embeddings z_y, (2) apply a transformation to the embeddings $F^T S^{-1/2} z_y$, where $FSF^T = \Sigma_y$ is the singular value decomposition of Σ_y. Nevertheless, since Σ_y can be ill-conditioned, directly applying $S^{-1/2}$ on embeddings z_y might amplify noisy signals due to numerical instability. Thus, we propose to reduce the dimension according to energy-preservation [102] (also called variance-based methods by [48]). We select to keep K dimensions according to $\arg\min_k \frac{\sum_{i=1}^k s_i}{\sum_{i=1}^n s_i} \geq 0.99$, where $s_i = \text{diag}(S)[i]$ is the i-th largest singular value of S. Till now, we see that the sentence embeddings are transformed to an $\mathbb{R}^K$ vector space via $F^T_{:,1:k} S^{-1/2}_{1:k,1:k} z_y$. We perform these operations for both classes ($y = 1$ and $y = -1$) separately. Since

we want the transformed embeddings to observe the original relative distance between two classes, we further scale the distance between two whitened Gaussians by $d_{\underline{\text{Inter}}\text{-class}}/d_{\underline{\text{Intra}}\text{-class}}$, where the numerator $d_{\underline{\text{Inter}}\text{-class}} = \|b_1 - b_{-1}\|$ calculates the inter-class distance (the distance between two class centers b_1 and b_{-1}), and the denominator $d_{\underline{\text{Intra}}\text{-class}} = \frac{1}{m_1+m_2}(\sum_{i=1}^{m_1} \|z_1^i - b_1\| + \sum_{j=1}^{m_2} \left\|z_{-1}^j - b_{-1}\right\|)$ calculates the intra-class distance (the average distance from class data to class mean) with m_1 and m_2 being the number of positive sentences and negative sentences, respectively. We let T_y denote the overall transformation operations and obtain transformed embeddings $\hat{z_1} = T_1(z_1)$ and $\hat{z_{-1}} = T_{-1}(z_{-1})$.

Decision margins induced by robust Bayes optimal classifiers. Recall that robust Bayes optimal classifiers explicitly give the optimal classification strategy for class-conditional Gaussian distribution in the presence of data perturbations [4, 36]. Here, we see that $(\hat{z}, y)$ are modeled as P_{μ_1,μ_2,I_K}: $\hat{z}|y = 1 \sim \mathcal{N}(\mu_1, I_K), \hat{z}|y = -1 \sim \mathcal{N}(\mu_2, I_K)$, and $y \in \mathcal{C} = \{+1, -1\}$. While finding the robust Bayes optimal classifier generally involves solving the optimization problem $\arg\min_{\|z\|_2 \leq \epsilon}(\mu - z)^T \Sigma^{-1}(\mu - z)$ (*cf.* Section 2.1.3), we can prove that, when the covariance is an identity matrix, the class priors $\mathbb{P}(y = 1) = \tau$, $\mathbb{P}(y = -1) = 1 - \tau$, the perturbation radius ϵ, then the optimal classifier is given as simply $f : \text{sign}(w^T(\hat{z} - \frac{\mu_1+\mu_2}{2}) - q/2)$, where $q = \log\{(1-\tau)/\tau\}$, $w = \tilde{\mu}(1 - \epsilon/\|\tilde{\mu}\|_2)$, and $\tilde{\mu} = \frac{\mu_1-\mu_2}{2}$. Furthermore, when the classes are balanced (*i.e.*, $\tau = 1/2$), the robust Bayes optimal classifier overlaps with the Bayes optimal classifier. That is, the (robust) Bayes optimal classifier is plainly $\text{sign}(\tilde{\mu}^T(\hat{z} - \frac{\mu_1+\mu_2}{2}))$, which is independent of ϵ. We then use this given classifier to calculate the accuracy on the synthetic datasets. In fact, we prove in Appendix A.4 that, as long as $\tilde{\mu}$ lies completely within a degenerate subspace of the eigenspace of the covariance matrix (*i.e.*, with eigenpairs $\{(\lambda_k, v_k), k \in [n]\}$, for $\forall\ i, j \in \{k : \lambda_k \neq 0, \tilde{\mu}^T v_k \neq 0\}$, $\lambda_i = \lambda_j = \lambda$), the ϵ-robust Bayes optimal classifiers overlap for all ϵ. In the case of an identity covariance matrix, the degenerated subspace of the eigenspace expands the whole $\mathbb{R}^K$, hence $\tilde{\mu}$ lies in the space naturally.

Now that we have specified the optimal robust classification rule on the transformed sentence embeddings, we write out the decision margin induced by the classifiers using

Algorithm 2 Benchmarking LMs using synthetic datasets (*SynTextBench*)

Input: Sentiment lexicons S, a range of difficulty levels P, an LM g, threshold accuracy a_T.

Output: SynTextBench score that quantifies the robustness-accuracy performance.

1: Construct positive/negative/neutral word lists from sentiment lexicon S.
2: **for** p in P **do**
3: Generate a synthetic binary classification task and obtain training set (x^{train}, y^{train}) and test set (x^{test}, y^{test}).
4: Calculate transformation T_1 and T_{-1} from $z_1^{train} = \{g(x) \mid (x, y) \in (x^{train}, y^{train}), y = 1\}$ and $z_{-1}^{train} = \{g(x) \mid (x, y) \in (x^{train}, y^{train}), y = -1\}$.
5: Transform training set and test set $\hat{z_1}^{train} = T_1(z_1^{train})$, $\hat{z_{-1}}^{train} = T_{-1}(z_{-1}^{train})$ and $\hat{z_1}^{test} = T_1(z_1^{test})$, $\hat{z_{-1}}^{test} = T_{-1}(z_{-1}^{test})$.
6: Derive the Bayes optimal classifier f according to $\text{sign}(\tilde{\mu}^T(\hat{z} - \frac{\mu_1+\mu_2}{2}))$ based on $\hat{z_1}^{train}$ and $\hat{z_{-1}}^{train}$, *i.e.* $\mu_1 = \text{mean}(\hat{z_1}^{train}), \mu_2 = \text{mean}(\hat{z_{-1}}^{train})$.
7: Read out the accuracy a of f on $\hat{z_1}^{test}$ and $\hat{z_{-1}}^{test}$, and calculate the average scale margin $\delta := avg(\|\bar{\Delta}_z\|_2)$ according to $\|\bar{\Delta}_z\|_2 = \frac{|(\hat{z} - \frac{\mu_1+\mu_2}{2})^T\tilde{\mu}|}{\|\tilde{\mu}\|_2^2}$ for correctly-classified sentence embeddings.
8: Denote the accuracy and average margin pair on the task by (a_p, δ_p).
9: **end for**
10: Define a goodness function $s(a) = \frac{1}{|P|}\sum_{\{p \in P, a_p > a\}} \delta_p$, for $a \in \mathbb{R}[0, 1]$.
11: SynTextBench score $= \int_{a_T}^{1} s(a)da$.

an informal but more intuitive statement: For any sample z, the Bayes optimal classifier f of class-balanced class-conditional Gaussian distribution P_{μ_1,μ_2,I_K}, yields a decision margin of $\|\Delta\|_2 = \frac{|(\hat{z} - \frac{\mu_1+\mu_2}{2})^T\tilde{\mu}|}{\|\tilde{\mu}\|_2}$, and if we scale the margin by the distance between two Gaussian centers, we obtain a scaled margin of $\|\bar{\Delta}_z\|_2 = \frac{|(\hat{z} - \frac{\mu_1+\mu_2}{2})^T\tilde{\mu}|}{\|\tilde{\mu}\|_2^2}$. We give the formal results for the generic class prior in Appendix Theorem A.4. To this end, we have prepared sentence embeddings and specified the way of calculating decision margins induced by a robust Bayes optimal classifier. In the following, we will state the complete algorithm for characterizing robustness-accuracy performance of LMs using synthetic datasets.

6.2.4 SynTextBench score and algorithm

With Section 6.2.2 and Section 6.2.3, we now can simulate synthetic tasks of a configured level of difficulty and evaluate their accuracy and margin. In our benchmarking process, we essentially build on this foundation to generate a sequence of tasks with different difficulty levels and inspect how the magnitude of decision margins changes with the classifier accuracy. In terms of robustness-accuracy characterization, it is desirable for an LM to consistently yield high classification accuracy, while maintaining a big decision margin (that is, less sensitive to perturbations in the embedding space). The pseudocode of the proposed framework, *SynTextBench*, is given in Algorithm 2.

In practice, we let $P = \{0, 0.05, \ldots, 0.9, 0.95\}$, and subsequently generate 20 synthetic datasets with $p = 0$ being the easiest and $p = 0.95$ being the hardest (*cf.* Section 6.2.2). Then, we perform analysis on the sentence embeddings of various synthetic datasets, and threshold the accuracy at a_T based on utility. The threshold serves as a penalty for poor sentence embeddings that lead to an undesirable accuracy under this threshold, matching our **task substance** of testing LM's ability to preserve linear separability. By referring to Figure 6-1, Line 2 in Algorithm 2 determines the word lists from a given lexicon. From Line 2 to Line 9, the for-loop generates one synthetic dataset at one time, on which we compute an (accuracy, average margin) pair (a_p, δ_p) and draw one point on the margin-accuracy 2D plot as in Figure 6-1. We apply Algorithm 2 on various models and obtain a margin-accuracy curve for each model. Since we not only care about the curvature of the curve but also how the (accuracy, average margin) pairs span on the curve, we define a goodness function $s(a) = \frac{1}{|P|} \sum_{\{p \in P, a_p > a\}} \delta_p$ on $\mathbb{R}[0, 1]$ in Line 10 to account for the span. By our definition, $s(a)$ will be a monotonically decreasing function (e.g., Figure 6-5) and calculate the expected margin conditioned on the accuracy level. The final SynTextBench score is defined by the integration over the desirable range of threshold accuracy in Line 11, *i.e.* SynTextBench score $= \int_{a_T}^{1} s(a)da$. We use SynTextBench as a quantifiable score to inform the accuracy-robustness aspect of a pretrained LM. In the later section, we will demonstrate the **metric informativeness** by measuring the correlation

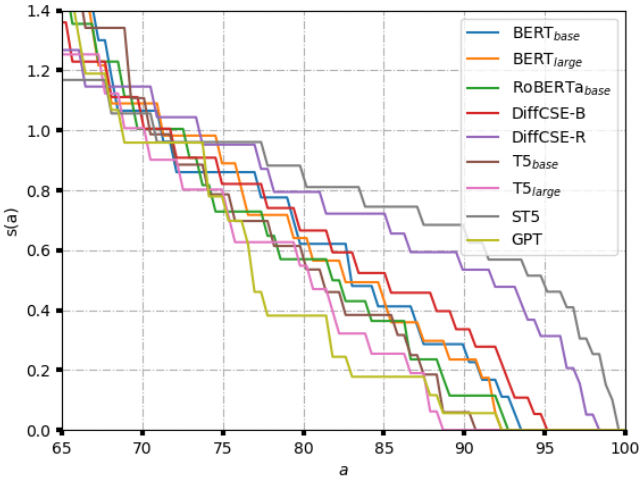


Figure 6-5: The goodness function $s(a)$ of nine pretrained LMs. The SynTextBench score is calculated by the area under the curve.

between SynTextBench scores and the average real-world sentence classification task performance.

6.3 Experiments

6.3.1 Setups

LMs. In the experiment, we will analyze the pretrained LMs predominantly considered by the sentence embedding literature [54, 169, 30], and also larger models such as LLaMA and OPT [179, 180, 220].. Specifically, we consider encoder models such as $\text{BERT}_{\text{base}}$, $\text{BERT}_{\text{large}}$ [38], $\text{RoBERTa}_{\text{base}}$ [112], DiffCSE-B, DiffCSE-R [30]; encoder-decoder models such as T5_{base}, T5_{large} [143], ST5 [125]; and decoder models such as DialogRPT [55]), LLaMA-7B, LLaMA-13B, LLaMA-30B [179], LLaMA-2-7B, LLaMA-2-13B [180], OPT-13B, OPT-30B [220]. For models that have an encoder component (encoder-only or encoder-decoder), we use the average output from the first and the last layer as sentence embeddings. For the decoder-only model, we use the embedding of the last token as sentence embeddings.

Baselines. We followed the open-source implementation of the literature [196]

and fed the pretrained LMs with synthetic texts generated according to Section 6.2.2 and reported the validation accuracy (Val loss), minimum description length (MDL), surplus description length (SDL), and ϵ-sample complexity (ϵSC) as baselines [8, 185, 196]. Since these methods take one dataset as inputs, we choose a relatively easy synthetic proxy task generated by $p = 0.2$ as the input dataset.

Objectives. Through the experiments, our main aim is to verify the feasibility of making performance assessments of possible downstream tasks by real-data-free evaluation methods. To achieve this, we will compare the Pearson correlation coefficients of assessments given by different real-data-free evaluation methods with the performance on real-world tasks. Since SynTextBench is intended to inform the robustness-accuracy performance, we will report both the accuracy and robustness on real-world tasks for studying correlation. We use PWWS attack [151] through TextAttack, a Python framework for adversarial attacks in NLP, to generate attacks. Essentially, the attacker will perturb the inputs gradually by changing more and more words until the perturbation leads to a wrong classification result. Therefore, we report the average number of perturbed words in a successful attack as an indicator of the level of model robustness. We will also demonstrate how SynTextBench can be used to do attribute comparisons. Finally, as more attentions have been drawn to large LMs lately, we will also conduct an extended study on large LMs and include discussions on in-context learning performance on SynTextBench synthetic data. We give more experimental details of our prompts in Appendix B.4.2.

6.3.2 Performance evaluation and discussion

We evaluate encoder models listed in Section 6.3.1 by SynTextBench framework as well as on real-world sentence embedding tasks. Specifically, we simulated 20 synthetic datasets as described in Section 6.2.4 and obtained one goodness function $s(a)$ for each LM. We plot these functions together in Figure 6-5, from which the final SynTextBench score can be determined by definition. We refer readers to Appendix Table B.14 for the exact numbers due to the page limit. To gauge the performance of these pretrained LMs on downstream real-world tasks, we evaluate the given models on

Table 6.2: Correlation between real-data-free evaluation metric and real-data accuracy at different synthetic dataset sizes.

n	4096	8192	16384	32768
Val loss	0.29±0.50	0.65±0.00	0.61±0.01	0.27±0.02
MDL	0.57±0.11	0.52±0.04	0.51±0.03	0.48±0.03
SDL, ε=1	0.57±0.11	0.51±0.04	0.43±0.02	0.31±0.01
εSC, ε=1	-	-	-	-0.04±0.00
SynTextBench	**0.94±0.01**	**0.97±0.01**	**0.96±0.00**	**0.93±0.00**

SentEval (the Evaluation Toolkit for Universal Sentence Representations [32]) and show the detailed numbers in Appendix Table B.15 and Figure B-8. SentEval tasks include seven semantic textual similarity tasks (denoted by "STS tasks"), where results are given by the Spearman's correlation with output range $[-1, 1]$, and seven transfer learning tasks (denoted by "Transfer task"), where results are given by the standard accuracy with range $[0, 1]$. We scale the former to the same range as the latter, $[0, 1]$, and take an average as the final accuracy indicator.

Correlation with real-world tasks. To demonstrate the informativeness of SynTextBench score, we list the Pearson correlation coefficients between real-data-free evaluation methods and the accuracy of SentEval tasks in Table 6.2. Five real-data-free metrics are considered that includes Val loss, MDL, SDL, εSC, and the proposed SynTextBench. Since the smaller the baseline metrics are, the better, we add a negative sign in front of them when calculating the Pearson correlation coefficient. As we have the flexibility of generating synthetic datasets with various sizes (number of sentences), we compare four configurations $n = \{4096, 8192, 16384, 32768\}$. From Table 6.2, we observe that SynTextBench consistently gives scores highly correlated with real-world task accuracy, with correlation coefficients that are above 0.9. For the four baselines, the highest correlation ever achieved is when $n = 8192$ and evaluated by Val loss, 0.65. It is noteworthy that SynTextBench is also a stabler metric as substantiated by the smaller standard deviation.

Ablation on the nesting structure. To showcase the effect of the nesting structure, we see that no nesting structure is a special case of our proposed framework when $p_n = 0$ (*cf.* equation 6.1). In Table 6.2, we have SynTextBench($p_n = 0.5$) = 0.97. In comparison, we run the analysis for $p_n = 0$ and obtain SynTextBench($p_n = 0$) = 0.92.

Table 6.3: Aggregated correlation with real-data-free evaluation metric and the robustness-accuracy performance, and its breakdown.

Correlation. w/	Rob.-Acc.	Rob.-STS	Rob.-Transfer
Val loss	-0.06±0.15	0.08±0.13	-0.13±0.24
MDL	0.64±0.06	0.55±0.08	0.62±0.03
SDL, ε=1	0.60±0.02	0.51±0.04	0.58±0.03
εSC, ε=1	-	-	-
SynTextBench	**0.76±0.04**	**0.76±0.03**	**0.69±0.05**

In conclusion, SynTextBench, with both parameters, outperform the baselines by large margins. Between the two, SynTextBench with the imposed structure further improves the correlation.

Robustness implications. To understand how real-data-free evaluation methods correlate with real-world task robustness-accuracy performance, we further analyze the correlation with the robustness indicator, the average number of perturbed words, on Transfer tasks when $n = 8192$. We focus on these tasks as they are classification tasks where adversarial attacks are well-defined. To combine robustness correlation with accuracy correlation, we add up two ranking vectors by robustness and accuracy measures, and calculate its Pearson correlation with the ranking by one of the real-data-free evaluation metrics (Val loss, MDL, SDL, ϵSC, SynTextBench). This way, we effectively obtain the aggregated Spearman correlation coefficient between real-data-free evaluation metrics and joint robustness-accuracy performance. We refer readers to Appendix B.4.4 for more experimental details. We list the results in Table 6.3. From the "Rob.-Acc." column, we see SynTextBench has an overall higher correlation with robustness-accuracy performance compared to other baselines. To be more precise, SynTextBench shows a coefficient of 0.76, whereas MDL and SDL are 0.64 and 0.60. Recall that accuracy results were aggregated from STS tasks and Transfer tasks. In Table 6.3, we also show how each component contributes to the correlation. In the "Rob.-STS" and "Rob.-Transfer" columns, we use only STS or Transfer task results as the accuracy measure when ranking the models, and the remaining steps follow. From the two columns, we see that SynTextBench still shows a stronger correlation compared to baselines, while having a slightly better correlation with Robustness-STS accuracy performance than Robustness-Transfer accuracy performance.

Table 6.4: Correlation between real-data-free evaluation metric and real-data accuracy on larger LMs.

Name	Pearson correlation
Val loss	0.80
MDL	-0.47
SDL, $\varepsilon = 1$	-0.55
εSC, $\varepsilon = 1$	-
SynTextBench	0.87

Case study on model comparisons. Besides having high correlation with real-world task performance, we show how SynTextBench can be used to make model comparisons. From Table B.14, one sees that, the SynTextBench score of ST5 is significantly higher than that of T5 across all dataset sizes n, e.g., ST5's 0.223 vs. T5's 0.130 when $n = 8192$. This indicates contrastive fine-tuning is beneficial for improving sentence embeddings. This conclusion is in sync with the observations from real-world tasks, where we see ST5 yields both higher accuracy and robustness according to Table B.14 and Table B.20. Specifically, ST5 has an average accuracy of 90.17 and robustness 13.23, whereas T5 has an average accuracy of 82.78 and robustness 12.21.

6.3.3 Extended study on large LMs

Since SynTextBench focuses on the sentence embeddings of LMs, of which larger decoder models generally do not have better performance than smaller encoder models [46], we have given most of our analysis on encoder models in [54]. Here, to demonstrate the generality of SynTextBench to various LM types, we analyze more large decoder LMs such as LLaMA and OPT [179, 180, 220].

Similar to Table 6.2, we calculated the Pearson correlation coefficients between real-data-free evaluation methods and the accuracy of SentEval tasks in Table 6.4. According to the table, SynTextBench also gives scores highly correlated with real-world task accuracy on decoder models, with a correlation coefficient of 0.87. We refer readers to Appendix Table B.17 for the complete results. Besides evaluating linear probing performance on our SynTextBench synthetic tasks, we also evaluate the few-shot in-context learning (ICL) performance on SynTextBench tasks. We calculate

the correlation between the ICL accuracy on SynTextBench tasks and that on SentEval tasks, and see again a strong correlation of 0.81 between them. We refer readers to Appendix B.4.2 for more details.

6.4 Discussions

6.4.1 Generating synthetic datasets with a language model

To generate synthetic sentences with configurable difficulties with an LM, we reuse the word lists constructed in Section 6.2.2 and constrain the LM vocabulary to be within the word lists. Concretely, let V be the original tokenizer vocabulary, POS be the set of positive words, NEU be the set of neutral words, NEG be the set of negative words, and STOP be the set of stop words (see B.4.3), then we constrain the LM vocabulary to be $\bar{V} = \tilde{V} \cup \text{STOP}$, where $\tilde{V}$ composes of $p \times 100\%$ NEU $\cap\, V$ elements and $(1-p) \times 100\%$ POS $\cap\, V$ elements for positive sentence generations ($(1-p) \times 100\%$ NEG $\cap\, V$ elements for negative sentence generations). Similar to the use of the mixing ratio p in Section 6.2.2, we intend to create a set of tasks with diverse difficulties herein via varying p. We generate synthetic sentences by completing any of the starting tokens {"There", "I", "You", "She", "He", "It", "They", "The"}. We print some generated sentence examples below:

POSITIVE

- "She's a sweet and kind girl."
- "The one thing that you have to do is look for other people."
- "There are also a number of new content that have been rolled out in recent times."
- "I had a lot of fun with this design."
- "She was one of several of several hundred people in the group to speak out against the police and their use of force."

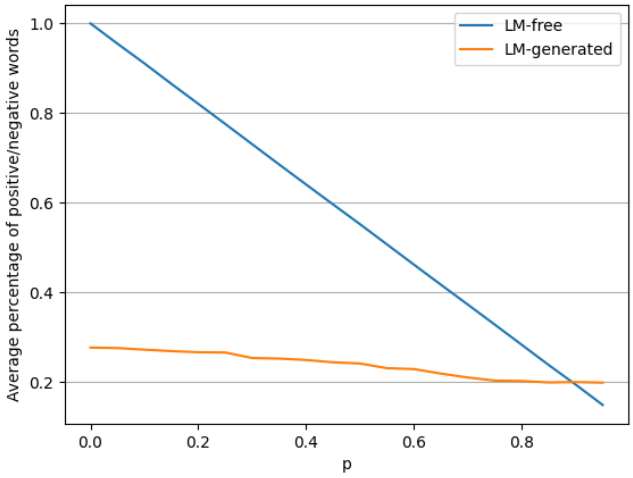


Figure 6-6: The average percentage of positive/negative words in the generated labeled positive/negative synthetic sentences.

- "You are very close to the truth.... if you are one of the first to see what is being done, that is very much a sign of an error.... you have to be very clear that it is a good thing that you are doing what you have to do....."

NEGATIVE

- "They were the worst of the worst."
- "She has no other option."
- "There's no question that the new and aggressive international community is headed for a bad start with its future in mind."
- "They do not want to see you there."
- "There's some real bad blood out there."
- "I just want to make sure that we are talking about our state government."

Discussions. Using LM-generated synthetic test sets, the rest of robustness-accuracy evaluation follows Section 6.2.3 and 6.2.4. We calculate the SynTextBench scores from LM-generated synthetic sentences and find that the Pearson correlation coefficient

between these scores and the actual downstream task performance is 0.633±0.011. This is in contrast to the higher correlation coefficient of above 0.9 observed from the LM-free synthetic sentences discussed in Section 6.2.2, as shown in Table B.5. In Figure 6-6, we plot the average percentage of positive and negative words in the generated labeled synthetic sentences. With an increasing mixing ratio p, we aim at configuring the task to be harder (data to be more mixed). While the percentage of positive/negative words does decrease in both LM-free synthetic sentences and LM-generated synthetic sentences, we have more control over LM-free generations in generating tasks at various difficulty levels (various y-axis values).

6.4.2 Synthetic sentence examples

POSITIVE

- "perfectibility lotus-eater shine shine health_care health_care pleasant-tasting"
- "convincingly gruesomely gruesomely convincingly deserve feeder exhaust exhaust debonaire stuffily stuffily anne_sexton wholeness wholeness rarefy conformable pretension pretension"
- "smarmily smarmily fairness covetously infuse soothing subtly subtly soothing"
- "precious grace the_right_way the_right_way absoluteness absoluteness"
- "personal_relation pleasurable sleekness cryptographically cryptographically correct delineate sink_in authenticated"
- "perfectibility lotus-eater shine shine health_care health_care pleasant-tasting"

NEGATIVE

- "unpleasant unpleasant mortal sympathetic dead dead choker nubbly fallout"
- "counterrevolutionary apprehensive thunderclap unskilled unskilled thunderclap apprehensive cheat shanny shanny cheat counterrevolutionary smooth smooth decayed decayed imagine imagine loser unpicturesque unnaturalized unnaturalized unrelieved unrelieved unhewn"

- "unpleasant unpleasant mortal sympathetic dead dead choker nubbly fallout"
- 'jostling weka offend engorged fouled fouled engorged intermittence space impaction impaction space intermittence dishonesty disgustingly"
- "blindly blindly"
- "second_class criminal_possession lousiness nonextensile linanthus_dianthiflorus nonarbitrary regular foolishness stabbing"

As we mentioned earlier in the chapter, the inclusion of the parenthesis is to guarantee we test for the linguistic structures, whose importance is repeatedly advocated in literatures from both machine learning and cognitive science. Therefore, when building synthetic test for the linguistic structures, we also follow the parenthesis and thus have non-grammatical test sets.

We would like to motivate their use based on the following example of sentiment analysis in food reviews. Upon seeing the review "love love fantastic!" in a food review, a reasonable language model should recognize the entailed positive sentiment, even though the sentence is non-grammatical. In our framework, to test the other basic skill for language acquisition in a systematic and scalable manner, we put words associated with binary labels (positive and negative) in the synthetic sentence and test sentence embeddings of LMs in identifying the words for sentence classification. Related to our setups herein, [92] also studies a range of summarization tasks from nonsense documents, in which a task is also designed to classify whether there are keywords indicating positive or negative sentiments ([92], Figure 1). Additional evidence of the usage of non-grammatical sentences can be found in [5], where authors also exploit non-grammartical synthetic sentence ([5], Appendix A) for constructing Gaussian logistic regression problems in improving reasoning ability in LMs, which manifests the value of non-grammatical language in learning/testing basic skills. Our high correlation with real-world tasks further suggests that better understanding of the synthetic sentences indeed implies better performance on real tasks. By construction, our framework is not limited to sentiment analysis as one can readily change the base

lexicon to test how LMs identify words describing other notions. For example, if we use the moral foundation lexicon, one can test how each LM identifies words that describe care, fairness, loyalty, authority, and sanctity.

6.5 Conclusion

In this chapter, we have proposed SynTextBench, a novel framework for evaluating the accuracy and robustness of LM sentence embeddings. SynTextBench is a configurable real-date-free lightweight testbed that generates steerable synthetic language datasets and proxy tasks, avoiding the risk of test-data leakage. SynTextBench is the pioneering effort in developing synthetic benchmarking methodologies for NLP, with a primary focus on sentence classification tasks and does not cover other NLP tasks such as question answering, machine translation, or summarization. By concentrating on this specific aspect, we have provided a solid foundation upon which future research can build. We believe that our work is a major step towards ensuring independent and sustainable auditing of LMs.

Chapter 7

Conclusions and Future Work

7.1 Summary of results

In this thesis, we have presented our progress towards understanding and improving the representational robustness of machine learning models. We summarize our results and contributions in Table 1.1. We discovered the potential side-effect of converting networks to their robust and smoothed counterparts, proved the occurrence of disparity in class-wise accuracy that could cause fairness concerns. Building on our understanding, we tried to improve the training of the base model by employing heavy augmentations. For generic non-smooth models, we provided an alternative way of interpreting contrastive learning, and proposed a new framework of training representation networks that simultaneously promote robustness. Eventually, we identified the drawbacks of current evaluations of representation networks, and gave a solution for assessing the robustness-accuracy quality of vision and language model representations in a task-agnostic way. We designed synthetic tests whose ground-truth is independent of the model to be evaluated and the test covers necessarily wide input domain.

Our main results of this thesis are summarized below.

Chapter 3 has pointed out the side effects of current randomized smoothing workflows and the limitations. We proved the hidden cost of randomized smoothing is class-wise fairness, *i.e.*, decision boundaries of smoothed classifiers will shrink,

resulting in disparity in class-wise accuracy. We further identified sufficient conditions under which Gaussian smoothing leads to a decrease in classification accuracy and characterized the theoretical lower bound of the shrinking rate. We also shown that noise augmentation in the training process (data augmentation) does not necessarily resolve the shrinking issue due to the inconsistent learning objectives. Finally, we analyzed the effect of noise augmentation and showed that it may leads to low classification accuracy for large σ on both synthetic and real datasets.

Chapter 4 has developed a generic framework called Integrated contrastive learning (**IntCl** and **IntNaCl**) that could simultaneously achieve good accuracy and robustness on downstream tasks. IntNaCl built on a link we established between contrastive learning and supervised neighborhood component analysis. We provided theoretical analysis on NaCl and show better generalization bounds over the baselines. The proposed integrated contrastive learning (IntCl and IntNaCl) could simultaneously achieve good accuracy and robustness on downstream tasks.

Chapter 5 has proposed synthetic data benchmarks for evaluating vision and language large representation networks (foundation models) called **SynBench**. To circumvent the need for real-world data in evaluation, we explored the use of synthetic binary classification tasks with Gaussian mixtures to probe pretrained models and compare the robustness-accuracy performance on pretrained representations with an idealized reference. SynBench offers a holistic evaluation, revealing intrinsic model capabilities and reducing the dependency on real-life data for model evaluation. Evaluated with various pretrained image models, the experimental results confirm that our task-agnostic evaluation correlates with actual linear probing performance on downstream tasks and can also guide parameter choice in robust linear probing to achieve a better robustness-accuracy trade-off.

Chapter 6 has extended SynBench to the language domain and proposed a new evaluation workflow that generates steerable synthetic language datasets and proxy tasks for benchmarking the performance of pertrained LMs on sentence classification tasks, named **SynTextBench**. SynTextBench utilizes a labeled lexicon and the nesting parenthesis structure to generate synthetic datasets. It allows for better

characterization of the joint analysis on the robustness and accuracy of LMs without risking sensitive information leakage through the API. It also provides a more controlled and private way to evaluate LMs that avoids overfitting specific test sets. Verified on various pretrained LMs, the proposed approach demonstrates promising high correlation with real downstream performance.

7.2 Future work

There exist a lot of topics worth further investigation. Below we summarize a few of them.

Geometry-aware randomized smoothing. As can be concluded from our analysis on smoothing methods using isotropic distribution, the smoothed classifier could suffer from worsened class-wise accuracy and therefore harm the fairness. This inspires us to think about how can one potentially choose smoothing distributions that are more robust to different data geometries. Without a proper smoothing distribution, one should stop using large smoothing factor in randomized smoothing. Furthermore, from the perspective of data augmentation, we see that applying data augmentation before performing randomized smoothing results in mismatched learning objectives. A preferred way may be augmenting the dataset by taking randomized smoothing risk into account and doing data-dependent randomized smoothing.

Representation learning. As discussed in Chapter 4, the current contrastive learning regime still falls under $k = 1$ for k-nearest neighbor in NCA. Thus, future work along the line includes addressing the current limitation of assuming $k = 1$ to $k > 1$ (kNCA [172]), by doing which we expect to extend the current framework to an even more general form. Additionally, we also want to point out the possibility of formulating contrastive learning as a multi-label classification problem and adopt

binary cross-entropy loss. That said, one can form the contrastive loss as follows:

$$\mathbb{E}_{\substack{x\sim\mathcal{D}, x_j^+\sim\mathcal{D}_x^+, \\ x_i^-\sim\mathcal{D}_x^-}}\left[-\frac{1}{M}\sum_{j=1}^{M}\log\sigma(f(x)^T f(x_j^+)) - \frac{1}{N}\sum_{i=1}^{N}\log\big(1-\sigma(f(x)^T f(x_i^-))\big)\right].$$

Task-agnostic foundation model evaluation. As the popularization of pretrained representations in various domains (e.g. vision, language, speech), we foresee SynBench and SynTextBench to be generalized to more domains and shed light on task-agnostic benchmarking designs that are simple and synthetic. Moreover, while Chapter 5 and 6 delve into the robustness-accuracy characterization of pretrained representations, we envision these frameworks to be further extended to other trustworthiness dimensions such as privacy, fairness, and other aspects. This extension is more intuitive in the language domain since one can easily swap the lexicon used to generate SynTextBench sentences to lexicons labeled with other trustworthiness dimensions. Lastly, while SynTextBench has a primary focus on sentence classification tasks, future work includes extensions to other NLP tasks such as question answering, machine translation, or summarization. We believe that our work is a major step towards ensuring independent and sustainable auditing of foundation models in general.

Appendix A

Proofs

A.1 Supporting proofs for Chapter 3

Definition A.1.1 (Smoothed)**.** If we use f to denote an original neural network function with outputs in the simplex $\Delta^c = \{z \in \mathbb{R}^c \mid \sum_{i=1}^{c} z_i = 1,\, 0 \leq z_i \leq 1,\, \forall i\}$, then its smoothed counterpart defined on d-dimensional inputs $x \in \mathbb{R}^d$ is defined by

$$f_{\text{smooth}}(x) = \int_{x' \in \mathbb{R}^d} f(x')p(x')dx',$$

where $p(x')$ is the probability density function of the filter.

Definition A.1.2 (Gaussian smoothing)**.** If $p(x')$ is the probability density function of a normally-distributed random variable with an expected value x and standard deviation σ, then we call f_{smooth} a Gaussian-smoothed function and denote it by f_σ.

Definition A.1.3 (Regularized Gamma Function)**.** The lower regularized gamma functions $Q(s, x)$ is defined by

$$Q(s, x) = \frac{\int_0^x t^{s-1}e^{-t}dt}{\int_0^\infty t^{s-1}e^{-t}dt}.$$

Lemma A.1. $\Phi[x] + \Phi[\frac{1}{x}] \geq 1.5$ *with equality holds iff* $x \in \{0, \infty\}$.

Proof. Let $f(x) = \Phi[x] + \Phi[\frac{1}{x}]$. We observe that $f(x) = f(1/x)$ by definition. So, it is

sufficient to show that for x in the interval $(1, \infty)$, $f(x) \geq 1.5$ with equality at $x \to \infty$. We prove this by showing that in the interval $(1, \infty)$, $f(x)$ is strictly decreasing and $\lim_{x\to\infty} f(x) = \lim_{x\to\infty} \Phi(x) + \Phi(1/x) = \Phi(\infty) + \Phi(0) = 1 + 0.5 = 1.5$. To show f(x) is strictly decreasing we proceed by taking the derivative wrt x,

$$\frac{d}{dx} f(x) = \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}} - \frac{e^{-\frac{1}{2x^2}}}{x^2\sqrt{2\pi}}$$

we show that for the interval $(1, \infty)$ this derivative is less than 0. So, we need to show that

$$\begin{aligned}
&\frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}} - \frac{e^{-\frac{1}{2x^2}}}{x^2\sqrt{2\pi}} < 0 \\
\Leftrightarrow \quad & x^2 e^{-\frac{x^2}{2}} < e^{-\frac{1}{2x^2}} \\
\Leftrightarrow \quad & \log\left(x^2\right) + \frac{1}{2x^2} < x^2 \\
& \text{Let } t = \log\left(x^2\right), x > 1 \to t > 0 \\
\Leftrightarrow \quad & 2t < e^t - e^{-t}
\end{aligned}$$

This holds for $t > 0$ as we have that at $t = 0$. $2 \cdot 0 = 0 = e^0 - e^{-0}$ and $2t$ increases at a rate of 2 while $e^t - e^{-t}$ increases at a rate of $e^t + e^{-t} > 2 \cdot \sqrt{e^t \cdot e^{-t}} = 2$ as $t > 1 \to e^t \neq e^{-t}$. Finally for $x = 1$, we calculate $f(x) \approx 1.6829 > 1.5$. □

Theorem 1. *Consider a classifier f_{train,σ_t} given as the naive-Bayes classifier obtained by training on the dataset $\mathcal{X}$ with data augmentation of variance σ_t. Let the class-wise accuracy of f_{train,σ_t} using the randomized smoothing prediction rule be given as $Acc_1(\sigma_t), Acc_2(\sigma_t)$. Then we define the bias ($\Delta(\sigma_t)$) to be the gap between class-wise accuracies ($\Delta(\sigma_t) = |Acc_1(\sigma_t) - Acc_2(\sigma_t)|$). For $k > \frac{1}{2\epsilon} - 1$, class I decision region grows in size at a rate of $O(\sigma_t^2)$ and thus the bias is large for large σ_t.*

Proof. In order to determine the accuracies we start by looking at the decision regions given by the two classifiers. We show that the decision region of class 1 increases with increasing σ effectively increasing the bias by increasing the class 1 accuracy while decreasing the class 2 accuracy.

From the structure of the dataset it is easy to show that the naive Bayes classifier yield decision regions:

class 1 : $[-(\frac{a}{2}+c_0(\sigma)), \frac{ka}{2}+d_0(\sigma)]$

class 2 : $[-\infty, -(\frac{a}{2}+c_0(\sigma))] \cup [\frac{ka}{2}+d_0(\sigma), +\infty]$

The likelihood ratio function $r_\sigma(x) = \frac{p(x\in \text{ class 2})}{p(x\in \text{ class 1})} = (1-2\epsilon)e^{\frac{-a(2x+a)}{2\sigma^2}} + 2\epsilon e^{\frac{(2x-ka)ka}{2\sigma^2}}$. This is a convex function is x resulting in the previous form of decision regions. Thus, we get the following decision regions after smoothing, class 1 $[-(\frac{a}{2}+c_1(\sigma)), \frac{ka}{2}+d_1(\sigma)]$ and the rest being class 2.

In this case we show that for $c_0(\sigma)$ grows at $\Theta(\sigma^2)$ with increasing σ by establishing a lower bound and upper bound which both grow at the rate of $O(\sigma^2)$.

For the lower bound consider the function $r^u_\sigma(x) = (1-2\epsilon)e^{\frac{ax}{\sigma^2}} + 2\epsilon e^{\frac{-kax}{\sigma^2}} > r_\sigma(-(\frac{a}{2}+x))$. If for any $c_l(\sigma)$ we have $r^u_\sigma(c_l(\sigma)) = 1$, then $r_\sigma(-(\frac{a}{2}+c_l(\sigma))) < 1$. Thus, we see that using the convexity argument from before $c_0(\sigma) > c_l(\sigma)$. But it is easy to see that if $c_l(1)$ is a solution of the equation $r^u_1(x) = 1$ at $\sigma = 1$, then $\sigma^2 c_l(1)$ is a solution for $r^u_\sigma(x) = 1$.

As r^u_1 is a continuous function with $r^u_1(0) = 1$ and $\lim_{x\to\infty} r^u_1(x) \to \infty$, it is sufficient to show that $\frac{d}{dx}r^u_1(0) = a(1-2\epsilon(k+1)) < 0$ (follows from the case condition) to show that $r^u_1(x) = 1$ has a positive real solution and consequently $c_0(\sigma) > \sigma^2 c_l(1) = O(\sigma^2)$. From the likelihood function, we can also clearly see that $r_\sigma(-(\frac{a}{2}+x)) > (1-2\epsilon)e^{\frac{ax}{\sigma^2}}$. Using this we can establish that $c_0(\sigma) < \frac{\sigma^2 - \log(1-2\epsilon)}{a}$ making $c_0(\sigma) = \Theta(\sigma^2)$.

As $d_0(\sigma) \geq 0$, we have that for all $\sigma \in (0,\infty)$ the size of the interval $[-(\frac{a}{2} + c_0(\sigma)), \frac{ka}{2}+d_0(\sigma)]$ is bigger that $C\sigma^2 + C$ for some positive constant C. Thus, we have that at $x = -(\frac{a}{2}+c_0(\sigma)-\frac{1}{C})$ the probability $x \in$ Class I after smoothing is given as $\Phi(\frac{C\sigma^2}{\sigma_t}) - \Phi(\frac{-1/C}{\sigma_t})$. By Lemma A.1, we get that $\Phi(\frac{C\sigma^2}{\sigma_t}) - \Phi(\frac{-1}{C\sigma_t}) > \Phi(\sigma_t C) - \Phi(\frac{-1}{C\sigma_t}) = \Phi(\sigma_t C) - (1-\Phi(\frac{1}{C\sigma_t})) = \Phi(\sigma_t C) + \Phi(\frac{1}{C\sigma_t}) - 1 > 0.5$. Thus, we have $c_1(\sigma) > c_0(\sigma) - \frac{1}{C}$. Combining this with the fact that clearly $c_0(\sigma) > c_1(\sigma)$, we have $c_1(\sigma) \in (c_0(\sigma) - \frac{1}{C}, c_0(\sigma))$ and similarly, we also have $d_1(\sigma) \in (d_0(\sigma) - \frac{1}{C}, d_0(\sigma))$. This also gives us $c_1(\sigma) = \Theta(\sigma^2) = \Theta(\sigma_t^2)$.

Consider the function $f_x(\sigma) = r_\sigma(x)$. By differentiating this function wrt σ we see that it has only one extremum point. Using the fact that $\lim_{\sigma\to\infty} f_x(\sigma) = 1$ we have

that if for any x, $f_x(\sigma) = 1$ then we see that there the extremum point lies between σ and ∞. If for any $\sigma' > \sigma$, $f_x(\sigma') = 1$, then there would be a two extremum points one between σ, σ' and another between σ', ∞. Using this along with the continuity of f_x we get that either $f_x(\sigma') < 1 \forall \sigma' > \sigma$ or $f_x(\sigma') > 1 \forall \sigma' > \sigma$. We can further use the fact that $f_x(0) \to \infty$ to see that f_x is decreasing at σ making $f_x(\sigma') < 1 \forall \sigma' > \sigma$. Thus, we see that $d_0(\sigma), c_0(\sigma)$ are increasing functions of σ. Combining this with the previous result shows that the decision region of class I after smoothing increases at $O(\sigma_t^2)$.

For the bias we see that as $\sigma_t \to \infty$, class I at least occupies the region $(-\infty, \frac{ka}{2}]$ while class II occupies at most the region $(\frac{ka}{2}, \infty)$. As a result the bias is lower bounded by $(1 - \Phi(\frac{-ka}{2\sigma_o})) - \epsilon(1 - \Phi(\frac{-ka}{2\sigma_o})) = (1 - \epsilon)(1 - \Phi(\frac{-ka}{2\sigma_o}))$ which is very high. □

Theorem 2. *Consider a classifier f_{train,σ_t} given as the naive-Bayes classifier obtained by training on the dataset $\mathcal{X}'$ with data augmentation of variance σ_t. The bias of the classifier f_{train,σ_t} using the randomized smoothing prediction rule is $1 - \epsilon$, if $k > \frac{e^2}{\epsilon} - 1$ and $\sigma_t \geq a\sqrt{\frac{k(k+1)}{2ln(2\epsilon(k+1)) - \frac{2k}{k+2}}}$.*

Proof. At $x = -a$, we see that if the decision region for class 1 is $[-(a + c), \frac{ka}{2} + d]$, then the probability after smoothing is

$$
\begin{aligned}
g(-a, 1) &= \int_{x' \in \mathbb{R}^d} d(-a, x')\psi(x', 1)dx' \\
&= \int_{-(a+c)}^{\frac{ka}{2}+d} d(-a, x')dx' \\
&= \int_{-\infty}^{\frac{ka}{2}+d} d(-a, x')dx' - \int_{-\infty}^{-(a+c)} d(-a, x')dx' \\
&= \Phi(\frac{\frac{ka}{2} + d + a}{\sigma}) - \Phi(\frac{-c}{\sigma}) \\
&\geq \Phi(\frac{\frac{k+2}{2}a}{\sigma}) - \Phi(\frac{-c}{\sigma}) \quad (\text{if } d \geq 0) \\
&\geq \Phi(\frac{\frac{k+2}{2}a}{\sigma}) - \Phi(-\frac{\sigma}{\frac{k+2}{2}a}) \quad (\text{if } c \geq \frac{2\sigma^2}{(k+2)a}) \\
&> 0.5. \quad (\text{by Lemma A.1})
\end{aligned}
$$

That's said, the bias will be at least $1-\epsilon$ if $d \geq 0$ and $c \geq \frac{2\sigma^2}{(k+2)a}$ are true. We now check for $d \geq 0$: for $x \in [0, \frac{ka}{2}]$,

$$\begin{aligned}
\psi(x,1) &= \int_{x' \in \mathbb{R}^d} d(x,x')\rho(x',1)dx' && = d(x,0)\rho(0,1) \\
&= \frac{1}{\sqrt{2\pi\sigma^2}}[\frac{1}{2}e^{-\frac{x^2}{2\sigma^2}}] && = \frac{1}{\sqrt{2\pi\sigma^2}}[(\frac{1}{2}-\epsilon)e^{-\frac{x^2}{2\sigma^2}} + \epsilon e^{-\frac{x^2}{2\sigma^2}}] \\
&> \frac{1}{\sqrt{2\pi\sigma^2}}[(\frac{1}{2}-\epsilon)e^{-\frac{(x+a)^2}{2\sigma^2}} + \epsilon e^{-\frac{(ka-x)^2}{2\sigma^2}}] && = d(x,-a)\rho(-a,2) + d(x,ka)\rho(ka,2) \\
&= \int_{x' \in \mathbb{R}^d} d(x,x')\rho(x',2)dx' = \psi(x,2),
\end{aligned}$$

implying $x \in [0, \frac{ka}{2}]$ belongs to class 1 for the naive Bayes classifier. Therefore the decision region for class 1 extends at least to $\frac{ka}{2} + d$ with $d \geq 0$. Next, we check for $c \geq \frac{2\sigma^2}{(k+2)a}$: at $x = -a - \frac{2\sigma^2}{(k+2)a}$, the probability is

$$\begin{aligned}
\psi(-a - \frac{2\sigma^2}{(k+2)a}, 1) &= \int_{x' \in \mathbb{R}^d} d(-\frac{2\sigma + a}{(k+2)\frac{a}{\sigma}}, x')\rho(x',1)dx' \\
&= \frac{1}{\sqrt{2\pi\sigma^2}}[\frac{1}{2}e^{-\frac{x^2}{2\sigma^2}}]\Big|_{x=-a-\frac{2\sigma^2}{(k+2)a}} \\
\psi(-a - \frac{2\sigma^2}{(k+2)a}, 2) &= \int_{x' \in \mathbb{R}^d} d(-\frac{2\sigma + a}{(k+2)\frac{a}{\sigma}}, x')\rho(x',2)dx' \\
&= \frac{1}{\sqrt{2\pi\sigma^2}}[(\frac{1}{2}-\epsilon)e^{-\frac{(x+a)^2}{2\sigma^2}} + \epsilon e^{-\frac{(ka-x)^2}{2\sigma^2}}]\Big|_{x=-a-\frac{2\sigma^2}{(k+2)a}}.
\end{aligned}$$

Therefore we see that $\psi(-a-\frac{2\sigma^2}{(k+2)a},1) > \psi(-a-\frac{2\sigma^2}{(k+2)a},2)$ if

$$
\begin{aligned}
& (1-2\epsilon)e^{(\frac{a}{\sigma})^2\frac{1}{2}+\frac{2}{k+2}} + 2\epsilon e^{-\frac{k(k+2)}{2}(\frac{a}{\sigma})^2-\frac{2k}{k+2}} < 1 \\
\Leftrightarrow \quad & (1-2\epsilon)[e^{(\frac{a}{\sigma})^2\frac{1}{2}+\frac{2}{k+2}} - 1] < 2\epsilon[1-e^{-\frac{k(k+2)}{2}(\frac{a}{\sigma})^2-\frac{2k}{k+2}}] \\
\Leftrightarrow \quad & \frac{1}{2\epsilon} - 1 < \frac{1-e^{-\frac{k(k+2)}{2}(\frac{a}{\sigma})^2-\frac{2k}{k+2}}}{e^{(\frac{a}{\sigma})^2\frac{1}{2}+\frac{2}{k+2}}-1} \\
\Leftrightarrow \quad & \frac{1}{2\epsilon} < \frac{e^{(\frac{a}{\sigma})^2\frac{1}{2}+\frac{2}{k+2}} - e^{-\frac{k(k+2)}{2}(\frac{a}{\sigma})^2-\frac{2k}{k+2}}}{e^{(\frac{a}{\sigma})^2\frac{1}{2}+\frac{2}{k+2}}-1} \\
\Leftrightarrow \quad & \frac{1}{2\epsilon} < \frac{\tau l - \tau^{-k(k+2)}l^{-k}}{\tau l - 1} \quad (\text{let } \tau = e^{(\frac{a}{\sigma})^2\frac{1}{2}}, l = e^{\frac{2}{k+2}}) \\
\Leftrightarrow \quad & \frac{1}{2\epsilon} < \tau^{-k(k+2)}l^{-k}\frac{\tau^{(k+1)^2}l^{k+1}-1}{\tau l - 1} \\
\Leftrightarrow \quad & \frac{1}{2\epsilon} < \tau^{-k(k+2)}l^{-k}\frac{\tau^{(k+1)^2}l^{k+1}-1}{\tau^{k+1}l-1}\frac{\tau^{k+1}l-1}{\tau l-1} \\
\Leftrightarrow \quad & \frac{1}{2\epsilon} < \tau^{-k(k+1)}l^{-k}(\Sigma_{i=0}^{k}(\tau^{k+1}l)^i)\frac{\tau^{k+1}l-1}{\tau l-1}\tau^{-k} \\
\Leftrightarrow \quad & \frac{1}{2\epsilon} < (\Sigma_{i=0}^{k}(\tau^{k+1}l)^{-i})\frac{\tau l-\tau^{-k}}{\tau l-1} \\
\Leftarrow \quad & \frac{1}{2\epsilon} \le \Sigma_{i=0}^{k}(\tau^{k+1}l)^{-i} \le (k+1)(\tau^{k+1}l)^{-k} \\
\Leftrightarrow \quad & 0 < ln(\tau) \le \frac{ln(2\epsilon(k+1)) - kln(l)}{k(k+1)} = \frac{ln(2\epsilon(k+1)) - \frac{2k}{k+2}}{k(k+1)} \\
\Leftarrow \quad & (\frac{a}{\sigma})^2\frac{1}{2} \le \frac{ln(2\epsilon(k+1)) - \frac{2k}{k+2}}{k(k+1)},\ k > \frac{e^2}{\epsilon} - 1 \\
\Leftrightarrow \quad & \sigma \ge a\sqrt{\frac{k(k+1)}{2ln(2\epsilon(k+1)) - \frac{2k}{k+2}}},\ k > \frac{e^2}{\epsilon} - 1.
\end{aligned}
$$

These conclude our proof. □

Lemma A.2. *For any two original decision regions A, B, if we have that $A \subset B$, then we also have that $A_\sigma \subset B_\sigma$, where A_σ and B_σ are the decision regions of the Gaussian-smoothed functions.*

Proof. Recalling that decision regions A_σ and B_σ satisfy $D_\sigma = \{x \in \mathbb{R}^d | f_\sigma^D(x)_1 \ge \frac{1}{c}\}$

for $D = A, B$. Therefore for $\forall x \in A_\sigma$, we have $f_\sigma^A(x) \geq \frac{1}{c}$. And

$$\begin{aligned} f_\sigma^B(x)_1 &= \int_{x' \in \mathbb{R}^d} f^B(x')_1 p(x') dx' = \int_{x' \in \mathbb{R}^d} \mathbb{1}_{x' \in B} p(x') dx' \\ &= \int_{x' \in B} p(x') dx' > \int_{x' \in A} p(x') dx' \\ &= \int_{x' \in \mathbb{R}^d} \mathbb{1}_{x' \in A} p(x') dx' = \int_{x' \in \mathbb{R}^d} f^A(x')_1 p(x') dx' \\ &= f_\sigma^A(x)_1 \geq \frac{1}{c}, \end{aligned}$$

implying $x \in B_\sigma$. That said, we have that if $x \in A_\sigma$, then $x \in B_\sigma$, making $A_\sigma \subseteq B_\sigma$. □

It is well-known that

$$Q\left(\frac{d}{2}, \frac{R^2}{2\sigma^2}\right) = \int_{x' \in \mathbb{R}^d, \|x'\|_2 \leq R} (2\pi\sigma^2)^{-\frac{d}{2}} e^{\frac{x'^T x'}{2\sigma^2}} dx'.$$

For the dimension d, we summarize the lemma based on regularized Gamma functions below.

Lemma A.3. *For $\forall d, c \in \mathbb{N}^+$, $Q(\frac{d}{2}, \frac{d}{2c}) < \frac{1}{c}$ holds.*

Proof. To prove $Q(\frac{d}{2}, \frac{d}{2c}) < \frac{1}{c})$, by definition A.1.3, we aim at proving $\int_0^\infty t^{\frac{d}{2}-1} e^{-t} dt > c \cdot \int_0^{\frac{d}{2c}} t^{\frac{d}{2}-1} e^{-t} dt$ $(\forall d \in \mathbb{N}^+)$. For c = 1, this is clearly true as $t^{x-1}e^{-t} \geq 0$ is true for $t \geq 0$. Then we show it also holds for $c \geq 2$.

Let $g(t) = t^{x-1}e^{-t}$, we have $g'(t) = t^{x-2}e^{-t}(x - 1 - t)$. Therefore $g(t)$ is increasing when $t \leq x - 1$ and decreasing when $t > x - 1$. Thus, giving us two equations

$$\begin{aligned} &\int_{\frac{x}{c}}^{x} t^{x-1} e^{-t} dt > \min\{x^{x-1}e^{-x}, (\frac{x}{c})^{x-1} e^{-\frac{x}{c}}\} \frac{(c-1)x}{c} \\ &\frac{x}{c} (\frac{x}{c})^{x-1} e^{-\frac{x}{c}} > \int_0^{\frac{x}{c}} t^{x-1} e^{-t} dt \end{aligned}$$

So, we see that for any x, c if we have $x^{x-1}e^{-x} \geq (\frac{x}{c})^{x-1} e^{-\frac{x}{c}}$ then $\int_{\frac{x}{c}}^{x} t^{x-1} e^{-t} dt > (c-1) \cdot \int_0^{\frac{x}{c}} t^{x-1} e^{-t} dt \Leftrightarrow \int_0^x t^{x-1} e^{-t} dt > c \cdot \int_0^{\frac{x}{c}} t^{x-1} e^{-t} dt$. Using $t^{x-1}e^{-t} \geq 0, \forall x \int_0^\infty t^{x-1} e^{-t} dt \geq \int_0^x t^{x-1} e^{-t} dt$. So, we have $\int_0^\infty t^{x-1} e^{-t} dt > c \cdot \int_0^{\frac{x}{c}} t^{x-1} e^{-t} dt$ as needed. So, for any x, c

it is sufficient to show

$$x^{x-1}e^{-x} \geq (\frac{x}{c})^{x-1}e^{-\frac{x}{c}}$$

in order to prove $\int_0^\infty t^{x-1}e^{-t}dt > c \cdot \int_0^{\frac{x}{c}} t^{x-1}e^{-t}dt$. The inequality can be re-written as $(x-1)\log(c) > \frac{c-1}{c}x$ or $(1-\frac{1}{x}) > (1-\frac{1}{c})\frac{1}{\log(c)}$. We observe that $(1-\frac{1}{c})\frac{1}{\log(c)}$ is a decreasing function of c for $c \geq 1$ and $(1-\frac{1}{x})$ is an increasing function of x.

For $x \geq 4, c \geq 2$, we see $(1-\frac{1}{x}) \geq 1-\frac{1}{4} = 0.75 > (1-\frac{1}{2})\frac{1}{\log(2)} \geq (1-\frac{1}{c})\frac{1}{\log(c)}$.

For $x \geq \frac{3}{2}, c \geq 20$, we have $(1-\frac{1}{x}) \geq 1-\frac{2}{3} > (1-\frac{1}{20})\frac{1}{\log(20)} \geq (1-\frac{1}{c})\frac{1}{\log(c)}$.

For $\frac{3}{2} \leq x < 4 \rightarrow 3 \leq d < 8$ and $2 \leq c < 20$, we numerically verify the values of $Q(\frac{d}{2}, \frac{d}{2c})$ to see the inequality is satisfied.

Thus, for $d \geq 3, c \geq 2$ we have the inequality.

For $d = 2$, we have $Q(\frac{d}{2}, \frac{d}{2c}) = Q(1, \frac{1}{c})$. This has a closed form solution $Q(1,x) = 1-e^{-x}$. So, we need to show that for $c \geq 2$ $1-e^{-\frac{1}{c}} < \frac{1}{c}$ or $e^{\frac{1}{c}} < \frac{c}{c-1}$ or $\frac{1}{c} < \log\big(1+\frac{1}{c-1}\big)$. But we know that for $x > -1, x \neq 0$, $\log(1+x) > \frac{x}{x+1}$, so $\log\big(1+\frac{1}{c-1}\big) > \frac{\frac{1}{c-1}}{1+\frac{1}{c-1}} = \frac{1}{c}$ which concludes the proof for $d = 2, c \geq 2$. □

Lemma A.4. *Assume the decision region of class 1 data is* $\{x \in \mathbb{R}^d \mid \|x\|_2 \leq R\}$, *the point at the origin has the highest probability to be classified as class 1 by the gaussian-smoothed classifier* f_σ, *i.e.* $f_\sigma(x)_1 \leq f_\sigma(0)_1$.

Proof. We do the proof by mathematical induction and begin by giving $d = 1$ case. For $\forall R > 0$ and $d = 1$, equation 3.1 reduces to

$$\begin{aligned} f_\sigma(x)_1 &= \int_{-R}^{R} (2\pi\sigma^2)^{-\frac{1}{2}} e^{-\frac{(x'-x)^2}{2\sigma^2}} dx' \\ &\overset{a=x'-x}{=\!=\!=} \int_{-R-x}^{R-x} (2\pi\sigma^2)^{-\frac{1}{2}} e^{-\frac{a^2}{2\sigma^2}} da \\ f'_\sigma(x)_1 &= -(2\pi\sigma^2)^{-\frac{1}{2}} e^{-\frac{(R-x)^2}{2\sigma^2}} - (-1)(2\pi\sigma^2)^{-\frac{1}{2}} e^{-\frac{(-R-x)^2}{2\sigma^2}} \end{aligned}$$

and $f'_\sigma(x)_1$ equals to zero only when $x = 0$. Now suppose the conclusion holds for

$d-1$ dimensional case, then when $x \in \mathbb{R}^d$ we scale $f_\sigma(x)_1$ by $(2\pi\sigma^2)^{\frac{d}{2}}$ and obtain

$$
\begin{aligned}
&\int_{\|x'\|_2 \leq R} e^{-\frac{(x'-x)^T(x'-x)}{2\sigma^2}} dx' \\
&= \int_{\sum_{k=1}^d {x'_k}^2 \leq R^2} e^{-\frac{\sum_{k=1}^d (x'_k - x_k)^2}{2\sigma^2}} dx' \\
&= \int_{-R}^{R} \int_{\sum_{k=1}^{d-1} {x'_k}^2 \leq R^2 - {x'_d}^2} e^{-\frac{\sum_{k=1}^{d-1} (x'_k - x_k)^2}{2\sigma^2}} dx'_1 \dots dx'_{d-1} e^{-\frac{(x'_d - x_d)^2}{2\sigma^2}} dx'_d \\
&\leq \int_{-R}^{R} \int_{\sum_{k=1}^{d-1} {x'_k}^2 \leq R^2 - {x'_d}^2} e^{-\frac{\sum_{k=1}^{d-1} {x'_k}^2}{2\sigma^2}} dx'_1 \dots dx'_{d-1} e^{-\frac{(x'_d - x_d)^2}{2\sigma^2}} dx'_d \\
&= \int_{\sum_{k=1}^d {x'_k}^2 \leq R^2} e^{-\frac{\sum_{k=1}^{d-1} {x'_k}^2}{2\sigma^2}} e^{-\frac{(x'_d - x_d)^2}{2\sigma^2}} dx' \\
&= \int_{\sum_{k=1}^{d-1} {x'_k}^2 \leq R^2} \int_{{x'_d}^2 \leq R^2 - \sum_{k=1}^{d-1} {x'_k}^2} e^{-\frac{(x'_d - x_d)^2}{2\sigma^2}} dx'_d e^{-\frac{\sum_{k=1}^{d-1} {x'_k}^2}{2\sigma^2}} dx'_1 \dots dx'_{d-1} \\
&\leq \int_{\sum_{k=1}^{d-1} {x'_k}^2 \leq R^2} \int_{{x'_d}^2 \leq R^2 - \sum_{k=1}^{d-1} {x'_k}^2} e^{-\frac{{x'_d}^2}{2\sigma^2}} dx'_d e^{-\frac{\sum_{k=1}^{d-1} {x'_k}^2}{2\sigma^2}} dx'_1 \dots dx'_{d-1} \\
&= \int_{\sum_{k=1}^d {x'_k}^2 \leq R^2} e^{-\frac{\sum_{k=1}^d {x'_k}^2}{2\sigma^2}} dx',
\end{aligned}
$$

where the first inequality comes from the assumption that the conclusion holds for $d-1$ dimensional case with equality if and only if $x_1 = \dots x_{d-1} = 0$, and the second inequality comes from an one dimensional observation with equality precisely when $x_d = 0$. This concludes our proof. □

Since the value of $f_\sigma(0)_1$ depends on the radius R of the decision region, the dimension d, and the smoothing factor σ, we denote $f_\sigma(0)_1$ by $q(R, d, \sigma)$, *i.e.* $q(R, d, \sigma) := f_\sigma(0)_1$.

Corollary A.1. *As $\mathcal{D} \subseteq \mathcal{C}^{\mathcal{D}}_{\theta,v}$, using Lemma A.2, we have that the smoothed decision region is contained within the smoothed version of $\mathcal{C}^{\mathcal{D}}_{\theta,v}$, i.e. $\mathcal{D}_\sigma \subseteq (\mathcal{C}^{\mathcal{D}}_{\theta,v})_\sigma$.* □

Lemma A.5. *If the decision region of class 1 data is $\mathcal{D} = \{x \in \mathbb{R}^d \mid v^T x + \|v\|\|x\| cos(\theta) \leq 0\}$, where $v = [0, \dots, 0, 1]^T \in \mathbb{R}^d$ and $2\theta \in (-\pi, \pi)$, then after smoothing among the set of points S_a with the same projection on v the point on the axis has the highest probability of being in class 1. For $S_a = \{x \mid v^T x = a\}$, we have $argsup_{x \in S_a} f_\sigma(x)_1 = a \cdot v$. Moreover if $a_1 > a_2$, then $f_\sigma(a_1 \cdot v)_1 < f_\sigma(a_2 \cdot v)_1$.*

Proof. For the first part of the proof consider the set of points $S_a = \{x \mid v^T x = a\}$. For any point x is S_a, we see that

$$\begin{aligned}
f_\sigma(x)_1 &= \int_{x' \in \mathbb{R}^d} f(x')_1 p(x') dx' \\
&= \int_{x'_d + \|x'\| cos(\theta) \leq 0} (2\pi\sigma^2)^{-\frac{d}{2}} e^{-\frac{(x'-x)^T(x'-x)}{2\sigma^2}} dx' \\
&= (2\pi\sigma^2)^{-\frac{d}{2}} \int_{-\infty}^{0} \int_{\Sigma_{k=1}^{d-1} {x'_k}^2 \leq tan^2(\theta) {x'_d}^2} e^{-\frac{\sum_{k=1}^{d-1}(x'_k - x_k)^2}{2\sigma^2}} dx'_1 \dots dx'_{d-1} e^{-\frac{(x'_d - a)^2}{2\sigma^2}} dx'_d \\
&\leq (2\pi\sigma^2)^{-\frac{d}{2}} \int_{-\infty}^{0} \int_{\Sigma_{k=1}^{d-1} {x'_k}^2 \leq tan^2(\theta) {x'_d}^2} e^{-\frac{\sum_{k=1}^{d-1} {x'_k}^2}{2\sigma^2}} dx'_1 \dots dx'_{d-1} e^{-\frac{(x'_d - a)^2}{2\sigma^2}} dx'_d \\
&= f_\sigma(av)_1.
\end{aligned}$$

where the inequality comes from Lemma A.4 with equality iff $x_1 = \dots x_{d-1} = 0$, *i.e.* $x = [0, \dots, 0, a] \in \mathcal{V}$. Now for the second part of the proof, let $x_1 = a_1 v$, $x_2 = a_2 v$ such that $a_1 > a_2$. Then

$$\begin{aligned}
f_\sigma(x_1)_1 &= (2\pi\sigma^2)^{-\frac{d}{2}} \int_{-\infty}^{-a_1} \int_{\Sigma_{k=1}^{d-1} {x'_k}^2 \leq tan^2(\theta)(x'_d + a_1)^2} e^{-\frac{\sum_{k=1}^{d-1} {x'_k}^2}{2\sigma^2}} dx'_1 \dots dx'_{d-1} e^{-\frac{x'^2_d}{2\sigma^2}} dx'_d \\
&\quad \text{As } a_1 + x'_d \leq 0, (a_1 + x'_d)^2 < (a_2 + x'_d)^2 \\
&< (2\pi\sigma^2)^{-\frac{d}{2}} \int_{-\infty}^{-a_1} \int_{\Sigma_{k=1}^{d-1} {x'_k}^2 \leq tan^2(\theta)(x'_d + a_2)^2} e^{-\frac{\sum_{k=1}^{d-1} {x'_k}^2}{2\sigma^2}} dx'_1 \dots dx'_{d-1} e^{-\frac{x'^2_d}{2\sigma^2}} dx'_d \\
&< (2\pi\sigma^2)^{-\frac{d}{2}} \int_{-\infty}^{-a_2} \int_{\Sigma_{k=1}^{d-1} {x'_k}^2 \leq tan^2(\theta)(x'_d + a_2)^2} e^{-\frac{\sum_{k=1}^{d-1} {x'_k}^2}{2\sigma^2}} dx'_1 \dots dx'_{d-1} e^{-\frac{x'^2_d}{2\sigma^2}} dx'_d \\
&= f_\sigma(x_2)_1.
\end{aligned}$$

□

Lemma A.6. $\forall a > 0, k \geq 1$, $\frac{\Phi(-a)}{\Phi(-ka)} \geq e^{\frac{(k^2-1)a^2}{2}}$.

Proof. Consider the function $h(x) = \frac{\sqrt{2\pi}\Phi(-x)}{e^{-x^2/2}}$ and we will show in the following that it is strictly decreasing for $x > 0$. Alternatively, we take the derivative *w.r.t.* x,

$$\frac{d}{dx} h(x) = \frac{\sqrt{2\pi} x \Phi(-x)}{e^{-x^2/2}} - 1,$$

and show that it is negative for $x > 0$. Since $e^{-x^2/2} > 0$, it is sufficient to show that $\sqrt{2\pi}x\Phi(-x) - e^{-x^2/2} < 0$. Combining that 1) $\sqrt{2\pi}x\Phi(-x) - e^{-x^2/2}$ is increasing as

$$\begin{aligned}\frac{d}{dx}\left(x\Phi(-x) - \frac{e^{-x^2/2}}{\sqrt{2\pi}}\right) &= \Phi(-x) - \frac{xe^{-x^2/2}}{\sqrt{2\pi}} - \frac{-xe^{-x^2/2}}{\sqrt{2\pi}} \\ &= \Phi(-x) > 0\end{aligned}$$

and 2) $\sqrt{2\pi}x\Phi(-x) - e^{-x^2/2} \to 0$ when $x \to \infty$, we have that $\sqrt{2\pi}x\Phi(-x) - e^{-x^2/2} < 0$. As $h(x)$ is strictly decreasing we have that for any $a > 0$ and $k > 1$, $ka > a$. Thus,

$$\frac{\sqrt{2\pi}\Phi(-a)}{e^{-a^2/2}} > \frac{\sqrt{2\pi}\Phi(-ka)}{e^{-(ka)^2/2}}.$$

Rearranging the terms gives the inequality. □

A.2 Supporting proofs for Chapter 4

We extend the theorems from [29] to get results for $\mathcal{L}_{\text{NCA}}$. The results we have here apply to $G = g_0$ and g_1. The case when $G = g_2$, $\mathcal{L}_{\text{MIXNCA}}$, and $\mathcal{L}_{\text{IntNaCl}}$ are left as future work.

Bridging the empirical estimator and asymptotic objective. We introduce an intermediate unbiased loss in order to extend our results. Let $h(x, y) = e^{f(x)^\top f(y)}$, then the unbiased loss with multiple positive pairs is given as

$$\widetilde{L}^{M,N}_{\text{Unbiased}}(f) = \mathbb{E}_{\substack{x\sim p \\ x_i^+\sim p_x^+}} \left[\log \frac{\sum_{i=1}^M h(x, x_i^+)}{\sum_{i=1}^M h(x, x_i^+) + M \cdot N \cdot \mathbb{E}_{x^-\sim p_x^-} h(x, x^-)} \right]$$

Then we can define a debiased loss by

$$L^{M,N,n,m}_{\text{Debiased}}(f) = \mathbb{E}_{\substack{x\sim p \\ x_i^+\sim p_x^+ \\ u_i\sim p; v_i\sim p_x^+}} \left[\log \frac{\sum_{i=1}^M h(x, x_i^+)}{\sum_{i=1}^M h(x, x_i^+) + M \cdot N \cdot G(x, \{u_i\}_{i=1}^n, \{v_i\}_{i=1}^m)} \right].$$

Theorem A.1. *For any embedding f and finite N and M, we have*

$$\left| \widetilde{L}^{M,N}_{\text{Unbiased}}(f) - L^{M,N,n,m}_{\text{Debiased}}(f) \right| \leq \frac{e^{3/2}}{\tau^-} \sqrt{\frac{\pi}{2n}} + \frac{e^{3/2}\tau^+}{\tau^-} \sqrt{\frac{\pi}{2m}}.$$

The proof of A.1 is the same as the proof of Theorem 3 in [29] with the help of the following slightly modified version of Lemma A.2 in [29]. Now if we let

$$\begin{aligned} \Delta = \Bigg| &- \log \frac{\sum_{i=1}^M h(x, x_i^+)}{\sum_{i=1}^M h(x, x_i^+) + M \cdot N \cdot G(x, \{u_i\}_{i=1}^n, \{v_i\}_{i=1}^m)} \\ &+ \log \frac{\sum_{i=1}^M h(x, x_i^+)}{\sum_{i=1}^M h(x, x_i^+) + M \cdot N \cdot \mathbb{E}_{x^-\sim p_x^-} h(x, x^-)} \Bigg|, \end{aligned}$$

where $h(x, \bar{x}) = \exp^{f(x)^\top f(\bar{x})}$, then one has the following lemma:

Lemma A.7. *Let x and x^+ in $\mathcal{X}$ be fixed. Further, let $\{u_i\}_{i=1}^n$ and $\{v_i\}_{i=1}^m$ be collections*

of i.i.d. random variables sampled from p and p_x^+ respectively. Then for all $\varepsilon > 0$,

$$\mathbb{P}(\Delta \geq \varepsilon) \leq 2\exp\left(-\frac{n\varepsilon^2(\tau^-)^2}{2e^3}\right) + 2\exp\left(-\frac{m\varepsilon^2(\tau^-/\tau^+)^2}{2e^3}\right).$$

Proof. We first decompose the probability as

$$\begin{aligned}
&\mathbb{P}\Bigg(\Bigg| -\log\frac{\sum_{i=1}^M h(x,x_i^+)}{\sum_{i=1}^M h(x,x_i^+) + M\cdot N\cdot G(x,\{u_i\}_{i=1}^n,\{v_i\}_{i=1}^m)} \\
&+\log\frac{\sum_{i=1}^M h(x,x_i^+)}{\sum_{i=1}^M h(x,x_i^+) + M\cdot N\cdot \mathbb{E}_{x^-\sim p_x^-}h(x,x^-)}\Bigg| \geq \varepsilon\Bigg) \\
&= \mathbb{P}\Bigg(\Bigg|\log\Big\{\sum_{i=1}^M h(x,x_i^+) + M\cdot N\cdot G(x,\{u_i\}_{i=1}^n,\{v_i\}_{i=1}^m)\Big\} \\
&-\log\Big\{\sum_{i=1}^M h(x,x_i^+) + M\cdot N\cdot \mathbb{E}_{x^-\sim p_x^-}h(x,x^-)\Big\}\Bigg| \geq \varepsilon\Bigg) \\
&= \mathbb{P}\Bigg(\log\Big\{\sum_{i=1}^M h(x,x_i^+) + M\cdot N\cdot G(x,\{u_i\}_{i=1}^n,\{v_i\}_{i=1}^m)\Big\} \\
&-\log\Big\{\sum_{i=1}^M h(x,x_i^+) + M\cdot N\cdot \mathbb{E}_{x^-\sim p_x^-}h(x,x^-)\Big\} \geq \varepsilon\Bigg) \\
&+\mathbb{P}\Bigg(-\log\Big\{\sum_{i=1}^M h(x,x_i^+) + M\cdot N\cdot G(x,\{u_i\}_{i=1}^n,\{v_i\}_{i=1}^m)\Big\} \\
&+\log\Big\{\sum_{i=1}^M h(x,x_i^+) + M\cdot N\cdot \mathbb{E}_{x^-\sim p_x^-}h(x,x^-)\Big\} \geq \varepsilon\Bigg),
\end{aligned}$$

where the final equality holds simply because $|X| \geq \varepsilon$ if and only if $X \geq \varepsilon$ or $-X \geq \varepsilon$.

The first term can be bounded as

$$
\begin{aligned}
&\mathbb{P}(\log\big\{\sum_{i=1}^{M} h(x,x_i^+) + M\cdot N\cdot G(x,\{u_i\}_{i=1}^n,\{v_i\}_{i=1}^m)\big\}\\
&-\log\big\{\sum_{i=1}^{M} h(x,x_i^+) + M\cdot N\cdot \mathbb{E}_{x^-\sim p_x^-} h(x,x^-)\big\}\geq\varepsilon)\\
&=\mathbb{P}(\log\frac{\sum_{i=1}^{M} h(x,x_i^+) + M\cdot N\cdot G(x,\{u_i\}_{i=1}^n,\{v_i\}_{i=1}^m)}{\sum_{i=1}^{M} h(x,x_i^+) + M\cdot N\cdot \mathbb{E}_{x^-\sim p_x^-} h(x,x^-)}\geq\varepsilon)\\
&\leq\mathbb{P}(\frac{M\cdot N\cdot G(x,\{u_i\}_{i=1}^n,\{v_i\}_{i=1}^m) - M\cdot N\cdot \mathbb{E}_{x^-\sim p_x^-} h(x,x^-)}{\sum_{i=1}^{M} h(x,x_i^+) + M\cdot N\cdot \mathbb{E}_{x^-\sim p_x^-} h(x,x^-)}\geq\varepsilon)\\
&=\mathbb{P}(G(x,\{u_i\}_{i=1}^n,\{v_i\}_{i=1}^m) - \mathbb{E}_{x^-\sim p_x^-} h(x,x^-)\geq\varepsilon\Big\{\frac{1}{MN}\sum_{i=1}^{M} h(x,x_i^+) + \mathbb{E}_{x^-\sim p_x^-} h(x,x^-)\Big\})\\
&\leq\mathbb{P}(G(x,\{u_i\}_{i=1}^n,\{v_i\}_{i=1}^m) - \mathbb{E}_{x^-\sim p_x^-} h(x,x^-)\geq\varepsilon e^{-1}).
\end{aligned}
\tag{A.1}
$$

The first inequality follows by applying the fact that $\log x\leq x-1$ for $x>0$. The second inequality holds since $\frac{1}{M\cdot N\cdot}\sum_{i=1}^{M} h(x,x_i^+) + \mathbb{E}_{x^-\sim p_x^-} h(x,x^-)\geq e^{-1}$. Next, we move on to bounding the second term, which proceeds similarly, using the same two bounds.

$$
\begin{aligned}
&\mathbb{P}\Big\{-\log\big(\sum_{i=1}^{M} h(x,x_i^+) + M\cdot N\cdot G(x,\{u_i\}_{i=1}^n,\{v_i\}_{i=1}^m)\big\}\\
&+\log\big\{\sum_{i=1}^{M} h(x,x_i^+) + M\cdot N\cdot \mathbb{E}_{x^-\sim p_x^-} h(x,x^-)\big\}\geq\varepsilon)\\
&=\mathbb{P}(\log\frac{\sum_{i=1}^{M} h(x,x_i^+) + M\cdot N\cdot \mathbb{E}_{x^-\sim p_x^-} h(x,x^-)}{\sum_{i=1}^{M} h(x,x_i^+) + M\cdot N\cdot G(x,\{u_i\}_{i=1}^n,\{v_i\}_{i=1}^m)}\geq\varepsilon)\\
&\leq\mathbb{P}(\frac{M\cdot N\cdot \mathbb{E}_{x^-\sim p_x^-} h(x,x^-) - M\cdot N\cdot G(x,\{u_i\}_{i=1}^n,\{v_i\}_{i=1}^m)}{\sum_{i=1}^{M} h(x,x_i^+) + M\cdot N\cdot G(x,\{u_i\}_{i=1}^N,\{v_i\}_{i=1}^M)}\geq\varepsilon)\\
&=\mathbb{P}(\mathbb{E}_{x^-\sim p_x^-} h(x,x^-) - G(x,\{u_i\}_{i=1}^n,\{v_i\}_{i=1}^m)\geq\varepsilon\Big\{\frac{1}{MN}\sum_{i=1}^{M} h(x,x_i^+) + G(x,\{u_i\}_{i=1}^n,\{v_i\}_{i=1}^m)\Big\})\\
&\leq\mathbb{P}(\mathbb{E}_{x^-\sim p_x^-} h(x,x^-) - G(x,\{u_i\}_{i=1}^n,\{v_i\}_{i=1}^m)\geq\varepsilon e^{-1}).
\end{aligned}
\tag{A.2}
$$

Combining equation A.1 and equation A.2, we have

$$
\mathbb{P}(\Delta\geq\varepsilon)\leq\mathbb{P}(\Big|G(x,\{u_i\}_{i=1}^n,\{v_i\}_{i=1}^m) - \mathbb{E}_{x^-\sim p_x^-} h(x,x^-)\Big|\geq\varepsilon e^{-1}).
$$

Lastly, one needs to bound the right hand tail probability. This part of the proof

remains exactly the same as in [29] and is therefore omitted. □

Bridging the asymptotic objective and supervised loss.

Lemma A.8. *For any embedding f, whenever $N \geq K-1$ we have*

$$L_{\text{Sup}}(f) \leq L^{\mu}_{\text{Sup}}(f) \leq \widetilde{L}^{M,N}_{\text{Unbiased}}(f).$$

Proof. We first show that $N = K-1$ gives the smallest loss:

$$\begin{aligned}
\widetilde{L}^{M,N}_{\text{Unbiased}}(f) &= \mathbb{E}_{\substack{x\sim p \\ x_i^+\sim p_x^+}}\left[-\log \frac{\sum_{i=1}^{M} e^{f(x)^T f(x_i^+)}}{\sum_{i=1}^{M} e^{f(x)^T f(x_i^+)} + M\cdot N\mathbb{E}_{x^-\sim p_x^-} e^{f(x)^T f(x^-)}}\right] \\
&\geq \mathbb{E}_{\substack{x\sim p \\ x_i^+\sim p_x^+}}\left[-\log \frac{\sum_{i=1}^{M} e^{f(x)^T f(x_i^+)}}{\sum_{i=1}^{M} e^{f(x)^T f(x_i^+)} + M\cdot (K-1)\mathbb{E}_{x^-\sim p_x^-} e^{f(x)^T f(x^-)}}\right] \\
&= L^{M,K-1}_{\text{Unbiased}}(f)
\end{aligned}$$

To show that $L^{M,K-1}_{\text{Unbiased}}(f)$ is an upper bound on the supervised loss $L_{\text{sup}}(f)$, we additionally introduce a task specific class distribution $\rho_{\mathcal{T}}$ which is a uniform distribution over all the possible K-way classification tasks with classes in $\mathcal{C}$. That is, we consider

all the possible task with K distinct classes $\{c_1, \ldots, c_K\} \subseteq \mathcal{C}$.

$$
\begin{aligned}
&L_{\text{Unbiased}}^{M,K-1}(f) \\
&= \mathop{\mathbb{E}}_{\substack{x\sim p \\ x_i^+\sim p_x^+}} \left[-\log \frac{\sum_{i=1}^M e^{f(x)^T f(x_i^+)}}{\sum_{i=1}^M e^{f(x)^T f(x_i^+)} + M\cdot(K-1)\mathbb{E}_{x^-\sim p_x^-} e^{f(x)^T f(x^-)}} \right] \\
&= \mathbb{E}_{\mathcal{T}\sim\mathcal{D}} \mathop{\mathbb{E}}_{\substack{c\sim\rho_{\mathcal{T}}; x\sim p(\cdot|c) \\ x_i^+\sim p(\cdot|c)}} \left[-\log \frac{\sum_{i=1}^M e^{f(x)^T f(x_i^+)}}{\sum_{i=1}^M e^{f(x)^T f(x_i^+)} + M\cdot(K-1)\mathbb{E}_{\mathcal{T}\sim\mathcal{D}}\mathbb{E}_{\rho_{\mathcal{T}}(c^-|c^-\neq h(x))}\mathbb{E}_{x^-\sim p(\cdot|c^-)} e^{f(x)^T f(x^-)}} \right] \\
&\geq \mathbb{E}_{\mathcal{T}\sim\mathcal{D}} \mathop{\mathbb{E}}_{c\sim\rho_{\mathcal{T}}; x\sim p(\cdot|c)} \left[-\log \frac{\sum_{i=1}^M e^{f(x)^T \mathbb{E}_{x_i^+\sim p(\cdot|c)} f(x_i^+)}}{\sum_{i=1}^M e^{f(x)^T \mathbb{E}_{x_i^+\sim p(\cdot|c)} f(x_i^+)} + M\cdot(K-1)\mathbb{E}_{\mathcal{T}\sim\mathcal{D}}\mathbb{E}_{\rho_{\mathcal{T}}(c^-|c^-\neq h(x))}\mathbb{E}_{x^-\sim p(\cdot|c^-)} e^{f(x)^T f(x^-)}} \right] \\
&\geq \mathbb{E}_{\mathcal{T}\sim\mathcal{D}} \mathop{\mathbb{E}}_{c\sim\rho_{\mathcal{T}}; x\sim p(\cdot|c)} \left[-\log \frac{\sum_{i=1}^M e^{f(x)^T \mathbb{E}_{x_i^+\sim p(\cdot|c)} f(x_i^+)}}{\sum_{i=1}^M e^{f(x)^T \mathbb{E}_{x_i^+\sim p(\cdot|c)} f(x_i^+)} + M\cdot(K-1)\mathbb{E}_{\rho_{\mathcal{T}}(c^-|c^-\neq h(x))}\mathbb{E}_{x^-\sim p(\cdot|c^-)} e^{f(x)^T f(x^-)}} \right] \\
&= \mathbb{E}_{\mathcal{T}\sim\mathcal{D}} \mathop{\mathbb{E}}_{c\sim\rho_{\mathcal{T}}; x\sim p(\cdot|c)} \left[-\log \frac{M e^{f(x)^T \mathbb{E}_{x^+\sim p(\cdot|c)} f(x^+)}}{M e^{f(x)^T \mathbb{E}_{x^+\sim p(\cdot|c)} f(x^+)} + M\cdot(K-1)\mathbb{E}_{\rho_{\mathcal{T}}(c^-|c^-\neq h(x))}\mathbb{E}_{x^-\sim p(\cdot|c^-)} e^{f(x)^T f(x^-)}} \right] \\
&\geq \mathbb{E}_{\mathcal{T}\sim\mathcal{D}} \mathop{\mathbb{E}}_{c\sim\rho_{\mathcal{T}}; x\sim p(\cdot|c)} \left[-\log \frac{e^{f(x)^T \mathbb{E}_{x^+\sim p(\cdot|c)} f(x^+)}}{e^{f(x)^T \mathbb{E}_{x^+\sim p(\cdot|c)} f(x^+)} + (K-1)\mathbb{E}_{\rho_{\mathcal{T}}(c^-|c^-\neq h(x))} e^{f(x)^T \mathbb{E}_{x^-\sim p(\cdot|c^-)} f(x^-)}} \right] \\
&= \mathbb{E}_{\mathcal{T}\sim\mathcal{D}} \mathop{\mathbb{E}}_{c\sim\rho_{\mathcal{T}}; x\sim p(\cdot|c)} \left[-\log \frac{\exp\big(f(x)^T \mu_c\big)}{\exp(f(x)^T \mu_c) + \sum_{c^-\in\mathcal{T}, c^-\neq c} \exp(f(x)^T \mu_{c^-})} \right] \\
&= \mathbb{E}_{\mathcal{T}\sim\mathcal{D}} L_{\text{Sup}}^{\mu}(\mathcal{T}, f) \\
&= \bar{L}_{\text{Sup}}^{\mu}(f),
\end{aligned}
$$

where the three inequalities follow from Jensen's inequality. The first and third inequality shift the expectations $\mathbb{E}_{x^+\sim p_{x,\mathcal{T}}^+}$ and $\mathbb{E}_{x^-\sim p(\cdot|c^-)}$, respectively, via the convexity of the functions and the second moves the expectation $\mathbb{E}_{\mathcal{T}\sim\mathcal{D}}$ out using concavity. Note that $\bar{L}_{\text{Sup}}(f) \leq \bar{L}_{\text{Sup}}^{\mu}(f)$ holds trivially. □

Generalization bounds. We wish to derive a data dependent bound on the downstream supervised generalization error of the debiased contrastive objective. Recall that a sample $(x, \{x_i^+\}_{i=1}^M, \{u_i\}_{i=1}^n, \{v_i\}_{i=1}^m)$ yields loss

$$
-\log\left\{ \frac{\sum_{i=1}^M e^{f(x)^\top f(x_i^+)}}{\sum_{i=1}^M e^{f(x)^\top f(x_i^+)} + M\cdot N\cdot G(x, \{u_i\}_{i=1}^n, \{v_i\}_{i=1}^m)} \right\} = \log\left\{ 1 + M\cdot N \frac{G(x, \{u_i\}_{i=1}^n, \{v_i\}_{i=1}^m)}{\sum_{i=1}^M e^{f(x)^\top f(x_i^+)}} \right\},
$$

which is equal to $\ell\left(\left\{\frac{e^{f(x)^\top f(u_j)}}{\sum_{i=1}^M e^{f(x)^\top f(x_i^+)}}\right\}_{j=1}^n, \left\{\frac{e^{f(x)^\top f(v_j)}}{\sum_{i=1}^M e^{f(x)^\top f(x_i^+)}}\right\}_{j=1}^m\right)$, where we define

$$\ell(\{a_i\}_{i=1}^n, \{b_i\}_{i=1}^m) \coloneqq \log\left\{1 + M \cdot N \max\left(\frac{1}{\tau^-}\frac{1}{n}\sum_{i=1}^n a_i - \frac{\tau^+}{\tau^-}\frac{1}{m}\sum_{i=1}^m b_i, e^{-1}\right)\right\}$$

$$\widehat{L}_{\text{Debiased}}^{M,N,n,m}(f) \coloneqq \frac{1}{T}\sum_{t=1}^T \ell\left(\left\{\frac{e^{f(x_t)^\top f(u_{tj})}}{\sum_{i=1}^M e^{f(x_t)^\top f(x_{ti}^+)}}\right\}_{j=1}^n, \left\{\frac{e^{f(x_t)^\top f(v_{tj})}}{\sum_{i=1}^M e^{f(x_t)^\top f(x_{ti}^+)}}\right\}_{j=1}^m\right)$$

$$\hat{f} \coloneqq \arg\min_{f\in\mathcal{F}} \widehat{L}_{\text{Debiased}}^{M,N,n,m}(f)$$

Theorem 8. *With probability at least $1-\delta$, for all $f \in \mathcal{F}$ and $N \geq K-1$,*

$$L_{\text{Sup}}(\hat{f}) \leq L_{\text{Debiased}}^{M,N,n,m}(f) + \mathcal{O}\left(\frac{1}{\tau^-}\sqrt{\frac{1}{n}} + \frac{\tau^+}{\tau^-}\sqrt{\frac{1}{m}} + \frac{\lambda\mathcal{R}_{\mathcal{S}}(\mathcal{F})}{T} + B\sqrt{\frac{\log\frac{1}{\delta}}{T}}\right),$$

where $\lambda = \frac{1}{M}\sqrt{\frac{1}{{\tau^-}^2}(\frac{m}{n}+1) + {\tau^+}^2(\frac{n}{m}+1)}$ and $B = \log N\left(\frac{1}{\tau^-} + \tau^+\right)$.

Proof. Considering the samples to be $\left\{\left(x_t, \left\{x_{ti}^+\right\}_{i=1}^M, \left\{u_{ti}\right\}_{i=1}^n, \left\{v_{ti}\right\}_{i=1}^m\right)\right\}_{t=1}^T$. Then, we can use the standard bounds for empirical versus population means of any $B-$bounded function g belonging to a function class G, we have that with probability at least $1-\frac{\delta}{2}$.

$$\mathbb{E}[g(x)] \leq \frac{1}{T}\sum_{t=1}^T g(x_i) + \frac{2\mathcal{R}_S(G)}{T} + 3B\sqrt{\frac{\log\left(\frac{4}{\delta}\right)}{2T}}. \tag{A.3}$$

In order to calculate $\mathcal{R}_S(G)$ we use the same trick as in [158]. We express it as a composition of functions $g = \ell\left(h\left(f\left(x_t, \left\{x_{ti}^+\right\}_{i=1}^M, \left\{u_{ti}\right\}_{i=1}^n, \left\{v_{ti}\right\}_{i=1}^m\right)\right)\right)$, where $f \in \mathcal{F}$ just maps each sample to corresponding feature vector and h maps the feature vectors to the $\{a\}_{i=1}^n, \{b\}_{i=1}^m$. Then we use contraction inequality to bound $\mathcal{R}_S(G)$ with $\mathcal{R}_S(\mathcal{F})$. In order to do this we need to compute the Lipschitz constant for the intermediate function h in the composition.

For h, we see that the Jacobian has the following form

$$\frac{\partial a_i}{\partial f(x)} = a_i \frac{\sum_{j=1}^{M}(f(u_i) - f(x_j))e^{f(x)^\top f(x_j^+)}}{\sum_{j=1}^{M} e^{f(x)^\top f(x_j^+)}}; \quad \frac{\partial b_i}{\partial f(x)} = b_i \frac{\sum_{j=1}^{M}(f(v_i) - f(x_j))e^{f(x)^\top f(x_j^+)}}{\sum_{j=1}^{M} e^{f(x)^\top f(x_j^+)}}$$

$$\frac{\partial a_i}{\partial f(x_j^+)} = -a_i \frac{f(x)e^{f(x)^\top f(x_j^+)}}{\sum_{k=1}^{M} e^{f(x)^\top f(x_k^+)}}; \quad \frac{\partial b_i}{\partial f(x_j^+)} = -b_i \frac{f(x)e^{f(x)^\top f(x_j^+)}}{\sum_{k=1}^{M} e^{f(x)^\top f(x_k^+)}}$$

$$\frac{\partial a_i}{\partial f(u_j)} = f(x)a_i\delta(i-j); \quad \frac{\partial b_i}{\partial f(v_j)} = f(x)b_i\delta(i-j).$$

Using the fact that $\|f(\cdot)\|_2 = 1$, we get $\frac{e^{-2}}{M} \leq a_i, b_i \leq \frac{e^2}{M}$ and

$$\begin{aligned}
\|J\|_2^2 \leq \|J\|_F^2 \leq \sum_{i=1}^{n} a_i^2 &\left(\left\| \frac{\sum_{j=1}^{M}(f(u_i) - f(x_j))e^{f(x)^\top f(x_j^+)}}{\sum_{j=1}^{M} e^{f(x)^\top f(x_j^+)}} \right\|_2^2 + \|f(x)\|_2^2 \frac{\sum_{j=1}^{M} e^{2f(x)^\top f(x_j^+)}}{\left(\sum_{j=1}^{M} e^{f(x)^\top f(x_j^+)}\right)^2} + \|f(x)\|_2^2 \right) \\
+ \sum_{i=1}^{m} b_i^2 &\left(\left\| \frac{\sum_{j=1}^{M}(f(v_i) - f(x_j))e^{f(x)^\top f(x_j^+)}}{\sum_{j=1}^{M} e^{f(x)^\top f(x_j^+)}} \right\|_2^2 + \|f(x)\|_2^2 \frac{\sum_{j=1}^{M} e^{2f(x)^\top f(x_j^+)}}{\left(\sum_{j=1}^{M} e^{f(x)^\top f(x_j^+)}\right)^2} + \|f(x)\|_2^2 \right) \\
&\leq \sum_{i=1}^{n} a_i^2 (4+1+1) + \sum_{i=1}^{m} b_i^2 (4+1+1) \leq \frac{6(n+m)e^4}{M^2}.
\end{aligned}$$

Using this and the Lipschitz constant, $O\left(\sqrt{\frac{1}{n\tau^{-2}} + \frac{\tau^{+2}}{m}}\right)$ of ℓ derived in [29], we get $\mathcal{R}_S(\mathcal{G}) = \lambda \mathcal{R}_S(\mathcal{F})$ where $\lambda = \mathcal{O}\left(\frac{1}{M}\sqrt{\frac{1}{\tau^{-2}}(\frac{m}{n}+1) + \tau^{+2}(\frac{n}{m}+1)}\right)$. From [29], we also get $B = O\left(\log N\left(\frac{1}{\tau^-} + \tau^+\right)\right)$. Combining this with equation A.3 gives us that with probability at least $1 - \frac{\delta}{2}$

$$L_{\text{Debiased}}^{M,N,n,m}(\hat{f}) \leq \widehat{L}_{\text{Debiased}}^{M,N,n,m}(\hat{f}) + \mathcal{O}\left(\frac{\lambda \mathcal{R}_{\mathcal{S}}(\mathcal{F})}{T} + B\sqrt{\frac{\log \frac{1}{\delta}}{T}}\right).$$

Using Theorem A.7, we get that

$$\begin{aligned}
L_{\text{Unbiased}}^{M,N}(\hat{f}) &\leq L_{\text{Debiased}}^{M,N,n,m}(\hat{f}) + \mathcal{O}\left(\frac{1}{\tau^-}\sqrt{\frac{1}{n}} + \frac{\tau^+}{\tau^-}\sqrt{\frac{1}{m}}\right) \\
&\leq \widehat{L}_{\text{Debiased}}^{M,N,n,m}(\hat{f}) + \mathcal{O}\left(\frac{1}{\tau^-}\sqrt{\frac{1}{n}} + \frac{\tau^+}{\tau^-}\sqrt{\frac{1}{m}} + \frac{\lambda \mathcal{R}_{\mathcal{S}}(\mathcal{F})}{T} + B\sqrt{\frac{\log \frac{1}{\delta}}{T}}\right).
\end{aligned}$$

Using Lemma A.8, we get

$$L_{\text{Sup}}(\hat{f}) \leq L^{M,N}_{\text{Unbiased}}(\hat{f}) \leq \widehat{L}^{M,N,n,m}_{\text{Debiased}}(\hat{f}) + \mathcal{O}\left(\frac{1}{\tau^-}\sqrt{\frac{1}{n}} + \frac{\tau^+}{\tau^-}\sqrt{\frac{1}{m}} + \frac{\lambda\mathcal{R}_{\mathcal{S}}(\mathcal{F})}{T} + B\sqrt{\frac{\log\frac{1}{\delta}}{T}}\right).$$

Finally we see that for any f, we can use M Hoeffding's inequality to show that with at least $1 - \frac{\delta}{2}$ probability

$$\widehat{L}^{M,N,n,m}_{\text{Debiased}}(f) \leq L^{M,N,n,m}_{\text{Debiased}}(f) + 3B\sqrt{\frac{\log\left(\frac{2}{\delta}\right)}{2T}}.$$

Combining all of the above results gives us that with probability at least $1 - \delta$,

$$\begin{aligned}
L_{\text{Sup}}(\hat{f}) &\leq L^{M,N}_{\text{Unbiased}}(\hat{f}) \leq \widehat{L}^{M,N,n,m}_{\text{Debiased}}(\hat{f}) + \mathcal{O}\left(\frac{1}{\tau^-}\sqrt{\frac{1}{n}} + \frac{\tau^+}{\tau^-}\sqrt{\frac{1}{m}} + \frac{\lambda\mathcal{R}_{\mathcal{S}}(\mathcal{F})}{T} + B\sqrt{\frac{\log\frac{1}{\delta}}{T}}\right) \\
&\leq \widehat{L}^{M,N,n,m}_{\text{Debiased}}(f) + \mathcal{O}\left(\frac{1}{\tau^-}\sqrt{\frac{1}{n}} + \frac{\tau^+}{\tau^-}\sqrt{\frac{1}{m}} + \frac{\lambda\mathcal{R}_{\mathcal{S}}(\mathcal{F})}{T} + B\sqrt{\frac{\log\frac{1}{\delta}}{T}}\right) \\
&\leq L^{M,N,n,m}_{\text{Debiased}}(f) + \mathcal{O}\left(\frac{1}{\tau^-}\sqrt{\frac{1}{n}} + \frac{\tau^+}{\tau^-}\sqrt{\frac{1}{m}} + \frac{\lambda\mathcal{R}_{\mathcal{S}}(\mathcal{F})}{T} + B\sqrt{\frac{\log\frac{1}{\delta}}{T}}\right) + \mathcal{O}\left(B\sqrt{\frac{\log\left(\frac{1}{\delta}\right)}{T}}\right).
\end{aligned}$$

□

A.3 Supporting proofs for Chapter 5

A.3.1 General ℓ_p results

We note that our ℓ_2 results can be straightforwardly generalized to ℓ_p. Given an ℓ_p adversarial budget ϵ:

Theorem A.2. *For any sample x, the optimal robust classifier f_ϵ for $P_{\mu_1,\mu_2,\Sigma}$ gives*

(i) the bound (decision margin) $\|\Delta\|_p = \frac{|(x-\frac{\mu_1+\mu_2}{2})^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))|}{\|\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))\|_q}$,

(ii) the scaled bound $\|\bar{\Delta}\|_p = \frac{|(x-\frac{\mu_1+\mu_2}{2})^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))|}{|\tilde{\mu}^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))|}$.

For sample $x \sim P_{\mu_1,\mu_2,\Sigma}$, it further gives

(iii) the standard accuracy $a = \Phi(\frac{\tilde{\mu}^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))}{\|\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))\|_\Sigma})$,

(iv) the expected scaled bound of correct samples
$\mathbb{E}\left[\|\bar{\Delta}\|_p \mid f_\epsilon(x) = y\right] = \frac{1}{\sqrt{2\pi}}\frac{1}{a\Phi^{-1}(a)}e^{-\frac{1}{2}\left(\Phi^{-1}(a)\right)^2} + 1$,

where z_Σ is the solution of the convex problem $\arg\min_{\|z\|_p\leq\epsilon}(\tilde{\mu}-z)^T\Sigma^{-1}(\tilde{\mu}-z)$ *and Φ denotes the CDF of the standard normal distribution.*

Proof. We follow the proof of Theorem 9 and consider the classifier in equation 5.2. By Hölder's inequality, we now have the corresponding lower bound and scaled lower bound as

$$\begin{aligned}
\|\Delta\|_p &= \frac{|(x-\frac{\mu_1+\mu_2}{2})^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))|}{\|\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))\|_q} \\
\|\bar{\Delta}\|_p &= \frac{|(x-\frac{\mu_1+\mu_2}{2})^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))|}{\|\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))\|_q}\frac{\|\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))\|_q}{|\tilde{\mu}^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))|} \\
&= \frac{|(x-\frac{\mu_1+\mu_2}{2})^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))|}{|\tilde{\mu}^T\Sigma^{-1}(\tilde{\mu}-z_\Sigma(\tilde{\mu}))|},
\end{aligned}$$

where $\frac{1}{p}+\frac{1}{q}=1$. The remainder of the proof will then follow as in Theorem 9. □

Remark. In general, in the case that Σ is singular, we can apply the economy-size (thin) decomposition with nonzero eigenvalues $\Sigma = F\Lambda F^T$. Then, with a general

non-symmetric conditional Gaussians

$$x|y=1 \sim \mathcal{N}(\mu_1, \Sigma), \quad x|y=-1 \sim \mathcal{N}(\mu_2, \Sigma),$$

we apply proper translation to symmetric conditional Gaussians

$$F^T x|y=1 \sim \mathcal{N}(F^T\mu_1, \Lambda), \quad F^T x|y=-1 \sim \mathcal{N}(F^T\mu_2, \Lambda),$$
$$F^T x - F^T\frac{\mu_1+\mu_2}{2}|y=1 \sim \mathcal{N}(\tilde{\mu}, \Lambda), \quad F^T x - F^T\frac{\mu_1+\mu_2}{2}|y=-1 \sim \mathcal{N}(-\tilde{\mu}, \Lambda),$$

where $\tilde{\mu} = F^T\frac{\mu_1-\mu_2}{2}$.

A.3.2 Class imbalance results

Given an ℓ_2 adversarial budget $\epsilon \leq \|\mu\|_2$, consider the conditional Gaussian in equation 5.1 with $\Sigma = I_d$ (d by d identity matrix) and general class prior τ, then the following theorem holds.

Theorem A.3. *For any sample x, the optimal robust classifier f_ϵ for P_{μ_1,μ_2,I_d} gives*

(i) the bound (decision margin) $\|\Delta\|_2 = \frac{|(x-\frac{\mu_1+\mu_2}{2})^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)-q/2|}{\|\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)\|_2}$,

(ii) the scaled bound $\|\bar{\Delta}\|_2 = \frac{2|(x-\frac{\mu_1+\mu_2}{2})^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)-q/2|}{|\tilde{\mu}^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)-q/2|+|\tilde{\mu}^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)+q/2|}$.

For a sample $x \sim P_{\mu_1,\mu_2,I_d}$, it further gives

(iii) the standard accuracy $a = \tau\Phi(\frac{\tilde{\mu}^T w-q/2}{\|w\|_2}) + (1-\tau)\Phi(\frac{\tilde{\mu}^T w+q/2}{\|w\|_2})$,

(iv) the expected scaled bound of correct samples

$$\mathbb{E}\left[\|\bar{\Delta}\|_2 \mid f_\epsilon(x)=y\right] = \frac{2\tau}{|\tilde{\mu}^T w-q/2| + |\tilde{\mu}^T w+q/2|}\left(\tilde{\mu}^T w - q/2 + \|w\|_2\frac{\phi(\frac{-\tilde{\mu}^T w+q/2}{\|w\|_2})}{\Phi(\frac{\tilde{\mu}^T w-q/2}{\|w\|_2})}\right)$$
$$+ \frac{2(1-\tau)}{|\tilde{\mu}^T w-q/2| + |\tilde{\mu}^T w+q/2|}\left(\tilde{\mu}^T w + q/2 + \|w\|_2\frac{\phi(\frac{\tilde{\mu}^T w+q/2}{\|w\|_2})}{\Phi(\frac{\tilde{\mu}^T w+q/2}{\|w\|_2})}\right).$$

where $q = ln\{(1-\tau)/\tau\}$, $w = \tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)$, ϕ and Φ denotes the PDF and CDF of the standard normal distribution.

Proof. (i) Consider the Bayes optimal ℓ_2 ϵ-robust classifier [39, Theorem 4.1]

$$f_\epsilon(x) = sign\left\{\left(x - \frac{\mu_1+\mu_2}{2}\right)^T \tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) - q/2\right\}, \tag{A.4}$$

where $q = ln\{(1-\tau)/\tau\}$. For any x,

$$\|\Delta\|_2 = \frac{|(x - \frac{\mu_1+\mu_2}{2})^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) - q/2|}{\|\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)\|_2}.$$

(ii) Since the bound $\|\Delta\|_2$ is subject to the positions of two Gaussians, we scale the bound by the distance from Gaussian centers to the classifier. We note that now the distances from the two Gaussian centers to the classifier are different, $\frac{|\tilde{\mu}^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)-q/2|}{\|\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)\|_2}$ and $\frac{|\tilde{\mu}^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)+q/2|}{\|\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)\|_2}$. We hereby take their average as the scaling factor and obtain

$$\begin{aligned}\|\bar{\Delta}\|_2 &= \frac{|(x - \frac{\mu_1+\mu_2}{2})^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) - q/2|}{\|\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)\|_2}\frac{2\|\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)\|_2}{|\tilde{\mu}^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) - q/2| + |\tilde{\mu}^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) + q/2|} \\ &= \frac{2|(x - \frac{\mu_1+\mu_2}{2})^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) - q/2|}{|\tilde{\mu}^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) - q/2| + |\tilde{\mu}^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2) + q/2|}.\end{aligned}$$

(iii) For sample $x \sim P_{\mu_1,\mu_2,I_d}$, consider the Bayes optimal robust classifier in equa-

tion 5.2, we can calculate the analytical standard accuracy by

$$
\begin{aligned}
&\mathbb{P}(y=1)\mathbb{P}\left[f_\epsilon(x)=1 \mid y=1\right]+\mathbb{P}(y=-1)\mathbb{P}\left[f_\epsilon(x)=-1 \mid y=-1\right] \\
=&\tau\mathbb{P}\left[f_\epsilon(x)=1 \mid y=1\right]+(1-\tau)\left[f_\epsilon(x)=-1 \mid y=-1\right] \\
=&\tau\mathbb{P}\left[(x-\frac{\mu_1+\mu_2}{2})^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)-q/2>0 \mid y=1\right] \\
+&(1-\tau)\mathbb{P}\left[(x-\frac{\mu_1+\mu_2}{2})^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)-q/2<0 \mid y=-1\right] \\
=&\tau\mathbb{P}\left[(\tilde{\mu}+w)^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)-q/2>0\right], \\
+&(1-\tau)\mathbb{P}\left[(-\tilde{\mu}+w)^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)-q/2<0\right], \quad w\sim\mathcal{N}(0,I_d) \\
=&\tau\mathbb{P}\left[w^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)>q/2-\tilde{\mu}^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)\right], \\
+&(1-\tau)\mathbb{P}\left[w^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)<q/2+\tilde{\mu}^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)\right], \quad w\sim\mathcal{N}(0,I_d) \\
=&\tau\mathbb{P}\left[\frac{w^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)}{\|\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)\|_2}>\frac{q/2-\tilde{\mu}^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)}{\|\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)\|_2}\right], \\
+&(1-\tau)\mathbb{P}\left[\frac{w^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)}{\|\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)\|_2}<\frac{q/2+\tilde{\mu}^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)}{\|\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)\|_2}\right], \quad \frac{w^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)}{\|\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)\|_2}\sim\mathcal{N}(0,1) \\
=&\tau\Phi(\frac{\tilde{\mu}^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)-q/2}{\|\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)\|_2})+(1-\tau)\Phi(\frac{\tilde{\mu}^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)+q/2}{\|\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)\|_2}).
\end{aligned}
$$

Let w denote $\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)$, the we got the accuracy

$$a=\tau\Phi(\frac{\tilde{\mu}^Tw-q/2}{\|w\|_2})+(1-\tau)\Phi(\frac{\tilde{\mu}^Tw+q/2}{\|w\|_2}).$$

(iv) For sample $x\sim P_{\mu_1,\mu_2,I_d}$, let t denote $x-\frac{\mu_1+\mu_2}{2}$, and w denote $\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)$. According to Theorem A.3(iii), when $\tilde{\mu}^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)-q/2>0$, the accuracy would be higher than 0.5. Therefore we consider $\tilde{\mu}^Tw-q/2>0$.

Now consider the classifier in equation A.4 and the corresponding scaled bound from (ii),

$$\|\bar{\Delta}\|_2=\frac{2|(x-\frac{\mu_1+\mu_2}{2})^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)-q/2|}{|\tilde{\mu}^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)-q/2|+|\tilde{\mu}^T\tilde{\mu}(1-\epsilon/\|\tilde{\mu}\|_2)+q/2|}=\frac{2|t^Tw-q/2|}{|\tilde{\mu}^Tw-q/2|+|\tilde{\mu}^Tw+q/2|}.$$

Since $t|y\sim\mathcal{N}(y\tilde{\mu},I_d)$, we have $t^Tw-q/2|y\sim\mathcal{N}(y\tilde{\mu}^Tw-q/2,w^Tw)$. When we only

want to get the expected scaled bound of the correctly-classified samples, we have that

$$
\begin{aligned}
&\mathbb{E}\left[\|\bar{\Delta}\|_2 \mid f_\epsilon(x)=y\right] \\
&= \frac{2}{|\tilde{\mu}^T w - q/2| + |\tilde{\mu}^T w + q/2|}\mathbb{E}\left[|t^T w - q/2| \mid f_\epsilon(x)=y\right] \\
&= \frac{\tau\Phi(\frac{\tilde{\mu}^T w - q/2}{\|w\|_2})}{\tau\Phi(\frac{\tilde{\mu}^T w - q/2}{\|w\|_2}) + (1-\tau)\Phi(\frac{\tilde{\mu}^T w + q/2}{\|w\|_2})}\frac{2}{|\tilde{\mu}^T w - q/2| + |\tilde{\mu}^T w + q/2|}\mathbb{E}\left[|t^T w - q/2| \mid f_\epsilon(x)=y=1\right] \\
&+ \frac{(1-\tau)\Phi(\frac{\tilde{\mu}^T w + q/2}{\|w\|_2})}{\tau\Phi(\frac{\tilde{\mu}^T w - q/2}{\|w\|_2}) + (1-\tau)\Phi(\frac{\tilde{\mu}^T w + q/2}{\|w\|_2})}\frac{2}{|\tilde{\mu}^T w - q/2| + |\tilde{\mu}^T w + q/2|}\mathbb{E}\left[|t^T w - q/2| \mid f_\epsilon(x)=y=-1\right] \\
&= \frac{\tau\Phi(\frac{\tilde{\mu}^T w - q/2}{\|w\|_2})}{\tau\Phi(\frac{\tilde{\mu}^T w - q/2}{\|w\|_2}) + (1-\tau)\Phi(\frac{\tilde{\mu}^T w + q/2}{\|w\|_2})}\frac{2}{|\tilde{\mu}^T w - q/2| + |\tilde{\mu}^T w + q/2|}\mathbb{E}\left[t^T w - q/2 \mid y=1, t^T w - q/2 \geq 0\right] \\
&+ \frac{(1-\tau)\Phi(\frac{\tilde{\mu}^T w + q/2}{\|w\|_2})}{\tau\Phi(\frac{\tilde{\mu}^T w - q/2}{\|w\|_2}) + (1-\tau)\Phi(\frac{\tilde{\mu}^T w + q/2}{\|w\|_2})}\frac{2}{|\tilde{\mu}^T w - q/2| + |\tilde{\mu}^T w + q/2|}\mathbb{E}\left[-t^T w + q/2 \mid y=-1, t^T w - q/2 < 0\right].
\end{aligned}
$$

Recall that $t^T w - q/2 | y \sim \mathcal{N}(y\tilde{\mu}^T w - q/2, w^T w)$, then by the mean of truncated normal distribution, it is true that

$$
\begin{aligned}
\mathbb{E}\left[t^T w - q/2 \mid y=1, t^T w - q/2 \geq 0\right] &= \tilde{\mu}^T w - q/2 + \|w\|_2 \frac{\phi(\frac{0-\tilde{\mu}^T w + q/2}{\|w\|_2})}{1-\Phi(\frac{0-\tilde{\mu}^T w + q/2}{\|w\|_2})} \\
&= \tilde{\mu}^T w - q/2 + \|w\|_2 \frac{\phi(\frac{-\tilde{\mu}^T w + q/2}{\|w\|_2})}{\Phi(\frac{\tilde{\mu}^T w - q/2}{\|w\|_2})} \\
\mathbb{E}\left[-t^T w + q/2 \mid y=-1, t^T w - q/2 < 0\right] &= -\mathbb{E}\left[t^T w - q/2 \mid y=-1, t^T w - q/2 < 0\right] \\
&= -\left(-\tilde{\mu}^T w - q/2 - \|w\|_2 \frac{\phi(\frac{0+\tilde{\mu}^T w + q/2}{\|w\|_2})}{\Phi(\frac{0+\tilde{\mu}^T w + q/2}{\|w\|_2})}\right) \\
&= \tilde{\mu}^T w + q/2 + \|w\|_2 \frac{\phi(\frac{\tilde{\mu}^T w + q/2}{\|w\|_2})}{\Phi(\frac{\tilde{\mu}^T w + q/2}{\|w\|_2})}.
\end{aligned}
$$

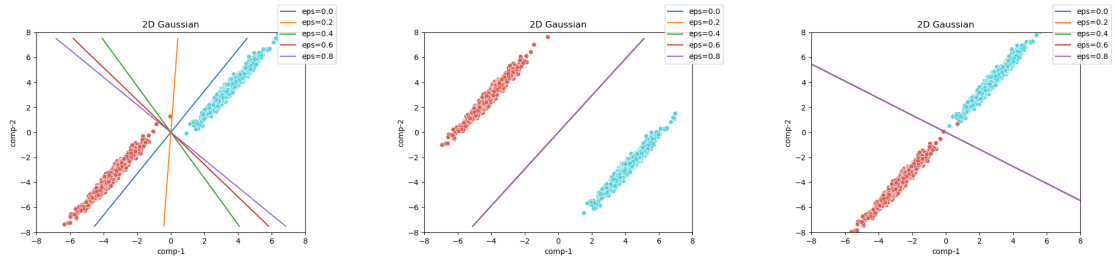
Therefore

$$
\begin{aligned}
&\mathbb{E}\left[\|\bar{\Delta}\|_2 \mid f_\epsilon(x)=y\right] \\
&= \frac{\tau\Phi(\frac{\tilde{\mu}^T w - q/2}{\|w\|_2})}{\tau\Phi(\frac{\tilde{\mu}^T w - q/2}{\|w\|_2}) + (1-\tau)\Phi(\frac{\tilde{\mu}^T w + q/2}{\|w\|_2})}\frac{2}{|\tilde{\mu}^T w - q/2| + |\tilde{\mu}^T w + q/2|}\left(\tilde{\mu}^T w - q/2 + \|w\|_2 \frac{\phi(\frac{-\tilde{\mu}^T w + q/2}{\|w\|_2})}{\Phi(\frac{\tilde{\mu}^T w - q/2}{\|w\|_2})}\right) \\
&+ \frac{(1-\tau)\Phi(\frac{\tilde{\mu}^T w + q/2}{\|w\|_2})}{\tau\Phi(\frac{\tilde{\mu}^T w - q/2}{\|w\|_2}) + (1-\tau)\Phi(\frac{\tilde{\mu}^T w + q/2}{\|w\|_2})}\frac{2}{|\tilde{\mu}^T w - q/2| + |\tilde{\mu}^T w + q/2|}\left(\tilde{\mu}^T w + q/2 + \|w\|_2 \frac{\phi(\frac{\tilde{\mu}^T w + q/2}{\|w\|_2})}{\Phi(\frac{\tilde{\mu}^T w + q/2}{\|w\|_2})}\right)
\end{aligned}
$$

□

A.4 Supporting proofs for Chapter 6

To motivate our findings, we first plot the Bayes optimal robust classifiers together with the Bayes optimal classifier in three 2D cases in Figure A-1. From the plot, we see that as long as the direction of μ is in parallel to one of the two eigenvectors, the robust Bayes optimal classifiers would overlap with the Bayes optimal classifier.



(a) No alignment (b) $\mu \parallel v_1$ (c) $\mu \parallel v_2$

Figure A-1: Three 2D examples of the Bayes optimal classifier and robust Bayes optimal classifiers with different magnitudes of expected perturbation ϵ. Figure A-1(a) - no alignment between the mean vector μ and the eigenvectors. Figure A-1(b) and Figure A-1(c) - μ is parallel to the eigenvector corresponding to either of the two eigenvalues.

To generalize the result, we prove the following theorem that specifies a sufficient condition for all ϵ-robust Bayes optimal classifiers to overlap with each other (including $\epsilon = 0$, *i.e.* Bayes optimal classifier). Intuitively, if the ϵ-robust Bayes optimal classifiers overlap with the Bayes optimal classifiers, then there is no robustness-accuracy trade-off.

Theorem A.4. *The ϵ-robust Bayes optimal classifiers overlap for all ϵ if the vector difference μ between the centers of the two gaussians lies completely within a degenerate subspace of the eigenspace of the covariance matrix, i.e. with eigenpairs $\{(\lambda_k, v_k), k \in [n]\}$, for $\forall\ i, j \in \{k : \lambda_k \neq 0, \mu^T v_k \neq 0\}$, $\lambda_i = \lambda_j = \lambda$.*

Proof. Let $v_1, \ldots, v_n$ and $\lambda_1, \ldots, \lambda_n$ be the orthonormal eigenbasis and the corresponding eigenvalues of the covariance matrix Σ, then we have $\Sigma^{-1} = \sum_{i=1}^{n} \frac{1}{\lambda_i} v_i v_i^T$. Following [36], we see that the ϵ-robust classifier is given as $\operatorname{sign} w^{\epsilon\top} x$, where $w^\epsilon =$

$\Sigma^{-1}\left(\mu - z_\Sigma^\epsilon(\mu)\right)$ and

$$z_\Sigma^\epsilon(\mu) = \arg\min_{\|z\| \le \epsilon} \|\mu - z\|_{\Sigma^{-1}}^2.$$

Let $\mu = \sum_{i=1}^n a_i v_i$ and we re-parameterize $z = \sum_{i=1}^n b_i v_i$. Then,

$$z_\Sigma^\epsilon(\mu) = \sum_{i=1}^n b_i^\epsilon v_i, \quad \text{where } b^\epsilon = \langle b_i^\epsilon \rangle_{i=1}^n = \arg\min_{\sum_{i=1}^n b_i^2 \le \epsilon^2} \sum_{i=1}^n \frac{(a_i - b_i)^2}{\lambda_i}$$

By using the Lagrange multiplier γ_ϵ with first-order optimality condition, we see that $\forall\ i$

$$\frac{b_i^\epsilon - a_i}{\lambda_i} + \gamma_\epsilon b_i^\epsilon = 0 \iff \frac{a_i - b_i^\epsilon}{\lambda_i} = \gamma_\epsilon b_i^\epsilon \iff b_i^\epsilon = \frac{a_i}{1 + \lambda_i \gamma_\epsilon} \tag{A.5}$$

and $\sum_{i=1}^n (b_i^\epsilon)^2 \le \epsilon^2$. In order for all the robust classifiers to overlap we need $w^\epsilon / \|w^\epsilon\|$ to the independent of ϵ. That is,

$$\frac{w^\epsilon}{\|w^\epsilon\|} = \frac{\sum_{i=1}^n v_i \frac{a_i - b_i^\epsilon}{\lambda_i}}{\sqrt{\sum_{i=1}^n \left(\frac{a_i - b_i^\epsilon}{\lambda_i}\right)^2}} = \frac{\sum_{i=1}^n \gamma^\epsilon b_i^\epsilon v_i}{\sqrt{\sum_{i=1}^n (\gamma^\epsilon)^2 (b_i^\epsilon)^2}} = \frac{\sum_{i=1}^n b_i^\epsilon v_i}{\sqrt{\sum_{i=1}^n (b_i^\epsilon)^2}} = \frac{\sum_{i \in S} b_i^\epsilon v_i}{\sqrt{\sum_{i \in S}^n (b_i^\epsilon)^2}},$$

where the S in the last equation denotes the set of indices for which $a_i \neq 0$. For $\forall\ i$ with $a_i = 0$, from equation A.5, we clearly have $b_i^\epsilon = 0$.

The condition μ lies completely within a degenerate subspace of the eigenspace of Σ is equivalent to saying $\lambda_i = \lambda_j = \lambda$ for $\forall\ i, j \in S$. In this case, we see that for $\forall\ i \in S$,

$$\epsilon^2 \ge \sum_{i=1}^n (b_i^\epsilon)^2 = \sum_{i \in S} (b_i^\epsilon)^2 = \left(\frac{1}{1 + \lambda \gamma_\epsilon}\right)^2 \sum_{i \in S} a_i^2,$$

so $\frac{1}{1+\lambda\gamma_\epsilon} \le \epsilon \frac{1}{\sqrt{\sum_{i \in S} a_i^2}}$, $b_i^\epsilon \le \frac{\epsilon}{\sqrt{\sum_{i \in S} a_i^2}} a_i$. So, we get $b_i^\epsilon = m_\epsilon \cdot a_i$ where $m_\epsilon =$

$\min\left(1, \frac{\epsilon}{\sqrt{\sum_{i\in S} a_i^2}}\right)$

$$\frac{w^\epsilon}{\|w^\epsilon\|} = \frac{\sum_{i\in S} b_i^\epsilon v_i}{\sqrt{\sum_{i\in S}^n (b_i^\epsilon)^2}} = \frac{\sum_{i\in S} m_\epsilon a_i v_i}{m_\epsilon \sqrt{\sum_{i\in S} a_i^2}} = \sum_{i\in S} \frac{a_i}{\sqrt{\sum_{i\in S}(a_i)^2}} v_i,$$

which is independent of ϵ. □

Appendix B

Analysis

B.1 Complete analysis in Chapter 3

B.1.1 Shrinking effect for unidimensional data

Bounded decision region. Without loss of generality, let the decision region be interval $\mathcal{D} = [-R, R]$. By the symmetric nature of Gaussian smoothing, we see that $\mathcal{D}_\sigma$ is also an interval of the form $[-a, a]$. We claim that for large σ, $a < R$ and for even larger σ, $\mathcal{D}_\sigma$ disappears. Formally, we do the analysis as follows.

For the shrinking, we check the value of $f_\sigma(R)_1$. By definition A.1.2, we see that $f_\sigma(R)_1 = \Phi(\frac{2R}{\sigma}) - \Phi(0)$ and if

$$\sigma > \frac{2R}{\Phi^{-1}(\frac{1}{2} + \frac{1}{c})},$$

$f_\sigma(R) < \frac{1}{c}$ is true. Thus, the bounded decision region of unidimensional data shrinks with smoothing factor $\sigma > \frac{2R}{\Phi^{-1}(\frac{1}{c}+\frac{1}{2})}$.

For the vanishing rate, we check the value of $f_\sigma(x)_1$ at $x = 0$. Now since $f_\sigma(0)_1 = \Phi(\frac{R}{\sigma}) - \Phi(-\frac{R}{\sigma})$, we have that if

$$\sigma > \frac{R}{\Phi^{-1}(\frac{1}{2} + \frac{1}{2c})},$$

$f_\sigma(0)_1 < \frac{1}{c}$ is true, *i.e.*, $\mathcal{D}_\sigma$ vanishes.

Semi-bounded decision region. In a unidimensional case, our definition of semi-bounded regions degenerates into an interval I of the form $[a, \infty)$. In this case, Theorem 7 gives a trivial bound of 0 for the shrinkage of the decision region, suggesting that no shrinking happens. However, we emphasize that in practice, shrinking might still happens and more detailed analysis is left for future work.

B.1.2 Bounded decision region behaviors

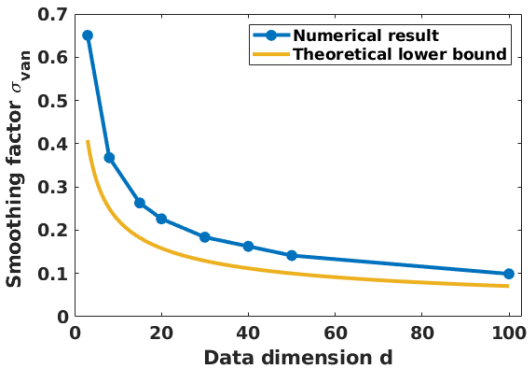


Figure B-1: The vanishing smoothing factor σ_{van} with an increasing input-space dimension in the exemplary adversarial ball.

The vanishing smoothing factors σ_{van} with different data dimensions implied by Figure 2 of the main text together with the theoretical lower bound found in Theorem 4 is given as Figure B-1.

Figure B-2 shows the certified radius behavior as a function of the distance of points from the origin (y-axis) and the smoothing factor σ (x-axis) for dimension $d = 30$. The contour lines in Figure B-2 mark the certified radius of points under Gaussian smoothing. It is notable that points closer to the origin generally have larger certified radii and the certified radius of the point at the origin (y-axis $y = 0$) drops to zero at vanishing smoothing factor $\sigma_{\text{van}} = 0.184$ as specified in Figure B-1. Specifically, one can readily verify that the certified radii of points closer to the origin increase with the growing smoothing factor σ but begin to decrease at certain point, which is coherent with our observations through Figure 3(a) of the main text. Conducting similar experiments for different dimensions completes the maximum certified radius

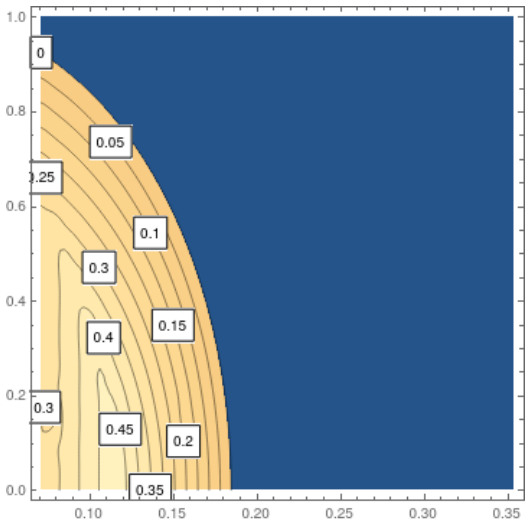


Figure B-2: The certified radius of smoothed classifiers with an increasing input-space dimension when $d = 30$.

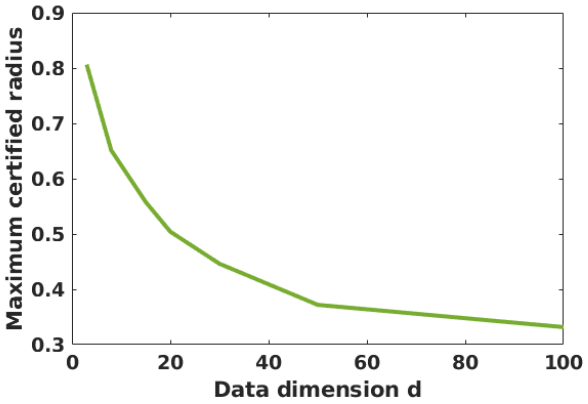


Figure B-3: The maximum certified radius with an increasing input-space dimension in the exemplary case.

vs. data dimension relationship as shown in Figure B-3.

B.1.3 Semi-bounded decision region certified radius behaviors w.r.t data dimensions

In Figure B-4, we show the unscaled certified radius r as a function of an increasing smoothing factor σ for different input data dimension d with fixed narrowness $\theta = 45°$. One can then see similar trend as told in Figure 3(a) of the main text in the bounded decision region case, the maximum certified radius (the peak) also decreases with the

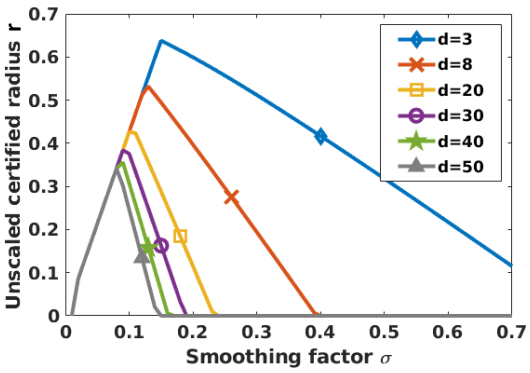


Figure B-4: The unscaled certified radius r of a point on the axis v for different input data dimension d.

increasing dimension.

B.2 Complete experimental details in Chapter 4

B.2.1 Full results of Section 4.3

Table B.1: The effectiveness evaluation of NaCl on SimCLR (*i.e.* $\alpha = 0, G^1 = g_0$). The best performance within each loss type is in boldface.

M	$\alpha = 0,\ \mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{NCA}}(g_0, M)$			
	CIFAR100 Acc.	FGSM Acc.	CIFAR10 Acc.	FGSM Acc.
1	53.69±0.25	25.17±0.55	76.34±0.28	43.50±0.41
2	55.72±0.15	27.04±0.45	77.40±0.14	44.58±0.41
3	56.67±0.12	**28.41±0.24**	77.53±0.24	**45.21±0.89**
4	57.09±0.26	28.20±0.81	77.75±0.22	45.13±0.44
5	**57.32±0.17**	28.33±0.59	**77.93±0.40**	44.46±0.53
	$\alpha = 0,\ \mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_0, M, 0.5)$			
1	53.69±0.25	**25.17±0.55**	76.34±0.28	**43.50±0.41**
2	54.76±0.29	23.66±0.27	76.78±0.26	40.76±0.66
3	55.21±0.17	24.46±0.44	77.45±0.18	41.78±0.80
4	55.68±0.27	24.19±0.46	77.40±0.24	41.33±0.34
5	**55.85±0.16**	24.01±0.91	**77.50±0.16**	40.77±0.66
	$\alpha = 0,\ \mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_0, M, 0.6)$			
1	53.69±0.25	25.17±0.55	76.34±0.28	**43.50±0.41**
2	54.84±0.35	25.94±0.81	77.11±0.15	42.81±0.83
3	55.49±0.13	**26.25±0.89**	76.95±0.32	42.99±0.96
4	55.65±0.24	25.41±0.53	**77.39±0.37**	42.69±1.20
5	**55.66±0.22**	26.01±0.60	77.26±0.48	43.06±0.79
	$\alpha = 0,\ \mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_0, M, 0.7)$			
1	53.69±0.25	25.17±0.55	76.34±0.28	43.50±0.41
2	55.57±0.32	27.67±0.60	77.09±0.27	44.68±0.71
3	55.83±0.25	27.72±0.59	77.23±0.28	43.68±0.72
4	56.29±0.25	**27.92±0.60**	77.33±0.29	44.69±0.82
5	**56.37±0.32**	27.78±0.54	**77.40±0.20**	**45.07±0.98**
	$\alpha = 0,\ \mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_0, M, 0.8)$			
1	53.69±0.25	25.17±0.55	76.34±0.28	43.50±0.41
2	55.75±0.21	29.30±0.86	76.80±0.20	46.56±1.02
3	56.27±0.26	**29.96±0.29**	77.11±0.37	46.52±0.50
4	**56.39±0.26**	29.49±0.65	77.34±0.31	46.79±0.93
5	56.23±0.13	29.47±0.95	**77.40±0.14**	**47.36±0.69**
	$\alpha = 0,\ \mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_0, M, 0.9)$			
1	53.69±0.25	25.17±0.55	76.34±0.28	43.50±0.41
2	56.20±0.33	30.95±0.36	76.96±0.15	**48.85±0.75**
3	56.41±0.13	**30.98±0.90**	77.10±0.21	48.76±0.63
4	56.00±0.42	29.90±0.63	**77.11±0.40**	48.16±0.40
5	**56.63±0.31**	30.58±0.52	77.04±0.19	47.96±0.46

Table B.2: The effectiveness evaluation of NaCl on Debised+HardNeg (*i.e.* $\alpha = 0, G^1 = g_2$). The best performance within each loss type is in boldface.

M	CIFAR100 Acc.	FGSM Acc.	CIFAR10 Acc.	FGSM Acc.
	$\alpha = 0,\ \mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{NCA}}(g_2, M)$			
1	56.83±0.20	31.03±0.41	77.24±0.29	48.38±0.70
2	57.87±0.15	32.50±0.48	77.43±0.11	48.14±0.31
3	58.42±0.23	**33.19±0.60**	77.41±0.17	48.09±0.93
4	**58.86±0.18**	32.65±1.07	77.46±0.29	**48.43±0.94**
5	58.81±0.21	32.86±0.47	**77.58±0.23**	48.30±0.39
	$\alpha = 0,\ \mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_2, M, 0.5)$			
1	56.83±0.20	31.03±0.41	77.24±0.29	48.38±0.70
2	59.41±0.19	**32.22±0.35**	79.36±0.65	48.86±0.34
3	59.81±0.25	32.04±0.67	79.41±0.17	48.91±0.81
4	59.75±0.33	32.03±0.34	79.42±0.18	**49.05±0.71**
5	**59.85±0.30**	32.06±0.72	**79.45±0.20**	48.32±0.70
	$\alpha = 0,\ \mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_2, M, 0.6)$			
1	56.83±0.20	31.03±0.41	77.24±0.29	48.38±0.70
2	58.94±0.29	32.65±0.36	78.67±0.15	**49.86±0.59**
3	59.43±0.35	32.91±0.40	78.94±0.19	48.84±1.09
4	**59.54±0.28**	33.02±0.62	78.92±0.29	49.64±0.74
5	59.52±0.28	**33.10±0.50**	**79.29±0.21**	49.39±1.02
	$\alpha = 0,\ \mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_2, M, 0.7)$			
1	56.83±0.20	31.03±0.41	77.24±0.29	48.38±0.70
2	58.24±0.19	33.24±0.90	78.30±0.31	**50.40±0.83**
3	58.74±0.26	33.12±0.59	78.49±0.30	49.85±0.38
4	58.79±0.38	**33.63±0.53**	78.51±0.29	49.88±0.75
5	**58.99±0.18**	32.93±0.81	**78.57±0.12**	49.53±1.55
	$\alpha = 0,\ \mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_2, M, 0.8)$			
1	56.83±0.20	31.03±0.41	77.24±0.29	48.38±0.70
2	57.60±0.15	34.14±0.22	**77.96±0.07**	**51.82±0.68**
3	58.04±0.28	33.93±0.45	77.55±0.18	50.30±0.81
4	58.05±0.16	**34.16±0.54**	77.90±0.21	50.40±0.43
5	**58.43±0.27**	33.87±0.62	77.90±0.17	50.78±0.95
	$\alpha = 0,\ \mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_2, M, 0.9)$			
1	56.83±0.20	31.03±0.41	77.24±0.29	48.38±0.70
2	57.16±0.15	34.25±0.55	77.19±0.09	**51.42±0.45**
3	57.08±0.10	33.96±0.19	77.21±0.26	51.30±1.05
4	57.36±0.19	**34.29±0.15**	**77.34±0.34**	51.16±0.55
5	**57.38±0.16**	34.25±0.30	77.13±0.16	50.68±0.74

Table B.3: The effectiveness evaluation of NaCl ($M \neq 1$) on IntCl ($M = 1$) when $\alpha = 1, G^1 = G^2 = g_2$. The best performance within each loss type is in boldface.

M	$\alpha \neq 0,\ \mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{NCA}}(g_2, M)$			
	CIFAR100 Acc.	FGSM Acc.	CIFAR10 Acc.	FGSM Acc.
1	56.22±0.15	40.05±0.67	76.39±0.10	**59.33±0.94**
2	56.71±0.11	39.80±0.57	76.55±0.27	58.44±0.31
3	57.13±0.26	40.53±0.29	**76.67±0.22**	58.47±0.31
4	57.06±0.19	40.85±0.31	76.34±0.22	58.91±0.62
5	**57.46±0.04**	**41.00±0.86**	76.60±0.37	57.98±0.47
	$\alpha \neq 0,\ \mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_2, M, 0.5)$			
1	56.22±0.15	40.05±0.67	76.39±0.10	59.33±0.94
2	58.97±0.19	40.25±0.52	78.61±0.20	58.41±0.59
3	59.26±0.18	40.96±0.58	**78.83±0.22**	59.20±1.25
4	59.32±0.21	40.82±0.54	78.83±0.27	59.03±0.52
5	**59.43±0.23**	**41.01±0.34**	78.80±0.21	**59.51±0.93**
	$\alpha \neq 0,\ \mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_2, M, 0.6)$			
1	56.22±0.15	40.05±0.67	76.39±0.10	59.33±0.94
2	58.55±0.34	**40.85±0.62**	78.34±0.22	**59.56±0.88**
3	59.05±0.21	40.83±0.44	78.41±0.12	59.14±0.78
4	59.06±0.25	40.80±0.89	78.61±0.22	58.41±1.00
5	**59.10±0.23**	40.68±0.50	**78.63±0.21**	58.92±0.76
	$\alpha \neq 0,\ \mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_2, M, 0.7)$			
1	56.22±0.15	40.05±0.67	76.39±0.10	59.33±0.94
2	58.00±0.18	40.35±0.34	77.73±0.24	59.40±1.27
3	58.23±0.18	40.94±0.75	77.91±0.25	**59.57±0.81**
4	58.20±0.25	40.95±0.45	77.89±0.20	59.49±0.49
5	**58.37±0.14**	**41.15±0.48**	**78.27±0.26**	59.17±0.94
	$\alpha \neq 0,\ \mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_2, M, 0.8)$			
1	56.22±0.15	40.05±0.67	76.39±0.10	59.33±0.94
2	57.07±0.24	41.29±0.57	77.27±0.28	60.16±0.51
3	**57.62±0.22**	40.93±0.49	77.54±0.27	59.47±0.52
4	57.61±0.25	**41.36±0.41**	77.50±0.34	**60.28±0.68**
5	57.56±0.18	40.71±0.34	**77.58±0.42**	59.99±0.30
	$\alpha \neq 0,\ \mathcal{L}_{\text{NaCl}}(G^1, M, \lambda) = \mathcal{L}_{\text{MIXNCA}}(g_2, M, 0.9)$			
1	56.22±0.15	40.05±0.67	76.39±0.10	59.33±0.94
2	56.54±0.33	40.85±0.13	76.81±0.22	60.40±0.46
3	56.69±0.11	41.23±0.66	**76.98±0.22**	60.13±0.56
4	56.43±0.26	**41.56±0.56**	76.97±0.20	**61.21±0.49**
5	**56.86±0.11**	41.09±0.31	76.91±0.21	60.09±0.39

B.2.2 Experimental details

Architecture. We follow [23, 154] to incorporate an MLP projection head during the contrastive learning on resnet18.

Optimizer. Adam optimizer with a learning rate of $3e-4$.

Training epochs. The representation network is trained for 100 epochs. For CIFAR100 and CIFAR10, the downstream fully-connected layer is trained for 1000 epochs. For TinyImagenet, the fully-connected layer is trained for 200 epochs.

Methodological hyperparameters. Throughout out experiments, we use $\tau^+ = 0.01$ and $\beta = 1.0$ for $\mathcal{L}_{\text{Debiased}}$ [29] and $\mathcal{L}_{\text{Debiased+HardNeg}}$ [154], $\alpha = 1$ for $\mathcal{L}_{\text{Adv}}$ [75]. The same set of hyperparameters are used in our IntCl and IntNaCl.

Data augmentation. Our data augmentation includes random resized crop, random horizontal flip, random grayscale, and color jitter. Specifically, we implement the color jitter by calling $torchvision.transforms.ColorJitter(0.8*s, 0.8*s, 0.8*s, 0.2*s)$ and execute with probability 0.8. Random grayscale is performed with probability 0.2.

Adversarial hyperparameters. When evaluating the adversarial robustness using the codebase provided in [200], we use a PGD step size of $1e-2$, 10 iterations, and 2 random restarts.

Error bar. We run five independent trials for each of the experiments and report the mean and standard deviation for all tables and figures. The error bars in Figure B-5 is omitted for better visual clarity.

Robust Accuracy. In Figure B-5, we show the robust accuracy as a function of the FGSM attack strength ϵ. Specifically, we range the attack strength from 0.002 to 0.032 and give the robust accuracy of our proposals (IntCl & IntNaCl) together with baselines under all attacks. From Figure B-5, one can see that among all baselines, Adv

demonstrates the best adversarial robustness, whereas our proposals still consistently win over it by a noticeable margin.

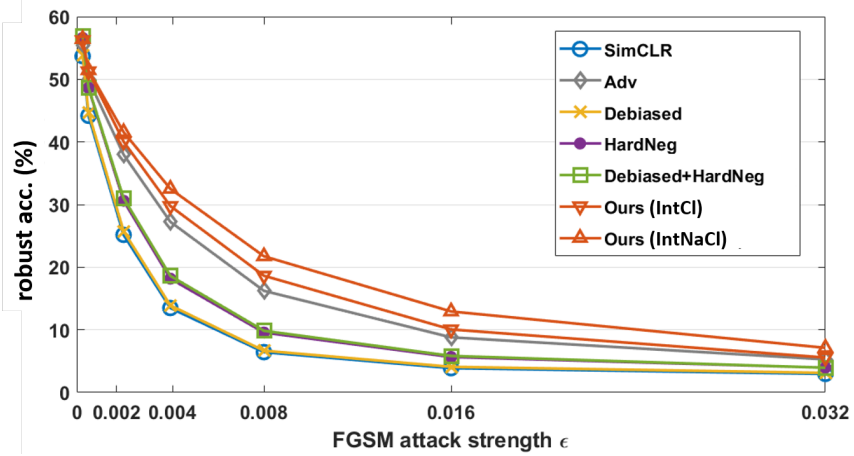


Figure B-5: The robust accuracy under FGSM attacks of different strength on CIFAR100.

B.3 Complete experimental details in Chapter 5

B.3.1 Full results of Section 5.3.2

a_t	Model	$\epsilon = 0$	$\epsilon = 0.1$	$\epsilon = 0.2$	$\epsilon = 0.3$	$\epsilon = 0.4$	$\epsilon = 0.5$	$\epsilon = 0.6$	$\epsilon = 0.7$	$\epsilon = 0.8$
0.7	ViT-Ti/16	0.01	0.01	0	0	0	0	0	0	0
	ViT-B/16	0.33	0.36	0.37	0.35	0.32	0.27	0.20	0.13	0.07
	ViT-B/16-in21k	0.20	0.22	0.23	0.21	0.17	0.13	0.07	0.03	0.01
	ViT-L/16	0.26	0.30	0.33	0.32	0.30	0.27	0.22	0.17	0.11
	ViT-S/16-DINO	0.48	0.48	0.47	0.45	0.42	0.37	0.32	0.25	0.17
	ViT-B/16-DINO	0.55	0.58	0.58	0.56	0.53	0.50	0.46	0.41	0.35
	ViT-S/8-DINO	0.40	0.42	0.42	0.41	0.39	0.37	0.34	0.30	0.26
	ViT-B/8-DINO	0.50	0.55	0.56	0.54	0.50	0.45	0.40	0.35	0.30
	Resnet50-SimCLRv2	0.66	0.53	0.50	0.49	0.50	0.49	0.48	0.48	0.48
	Resnet101-SimCLRv2	0.60	0.74	0.64	0.58	0.56	0.52	0.51	0.50	0.48
0.75	ViT-Ti/16	0	0	0	0	0	0	0	0	0
	ViT-B/16	0.26	0.29	0.30	0.28	0.25	0.18	0.11	0.05	0.01
	ViT-B/16-in21k	0.12	0.15	0.16	0.14	0.10	0.06	0.02	0	0
	ViT-L/16	0.19	0.24	0.27	0.27	0.24	0.21	0.16	0.10	0.04
	ViT-S/16-DINO	0.42	0.42	0.41	0.39	0.36	0.32	0.26	0.19	0.11
	ViT-B/16-DINO	0.50	0.54	0.54	0.51	0.48	0.44	0.39	0.34	0.27
	ViT-S/8-DINO	0.33	0.34	0.34	0.33	0.31	0.29	0.25	0.21	0.16
	ViT-B/8-DINO	0.44	0.50	0.51	0.48	0.43	0.37	0.31	0.25	0.19
	Resnet50-SimCLRv2	0.33	0.44	0.40	0.39	0.39	0.39	0.39	0.39	0.39
	Resnet101-SimCLRv2	0.54	0.69	0.57	0.49	0.45	0.43	0.40	0.38	0.36
0.8	ViT-Ti/16	0	0	0	0	0	0	0	0	0
	ViT-B/16	0.19	0.22	0.23	0.21	0.17	0.11	0.04	0	0
	ViT-B/16-in21k	0.06	0.08	0.09	0.07	0.04	0.01	0	0	0
	ViT-L/16	0.12	0.17	0.21	0.20	0.18	0.14	0.09	0.04	0
	ViT-S/16-DINO	0.34	0.35	0.34	0.32	0.29	0.25	0.19	0.13	0.05
	ViT-B/16-DINO	0.45	0.49	0.49	0.46	0.42	0.37	0.32	0.26	0.17
	ViT-S/8-DINO	0.25	0.26	0.26	0.25	0.23	0.20	0.16	0.12	0.08
	ViT-B/8-DINO	0.38	0.45	0.46	0.42	0.36	0.29	0.22	0.16	0.10
	Resnet50-SimCLRv2	0.09	0.34	0.31	0.31	0.30	0.30	0.30	0.30	0.30
	Resnet101-SimCLRv2	0.46	0.62	0.50	0.39	0.35	0.32	0.29	0.26	0.24
0.85	ViT-Ti/16	0	0	0	0	0	0	0	0	0
	ViT-B/16	0.10	0.14	0.15	0.13	0.09	0.04	0	0	0
	ViT-B/16-in21k	0.01	0.03	0.02	0.02	0	0	0	0	0
	ViT-L/16	0.05	0.09	0.13	0.12	0.10	0.07	0.03	0	0
	ViT-S/16-DINO	0.25	0.26	0.25	0.24	0.21	0.17	0.12	0.06	0.01
	ViT-B/16-DINO	0.38	0.43	0.43	0.40	0.35	0.29	0.23	0.15	0.08
	ViT-S/8-DINO	0.16	0.17	0.17	0.15	0.13	0.10	0.07	0.04	0.02
	ViT-B/8-DINO	0.30	0.38	0.38	0.34	0.27	0.19	0.13	0.06	0.03
	Resnet50-SimCLRv2	0	0.22	0.20	0.20	0.19	0.19	0.19	0.20	0.19
	Resnet101-SimCLRv2	0.37	0.55	0.41	0.28	0.23	0.19	0.16	0.13	0.11
0.9	ViT-Ti/16	0	0	0	0	0	0	0	0	0
	ViT-B/16	0.02	0.04	0.05	0.04	0.01	0	0	0	0
	ViT-B/16-in21k	0	0	0	0	0	0	0	0	0
	ViT-L/16	0	0.01	0.04	0.04	0.03	0.01	0	0	0
	ViT-S/16-DINO	0.13	0.14	0.14	0.13	0.10	0.07	0.04	0.01	0
	ViT-B/16-DINO	0.28	0.34	0.34	0.30	0.23	0.16	0.10	0.05	0
	ViT-S/8-DINO	0.05	0.06	0.06	0.05	0.04	0.02	0.01	0	0
	ViT-B/8-DINO	0.20	0.29	0.28	0.23	0.15	0.07	0.03	0	0
	Resnet50-SimCLRv2	0	0.08	0.06	0.06	0.06	0.06	0.06	0.06	0.06
	Resnet101-SimCLRv2	0.23	0.42	0.28	0.15	0.08	0.06	0.04	0.02	0.01

Table B.4: Full table of Table 5.2.

n	Name	ViT-B/16	ViT-L/16	ViT-B/32	Resnet50-SimCLRv2	Resnet101-SimCLRv2	Pearson correlation
Real-life	Accuracy (%)	74.3	75.5	72.6	75.4	75.4	1.0
Transfer dataset	ImageNet-c acc.	66.4	72.2	61.4	47.4	50.1	0.64
	ImageNet-r acc.	56.8	64.3	49.4	39.4	44.1	-0.03
	ImageNet-a acc.	43.1	55.3	22.3	27.1	38.2	0.57
	CIFAR10-lc acc.	93.54	94.95	92.48	85.74	87.38	-0.36
2048	Val loss	3.10	4.12	4.10	1.31	0.98	-0.55
	MDL	6820.76	8094.06	8198.55	5881.34	2882.36	-0.50
	SDL, $\varepsilon = 1$	> 4978	> 6251	> 6356	> 4038	1052.37	-
	εSC, $\varepsilon = 1$	> 1843	> 1843	> 1843	> 1843	1843	-
	LogME	-0.726	-0.724	-0.729	2.791	1.503	0.54
	SFDA	0.584	0.635	0.567	0.947	0.593	0.46
	SynBench	0.33	0.26	0.02	0.66	0.60	**0.79**
8192	Val loss	0.73	1.50	2.92	0.62	0.52	-0.81
	MDL	9939.13	17672.6	23332.98	9646.09	5443.43	-0.68
	SDL, $\varepsilon = 1$	3479.59	> 10301	> 15961	3700.73	776.38	-
	εSC, $\varepsilon = 1$	7372	> 7372	> 7372	4045	669	-
	LogME	-0.710	-0.707	-0.727	-0.599	-0.622	0.65
	SFDA	0.525	0.531	0.513	0.581	0.543	0.67
	SynBench	0.52	0.49	0.01	0.69	0.84	**0.89**
32768	Val loss	0.68	0.79	3.91	0.53	0.51	-0.92
	MDL	30848.99	38718.04	107960.49	22022.08	17166.0	-0.91
	SDL, $\varepsilon = 1$	7043.32	12496.0	> 78469.49	4355.67	969.27	-
	εSC, $\varepsilon = 1$	14265	29491	> 29491.0	3338	1615	-
	LogME	-0.686	-0.687	-0.725	-0.580	-0.608	0.72
	SFDA	0.517	0.518	0.505	0.545	0.534	0.77
	SynBench	0.59	0.58	0.02	0.81	0.87	**0.92**

Table B.5: Pearson correlation between task agnostic metrics and the average accuracy on 27 real-life tasks [141, Table 10] . We report the 5 pretrained models out of the overall 10 due to the lack of reported results from the literature for the other pretrain models.

For completeness, we report several baseline metrics for the synthetic conditional Gaussian classification task. We follow the implementation of [196, 162] and set the training set size n to be $2048, 8192, 32768$. In Table B.5, we report validation loss (val loss), minimum description length (MDL) [185], surplus description length (SDL), ϵ-sample complexity (ϵ-SC) [196], logarithm of maximum evidence (LogME) [210] and self-challenging Fisher discriminant analysis (SFDA) [162] on our synthetic proxy task as baselines. We aim at calculating the Pearson correlation between task-agnostic metrics and possible downstream tasks. We take the average accuracy of 27 downstream tasks in the literature [141] for each pretrained model and treat it as the real-life performance measure. For an even more complete picture, we also consider some synthetic distribution shifts that include image corruptions (ImageNet-c), style transfer (ImageNet-r), and adversarial examples (ImageNet-a). To analyze how data with these synthetic distribution shifts can inform general pretrained models' performance, we quoted the their accuracy from [197] and calculated their correlation with the average

real-life accuracy in Table B.5. Furthermore, following [214], we perform "partially corrupted labels" experiments on CIFAR10 dataset with the level of label corruptions equals to 0.5. See line "CIFAR10-lc acc." for the results. We note that the correlation coefficients in these four cases suggest only moderate correlation to even negative correlation.

We set the training set size n to be $2048, 4096, 8192, 16384, 32768$ and compare the model selections between ViT-B/16 and ViT-B/16-in21k in Table B.6. In Table B.7, we report these metrics on all 10 pretrained representations for $n = 8192$.

n	Name	ViT-B/16	ViT-B/16-in21k
2048	Val loss	3.10	3.37
	MDL	6820.76	7114.12
	SDL, ε=1	$>$ 4977.76	$>$ 5271.12
	εSC, ε=1	$>$ 1843.0	$>$ 1843.0
	SynBench	0.33	0.20
4096	Val loss	1.77	1.41
	MDL	10813.95	9412.53
	SDL, ε=1	$>$ 7127.95	$>$ 5726.53
	εSC, ε=1	$>$ 3686.0	$>$ 3686.0
	SynBench	0.45	0.30
8192	Val loss	0.73	0.77
	MDL	9939.13	9773.16
	SDL, ε=1	3479.59	3153.33
	εSC, ε=1	7372	7372
	SynBench	0.52	0.38
16384	Val loss	0.85	0.86
	MDL	20936.18	20899.58
	SDL, ε=1	7266.8	7136.29
	εSC, ε=1	14745	14745
	SynBench	0.56	0.41
32768	Val loss	0.68	0.70
	MDL	30848.99	32944.76
	SDL, ε=1	7043.32	8611.49
	εSC, ε=1	14265	14265
	SynBench	0.59	0.44

Table B.6: Baseline metrics evaluating the representation quality on the conditional Gaussian synthetic data with $n = \{2048, 4096, 8192, 16384, 32768\}$. For Val loss, MDL, SDL, and ϵSC, the smaller the better; for SynBench, the bigger the better. Note that the model ranking of SynBench is consistent across different values of n, while other methods will change their rankings.

Name	Val loss	MDL	SDL, ε=1	εSC, ε=1
ViT-Ti/16	4.38	30071.64	> 22699.64	> 7372.0
ViT-B/16	0.73	9939.13	3479.59	7372
ViT-L/16	1.50	17672.6	> 10300.6	> 7372.0
ViT-B/16-in21k	0.77	9773.16	3153.33	7372
ViT-S/16-DINO	1.51	18536.93	> 11164.93	> 7372.0
ViT-S/8-DINO	0.70	8196.8	2056.69	4045
ViT-B/16-DINO	0.92	10535.11	3432.28	7372
ViT-B/8-DINO	0.64	6796.87	1185.31	2220
Resnet50-SimCLRv2	0.62	9646.09	3700.73	4045
Resnet101-SimCLRv2	0.52	5443.43	776.38	669

Table B.7: Baseline metrics evaluating the representation quality on the conditional Gaussian synthetic data with $n = 8192$.

B.3.2 Full results of Section 5.3.3

Models		CIFAR10				TinyImagenet			
		$\epsilon = 0$	$\epsilon = 0.1$	$\epsilon = 0.2$	$\epsilon = 0.3$	$\epsilon = 0$	$\epsilon = 0.1$	$\epsilon = 0.2$	$\epsilon = 0.3$
ViT-Ti/16	SynBench-Score(ϵ)	**0.01**	**0.01**	0	0	**0.01**	**0.01**	0	0
	ϵ-robust prob. δSA	0	-3.1	-5.9	-6.3	0	+0.3	-1.5	-1.9
	ϵ-robust prob. δRA	0	+1.4	+1.9	+1.6	0	+1.1	+0.4	+2.2
ViT-B/16	SynBench-Score(ϵ)	0.33	0.36	**0.37**	0.35	0.33	0.36	**0.37**	0.35
	ϵ-robust prob. δSA	0	+0.2	+0.1	+0.1	0	0	+0.7	+0.6
	ϵ-robust prob. δRA	0	+0.3	+2.7	+2.3	0	-1.0	+2.5	+2.4
ViT-B/16-in21k	SynBench-Score(ϵ)	0.20	0.22	**0.23**	0.21	0.20	0.22	**0.23**	0.21
	ϵ-robust prob. δSA	0	+0.9	+1.1	+1.1	0	+0.3	+0.3	+0.2
	ϵ-robust prob. δRA	0	+1.2	+1.4	+0.6	0	+1.3	+2.0	+2.0
ViT-L/16	SynBench-Score(ϵ)	0.26	0.30	**0.33**	0.32	0.26	0.30	**0.33**	0.32
	ϵ-robust prob. δSA	0	+0.2	+0.4	+0.4	0	-0.1	-0.2	-0.3
	ϵ-robust prob. δRA	0	-0.2	+3.0	+1.9	0	+4.2	+6.6	+0.7

Table B.8: Full Table of Table 5.3.

B.3.3 Full results of Section 5.3.4

$a_t = 0.7$	$\epsilon = 0$	$\epsilon = 0.1$	$\epsilon = 0.2$	$\epsilon = 0.3$	$\epsilon = 0.4$	$\epsilon = 0.5$	$\epsilon = 0.6$	$\epsilon = 0.7$	$\epsilon = 0.8$
ViT-B/16	0.18	0.22	0.24	0.23	0.20	0.15	0.10	0.05	0.01
ViT-B/16-in21k	0.07	0.10	0.11	0.10	0.07	0.04	0.01	0	0

Table B.9: SynBench-Score comparisons on the finetuning procedure in pretraining on synthetic data with heptadiagonal covariance.

$a_t = 0.7$	$\epsilon = 0$	$\epsilon = 0.1$	$\epsilon = 0.2$	$\epsilon = 0.3$	$\epsilon = 0.4$	$\epsilon = 0.5$	$\epsilon = 0.6$	$\epsilon = 0.7$	$\epsilon = 0.8$
ViT-Ti/16	0	0	0	0	0	0	0	0	0
ViT-B/16	0.18	0.22	0.24	0.23	0.20	0.15	0.10	0.05	0.01
ViT-L/16	0.18	0.24	0.28	0.29	0.28	0.27	0.23	0.18	0.12

Table B.10: SynBench-Score comparisons on the model sizes on synthetic data with heptadiagonal covariance.

$a_t = 0.7$	$\epsilon = 0$	$\epsilon = 0.1$	$\epsilon = 0.2$	$\epsilon = 0.3$	$\epsilon = 0.4$	$\epsilon = 0.5$	$\epsilon = 0.6$	$\epsilon = 0.7$	$\epsilon = 0.8$
ViT-S/16-DINO	0.47	0.47	0.46	0.44	0.39	0.31	0.23	0.13	0.03
ViT-B/16-DINO	0.42	0.50	0.52	0.52	0.51	0.48	0.45	0.40	0.35
ViT-S/8-DINO	0.36	0.38	0.38	0.38	0.36	0.33	0.30	0.26	0.20
ViT-B/8-DINO	0.42	0.52	0.55	0.53	0.50	0.45	0.40	0.33	0.28
Res50-SimCLRv2	0.24	0.53	0.47	0.38	0.36	0.34	0.33	0.32	0.31
Res101-SimCLRv2	0.30	0.47	0.37	0.34	0.32	0.31	0.30	0.29	0.29

Table B.11: SynBench-Scores of self-supervised pretrained representations on synthetic data with heptadiagonal covariance.

B.3.4 Intuitions on how SynBench predict classification performance across a broad range of tasks

Think of how representation learning research typically evaluate a model for transfer learning - by running tests on a broad range of downstream tasks. And the reason behind this is to see how the model behaves in different scenarios. To theorize things, we believe the general behavior of a pretrained representation is measured by how it perform on tasks of different difficulty levels. That is why we think a fundamental part of our design is to simulate tasks of different difficulty levels. One difference between SynBench and a traditional probing test is that, for example, we are using the classification problem of two highly overlapped Gaussian, instead of classifying ImageNet21k. We hope this clarification builds enough intuition to understand the following:

1. We vary s from 0.1 to 5 in increments of 0.1, which correspond to optimal accuracy (ground-truth difficulty) ranging from 55% to 100% and 50 difficulty levels. If we refer to Figure B-6, we see each of the red points correspond to one of our simulated trials with difficulty levels (x-axis).

2. Baseline methods are task/data dependant, which means they are somewhat

bound to tasks of that similar difficulty levels. If we refer to Figure B-6, it could be the single purple point with fixed level of difficulty.

3. If we include certain knowledge of possible downstream data properties, say locality of pixel dependencies, then the prediction will indeed be more accurate (see our section 5.3.4).

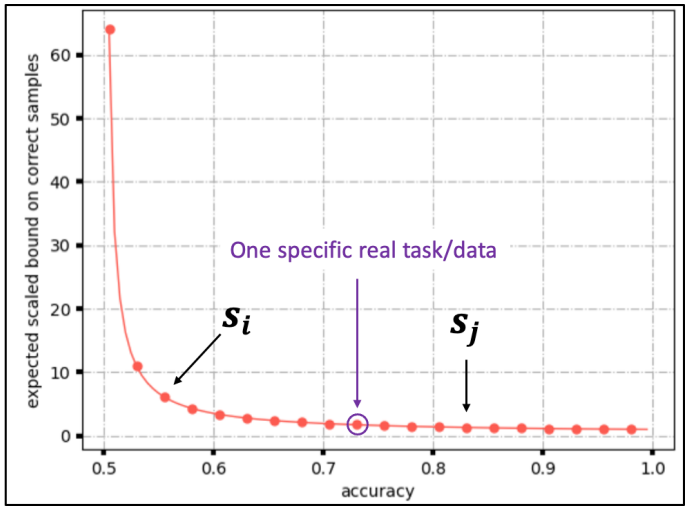


Figure B-6: Illustrations of the difference between SynBench synthetic data difficulty coverage and a specific real task/data.

B.3.5 Rejection mechanism

Dataset	Gaussian-I	Gaussian-H
CIFAR10	0.37	0.65
SVHN	0.58	0.83
TinyImageNet	0.31	0.51

Table B.12: The p-values in the hypothesis testing for Gaussian-I and Gaussian-H distributions.

SynBench is a task-agnostic benchmark and it is designed to be used to test pretrained models without the prior knowledge of the downstream task (e.g. model auditing etc). In the case when we do know some knowledge of the tasks, e.g. pixel dependencies, one can use the knowledge to fine-tune the GMM SynBench uses. However, in the case when we know exactly which downstream task will we do and

the downstream datasets are accessible and representative,, the best practice is to directly to apply linear probing. If we are to come up with a rejection mechanism, then one can potentially use goodness-of-fit tests to verify the null hypothesis that the downstream data of interest are generated from a Normal distribution. If the data follow Normal distribution, the Mahalanobis distances should follow a Chi-Squared distribution with degrees of freedom equal to the number of features. Then since the CDF for the appropriate degrees of freedom gives the probability of having obtained a value less extreme than this point, subtracting the CDF value from 1 gives the p-value. We conduct the experiment for CIFAR10, SVHN, and TinyImageNet, and report the p-values in Table B.12. Because these p-values are high, we can't reject this hypothesis. But if the p-value is below a threshold, one can reject this hypothesis.

B.3.6 Pearson and confidence interval

Let r be the Pearson correlation coefficient, p be the number of models. We ran the calculation for confidence intervals and see that the upper and lower confidence interval limits in z-space are $0.5\ln\left(\frac{1+r}{1-r}\right) \pm 1.645\sqrt{\frac{1}{p-3}} = 1.589 \pm 1.163$ for $r = 0.92$ and $p = 5$ when the training set size $n = 32768$. Translating to r-space by $r = \frac{e^{2z}-1}{e^{2z}+1}$ yields the upper limit of 0.992 and the lower limit of 0.402, if the desired confidence level is 90%. In the following Table B.13, we added four efficient nets' SynBench-scores, together with the average of their reported performance on 27 downstream tasks in [141], Table 10. We ran the same calculation for the Pearson correlation coefficient $r = 0.88$ and $p = 9$ to obtain the confidence interval of $[0.607, 0.967]$ which suggest at least moderate correlation up to strong correlation.

In the following Figure B-7, we plot the Pearson correlation coefficients of each methods with their confidence interval for a 90% confidence level when the training set size n = 2048.

n	Name	ViT-B/16	ViT-L/16	ViT-B/32	Resnet50-SimCLRv2	Resnet101-SimCLRv2	EfficientNet b0	EfficientNet b1	EfficientNet b2	EfficientNet b3	Pearson correlation
Real-life	Accuracy (%)	74.3	75.5	72.6	75.4	75.4	72.5	72.6	73.1	73.9	1.0
2048	Val loss	3.10	4.12	4.10	1.31	0.98	4.66	3.56	6.82	3.88	-0.63
	MDL	6820.76	8094.06	8198.55	5881.34	2882.36	8950.38	7654.88	15816.05	8138.87	-0.53
	SDL, $\varepsilon = 1$	> 4978	> 6251	> 6356	> 4038	1052.37	>7107	>5812	>13973	>6296	-
	εSC, $\varepsilon = 1$	> 1843	> 1843	> 1843	> 1843	1843	> 1843	> 1843	> 1843	> 1843	-
	LogME	-0.726	-0.724	-0.729	2.791	1.503	-0.721	-0.726	-0.725	-0.729	0.67
	SFDA	0.584	0.635	0.567	0.947	0.593	0.534	0.515	0.751	0.823	0.44
	SynBench	0.33	0.26	0.0	0.66	0.60	0.02	0.04	0	0	0.85
8192	Val loss	0.73	1.50	2.92	0.62	0.52	4.27	2.03	4.33	2.56	-0.78
	MDL	9939.13	17672.6	23332.98	9646.09	5443.43	32511.61	19479.78	43202.85	25964.38	-0.69
	SDL, $\varepsilon = 1$	3479.59	> 10301	> 15961	3700.73	776.38	>25140	>12108	>35831	>18592	-
	εSC, $\varepsilon = 1$	7372	> 7372	> 7372	4045	669	> 7372	> 7372	> 7372	> 7372	-
	LogME	-0.710	-0.707	-0.727	-0.599	-0.622	-0.714	-0.719	-0.721	-0.725	0.71
	SFDA	0.525	0.531	0.513	0.581	0.543	0.510	0.505	0.524	0.525	0.78
	SynBench	0.52	0.49	0.01	0.69	0.84	0.13	0.13	0.09	0.03	0.87
32768	Val loss	0.68	0.79	3.91	0.53	0.51	1.11	0.79	2.60	1.11	-0.58
	MDL	30848.99	38718.04	107960.49	22022.08	17166.0	56621.37	39158.90	109706.34	56621.37	-0.67
	SDL, $\varepsilon = 1$	7043.32	12496	> 78469	4356	969.27	>27130	12932	> 80215	>27130	-
	εSC, $\varepsilon = 1$	14265	29491	> 29491	3338	1615	> 29491	29491	> 29491	> 29491	-
	LogME	-0.686	-0.687	-0.725	-0.580	-0.608	-0.713	-0.719	-0.715	-0.718	0.79
	SFDA	0.517	0.518	0.505	0.545	0.534	0.505	0.504	0.508	0.508	0.84
	SynBench	0.59	0.58	0.02	0.81	0.87	0.19	0.19	0.17	0.09	0.88

Table B.13: The correlation between SynBench-score and the average accuracy on 27 real-life tasks.

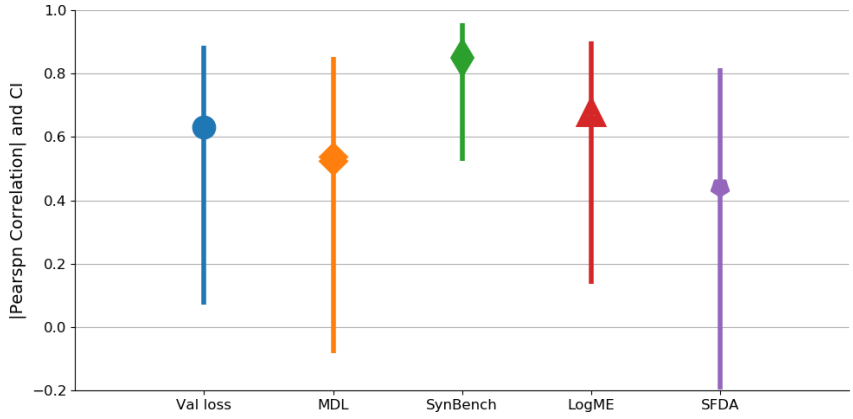


Figure B-7: The Pearson r and 90% confidence intervals.

B.4 Complete experimental details in Chapter 6

B.4.1 Full results of Section 6.3.2

Table B.14: Pearson correlation comparison between real-data-free evaluation methods and the average linear probing accuracy on the real-world tasks included in Table B.15. Since the smaller the Val loss, MDL, SDL and ϵSC, the better, we add a negative sign in front of them when calculating the Pearson correlation coefficient.

n	Name	$\text{BERT}_{\text{base}}$	DiffCSE-B	$\text{BERT}_{\text{large}}$	T5_{base}	T5_{large}	$\text{RoBERTa}_{\text{base}}$	DiffCSE-R	GPT	ST5	Pearson
	Real-life acc.	83.50	86.81	83.68	82.78	82.36	83.83	88.19	78.01	90.17	1.0
4096	Val loss	1.0e-06±1e-07	1.4e-06±3e-07	7.6e-07±5e-08	8.5e-08±1e-08	5.4e-08±9e-09	4.0e-06±3e-07	1.1e-06±8e-08	3.1e-03±8e-04	3.7e-03±5e-03	0.285±0.498
	MDL	5002±318	4755±129	5422±357	7318±119	6724±228	5396±181	4773±296	5604±366	4433±360	0.571±0.109
	SDL, ε=1	3090±318	2843±129	3510±357	5406±119	4812±228	3484±181	2861±296	3687±366	2514±368	0.570±0.110
	εSC, ε=1	3686±0	3686±0	3686±0	3686±0	3686±0	3686±0	3686±0	3686±0	3686±0	-
	SynTextBench	0.137±0.001	0.148±0.001	0.135±0.000	0.111±0.002	0.103±0.002	0.119±0.001	0.193±0.001	0.090±0.003	0.214±0.000	0.939±0.008
8192	Val loss	3.3e-06±3e-07	6.3e-04±9e-04	6.6e-04±9e-04	3.3e-07±9e-08	5.9e-04±8e-04	1.3e-05±1e-06	4.1e-06±2e-07	3.1e-02±1e-03	1.2e-03±5e-05	0.649±0.004
	MDL	8802±99	8687±260	10107±156	14664±464	14487±426	9801±489	8902±175	10001±291	7310±175	0.519±0.043
	SDL, ε=1	5262±99	5144±262	6564±155	11124±464	10944±426	6261±489	5362±175	6343±287	3766±175	0.509±0.043
	εSC, ε=1	7372±0	7372±0	7372±0	7372±0	7372±0	7372±0	7372±0	7372±0	7372±0	-
	SynTextBench	0.152±0.001	0.156±0.001	0.148±0.002	0.130±0.001	0.122±0.000	0.129±0.002	0.196±0.001	0.085±0.003	0.223±0.001	0.968±0.006
16384	Val loss	2.3e-03±2e-03	9.5e-04±7e-04	7.2e-04±1e-03	6.6e-04±9e-04	1.2e-03±9e-05	8.2e-04±1e-03	2.2e-03±2e-03	2.1e-01±3e-02	2.3e-02±9e-04	0.605±0.007
	MDL	15840±436	15253±455	18039±778	26004±879	25606±767	16629±117	15465±349	16794±440	11895±89	0.506±0.032
	SDL, ε=1	9266±429	8689±458	11477±786	19443±887	19040±767	10066±118	8891±365	8525±383	5153±93	0.425±0.021
	εSC, ε=1	14745±0	14745±0	14745±0	14745±0	14745±0	14745±0	14745±0	14745±0	14745±0	-
	SynTextBench	0.161±0.000	0.164±0.001	0.161±0.001	0.145±0.000	0.141±0.001	0.137±0.000	0.198±0.001	0.087±0.001	0.227±0.001	0.958±0.002
32768	Val loss	6.4e-03±8e-04	4.2e-03±2e-03	4.1e-03±3e-04	3.1e-02±1e-02	3.0e-03±7e-04	1.4e-02±2e-03	1.1e-02±1e-02	4.7e-01±2e-02	2.9e-01±1e-02	0.267±0.018
	MDL	27667±294	25793±898	29577±253	43955±1616	39692±1520	27151±33	27546±646	28930±471	21999±88	0.481±0.029
	SDL, ε=1	15417±282	13581±927	17367±252	31282±1860	27501±1518	14775±50	15214±489	9442±195	6076±106	0.311±0.008
	εSC, ε=1	29491±0	29491±0	29491±0	29491±0	29491±0	29491±0	29491±0	12139±0	12139±0	-0.044±0.000
	SynTextBench	0.170±0.001	0.169±0.000	0.173±0.001	0.158±0.001	0.156±0.000	0.140±0.001	0.202±0.000	0.092±0.001	0.230±0.000	0.934±0.002

Table B.15: The detailed SentEval linear probing performance. For STS tasks, we report Spearman's correlation (%), and for Transfer task, we report the standard accuracy (%).

	STS tasks							Transfer tasks							
Models	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	avg.
$\text{BERT}_{\text{base}}$	54.44	58.03	58.86	67.94	68.42	53.88	62.06	82.98	89.56	95.43	89.92	85.45	89.8	74.03	83.50
DiffCSE-B	68.88	76.21	73.88	79.76	78.84	75.51	67.70	82.2	88.11	95.44	91.03	84.46	88	75.71	86.81
$\text{BERT}_{\text{large}}$	53.33	56.86	56.23	63.43	66.69	54.43	58.06	85.96	89.59	96.43	90.96	89.13	91.8	73.16	83.68
T5_{base}	58.18	63.78	64.14	71.83	68.94	60.17	58.77	80.54	88.34	93.04	89.73	81.27	85.8	67.36	82.78
T5_{large}	58.34	62.59	63.50	71.36	67.88	59.67	58.02	79.31	86.86	93.53	90.43	80.72	82.8	68.75	82.36
$\text{RoBERTa}_{\text{base}}$	57.28	55.21	59.76	69.22	64.64	58.55	61.63	84.08	86.91	95.63	89.52	88.25	91.6	74.49	83.83
DiffCSE-R	69.77	78.70	76.08	81.75	80.86	81.17	70.34	84.75	90.99	95.2	89.75	87.92	89.4	77.28	88.19
GPT	44.16	23.99	34.73	40.78	55.11	41.05	43.65	81.08	88.53	92.81	87.87	86.6	93	70.49	78.01
ST5	74.32	82.83	81.50	86.14	85.95	86.04	79.76	85.88	91.81	94.4	91.09	90.88	95.8	74.26	90.17

B.4.2 Full results of Section 6.3.3

In-context learning. We evaluate the few-shot in-context learning (ICL) performance on SentEval transfer tasks and SynTextBench synthetic task. We do not include STS tasks since they are typically measured by cosine distance, whose ICL prompts are less obvious to us. We also excluded TREC as we have not found proper prompts that could lead to reasonable accuracy.

Figure B-8: The accuracy and robustness (average number of perturbed words) performance of pretrained models on SentEval tasks.

Table B.16: The detailed SentEval linear probing performance on decoder models. For STS tasks, we report Spearman's correlation (%), and for Transfer tasks, we report the standard accuracy (%).

	STS tasks							Transfer tasks							
Models	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	avg.
LLaMA-7B	10.51	9.68	5.85	2.60	5.87	15.58	15.01	71.20	75.87	87.59	81.94	77.59	62.80	64.12	64.55
LLaMA-13B	12.08	7.05	2.86	-0.84	7.38	3.50	10.93	70.88	77.06	88.04	81.53	76.77	64.00	63.19	63.78
LLaMA-30B	7.04	16.29	5.39	3.12	5.04	16.02	14.77	71.99	78.12	88.81	82.53	76.44	61.00	60.70	64.53
LLaMA-2-7B	11.95	22.85	10.85	16.31	44.42	20.13	47.17	91.07	91.95	97.30	89.22	94.78	96.80	67.88	76.13
LLaMA-2-13B	21.80	33.07	18.79	19.31	50.67	33.84	50.83	92.03	92.32	97.70	89.72	95.61	97.20	70.38	78.51
OPT-13B	24.20	40.78	24.91	25.75	56.70	39.44	51.32	91.23	92.45	97.13	89.28	95.00	96.80	72.58	79.72
OPT-30B	24.63	38.83	22.25	26.00	57.93	39.95	52.17	91.36	92.71	97.28	89.39	95.11	97.00	68.41	79.44

Table B.17: Pearson correlation comparison between real-data-free evaluation methods and the average linear probing accuracy on the real-world tasks of decoder models. Since the smaller the Val loss, MDL, SDL and ϵSC, the better, we add a negative sign in front of them when calculating the Pearson correlation coefficient.

n	Name	LLaMA-7B	LLaMA-13B	LLaMA-30B	LLaMA-2-7B	LLaMA-2-13B	OPT-13B	OPT-30B	Pearson
	Real-life acc.	64.55	63.78	64.53	76.13	78.51	79.72	79.44	1.0
8192	Val loss	0.036141	0.149492	0.075583	0.000002	0.0	0.010351	0.00362	0.803
	MDL	8114.26	7434.78	6920.22	10331.5	9331.91	7874.07	7589.82	-0.466
	SDL, $\varepsilon = 1$	4435.77	3321.93	3090.58	6791.49	5791.91	4294.41	4035.95	-0.548
	εSC, $\varepsilon = 1$	7372	7372	7372	7372	7372	7372	7372	-
	SynTextBench	0.062	0.027	0.048	0.097	0.075	0.089	0.093	0.871

The instructions we give include two demonstrations with one demonstration for each class. For example, in CR (customer review), we use the instruction: "Answer the sentiment of the following review, either Positive or Negative. \n\nQ: We tried it out christmas night and it worked great .\nA: Positive\n\nQ: very bad quality .\nA:

Negative\n\n".

We give the ICL accuracy in Appendix Table B.18. In Table B.19, we calculate the correlation between the ICL accuracy on SynTextBench synthetic tasks and the average ICL accuracy on subset SentEval tasks. We can see that the ICL accuracy on SynTextBench synthetic tasks shows strong correlation (above 0.8) with ICL accuracy on SentEval tasks. Future research will also be dedicated to investigate whether the success of SynTextBench can be explained by its ability to check the compositional features (e.g.induction head [129]) of transformers.

Table B.18: The detailed subset SentEval in-context learning accuracy on decoder models.

	Transfer tasks						
Models	CR	MR	MPQA	SUBJ	SST2	MRPC	avg.
LLaMA-7B	85.35	90.49	74.34	48.97	88.47	53.86	73.58
LLaMA-13B	91.07	62.78	70.07	50.02	69.74	66.20	68.31
LLaMA-30B	91.97	92.60	83.77	50.01	95.83	66.26	80.07
LLaMA-2-7B	90.83	53.25	47.06	81.60	71.00	66.49	68.37
LLaMA-2-13B	91.84	91.92	80.26	52.73	95.55	66.49	79.80
OPT-13B	90.01	69.66	69.92	49.85	76.99	66.49	70.49
OPT-30B	90.78	82.04	63.56	50.00	87.10	66.61	73.35

Table B.19: Pearson correlation comparison between the in-context learning accuracy on SynTextBench synthetic tasks and the average in-context learning accuracy on the real-world tasks of decoder models.

n	Name	LLaMA-7B	LLaMA-13B	LLaMA-30B	LLaMA-2-7B	LLaMA-2-13B	OPT-13B	OPT-30B	Pearson
	Real-life acc.	73.58	68.31	80.07	68.37	79.80	70.49	73.35	1.0
8192	SynTextBench	50.82	53.43	59.09	51.48	58.83	52.87	51.79	0.813

B.4.3 List of stop words

{'must', 'meanwhile', 'among', 'same', 'you', 'formerly', 'already', 'take', 'he', 'thereupon', 'done', 'anyhow', 'almost', 'ca', 'regarding', 'will', 'mostly', 'say', 'again', 'forty', 'seemed', 'still', 'they', "re", 'seem', 'latter', 'why', 'hers', 'thereby', 'themselves', 'your', 'nine', 'become', 'may', 'beyond', 'it', 'back', 'our', 'himself', "m", 'via', 'we', 'seems', 'throughout', 'yourself', 'bottom', 'only', 'whereby', 'move', 'else', 'front', 'within', 'after', 'every', 'quite', 'hereby', 'now', 'since', 'became', 'herself', 'behind', 'any', 'those', 'used', 'indeed', "'ve", 'first', 'moreover', 'ourselves', 'she', 'should', 'her', 'various', 'few', 'hundred', 'whoever', 'give', 'latterly', 'between', 'in', 'most', 'make', 'sixty',

‘therefore’, "’s", ’hence’, ‘amount’, ‘otherwise’, ‘’m’, ‘’re’, ‘’s’, ‘are’, ‘could’, ‘along’, ‘ours’, ‘of’, ‘that’, ‘everywhere’, ‘during’, ‘his’, ‘then’, ‘fifty’, ‘namely’, ‘when’, ‘around’, ‘all’, ‘keep’, ‘these’, ‘’ll’, ‘third’, ‘being’, ‘thus’, ‘more’, “s’, ‘is’, ‘where’, ‘further’, ‘them’, ‘towards’, ‘next’, ‘and’, ‘a’, ‘does’, ‘here’, ‘ten’, ‘whom’, ‘except’, ‘myself’, ‘somehow’, ‘ever’, ‘enough’, ‘there’, ‘mine’, ‘other’, ‘so’, ‘hereupon’, ‘who’, ‘eight’, ‘one’, ‘hereafter’, ‘amongst’, ‘seeming’, ‘its’, ‘each’, ‘sometime’, ‘this’, ‘me’, “ll’, ‘until’, ‘him’, ‘because’, ‘many’, ‘anyway’, ‘part’, ‘from’, ‘have’, ‘over’, ‘to’, "’re", ’becomes’, ‘too’, ‘as’, ‘name’, ‘whence’, ‘whole’, ‘herein’, ‘everything’, ‘against’, ‘call’, ‘upon’, ‘both’, ‘i’, ‘whenever’, ‘across’, ‘anywhere’, ‘six’, ‘us’, ‘thereafter’, ‘also’, ‘former’, ‘whither’, ‘whose’, ‘such’, ‘really’, ‘was’, ‘’d’, ‘someone’, “ve’, ‘eleven’, ‘wherein’, ‘yours’, ‘by’, ‘their’, ‘beside’, ‘or’, ‘re’, ‘has’, ‘off’, ‘which’, ‘put’, ‘whether’, ‘per’, ‘four’, ‘whereafter’, ‘often’, ‘doing’, ‘had’, ‘out’, ‘some’, ‘fifteen’, ‘others’, ‘once’, ‘somewhere’, ‘either’, ‘besides’, ‘though’, ‘been’, ‘do’, ‘very’, ‘thru’, ‘go’, ‘please’, ‘sometimes’, "’ll", ’perhaps’, ‘whereupon’, ‘whatever’, ‘about’, ‘for’, ‘itself’, ‘thence’, ‘at’, ‘how’, ‘made’, ‘three’, ‘might’, ‘another’, ‘did’, ‘alone’, ‘elsewhere’, ‘toward’, ‘were’, ‘would’, ‘due’, ‘what’, ‘an’, ‘wherever’, ‘be’, ‘can’, ‘something’, ‘side’, "’d", ’with’, "’m", ’am’, ‘therein’, ‘into’, ‘through’, "’ve", ’everyone’, ‘on’, ‘my’, ‘even’, ‘own’, ‘see’, ‘several’, ‘two’, ‘afterwards’, ‘show’, “d’, ‘beforehand’, ‘nowhere’, ‘becoming’, ‘last’, ‘onto’, ‘the’, ‘yourselves’, ‘five’, ‘anyone’, ‘together’, ‘before’, ‘always’, ‘get’, ‘using’}

B.4.4 Experimental details

When we calculate the correlation between real-data-free evaluation methods and real-world task robustness-accuracy performance, we need to aggregate two metrics - accuracy and robustness. For this purpose, we can obtain a ranking of the models according to the accuracy measure, R_1, and a ranking of the models according to the robustness measure, R_2. We aggregate two rankings by the simple and commonly-used mean aggregation[1] which yields the overall ranking of models based on accuracy-

[1]Wald, R., Khoshgoftaar, T.M. and Dittman, D., 2012, December. Mean aggregation versus robust rank aggregation for ensemble gene selection. In 2012 11th international conference on machine learning and applications (Vol. 1, pp. 63-69). IEEE.

robustness performance, R_{ref}. On the other hand, we can obtain another ranking of models based on one of the real-data-free evaluation methods (e.g. Val loss, MDL, SDL, ϵSC, SynTextBench), R. Lastly, we calculate the Pearson correlation coefficient between R and R_{ref}.

Moreover, when we calculate the robustness measures, we only perform attacks on Transfer tasks as they are classification tasks where adversarial attacks are well-defined. Since we use the average number of perturbed words by PWWS attacks [151] as the robustness indicator, we also excluded MPQA and TREC due to their short sentence lengths (MPQA and TREC average sentence lengths are 3.03 and 6.48, respectively). PWWS attacks focus on the text adversarial example generation that could guarantee little semantic shifting and therefore rarely cause ground truth label change (also lexical and grammatical correctness). To meet the semantic constraint, PWWS replaces words in the input texts with synonyms and replace named entities (NEs) with similar NEs to generate adversarial samples. Synonyms for each word can be found in WordNet, a large lexical database for the English language. NE refers to an entity that has a specific meaning in the sample text, such as a person's name, a location, an organization, or a proper noun. Replacement of an NE with a similar NE imposes a slight change in semantics but invokes no lexical or grammatical changes.

We list the robustness results in the following table:

Table B.20: The robustness (average number of perturbed words) of pretrained representations on Transfer tasks.

Models	MR	CR	SUBJ	SST	MRPC	avg.
$\text{BERT}_{\text{base}}$	14.48	13.99	20.2	15.07	5.45	13.838
DiffCSE-B	14.46	14.7	18.64	15.19	6.39	13.876
$\text{BERT}_{\text{large}}$	14.3	14.22	19.87	15.46	5.26	13.822
T5_{base}	12.71	12.82	16.8	13.66	5.05	12.208
T5_{large}	13.67	14.28	16.93	13.82	5.17	12.774
$\text{RoBERTa}_{\text{base}}$	16.4	18.35	20.74	17.26	7.12	15.974
DiffCSE-R	15.72	16.07	18.53	16.82	5.68	14.564
GPT	12.53	13.11	15.75	13.52	5.17	12.016
ST5	13.6	13.08	18.36	14.22	6.9	13.232

We also list the ranking of models from different metrics in the following table.

For example, to calculate SynTextBench correlation with robustness-and-accuracy performance, we calculate the Pearson correlation between (row "Overall accuracy" +

Table B.21: Ranking of models from different metrics at $n = 8192$.

Name	$\text{BERT}_{\text{base}}$	DiffCSE-B	$\text{BERT}_{\text{large}}$	T5_{base}	T5_{large}	$\text{RoBERTa}_{\text{base}}$	DiffCSE-R	GPT	ST5
Overall accuracy	6	3	5	7	8	4	2	9	1
STS accuracy	7	3	8	4	5	6	2	9	1
Transfer accuracy	5	6	2	8	9	4	3	7	1
Robustness	4	3	5	8	7	1	2	9	6
Val loss	8	4	3	9	5	6	7	1	2
MDL	7	8	3	1	2	5	6	4	9
SDL, ε=1	7	8	3	1	2	5	6	4	9
εSC, ε=1	5	5	5	5	5	5	5	5	5
SynTextBench	4	3	5	6	8	7	2	9	1

row "Robustness") / 2 and "SynTextBench". To calculate SynTextBench correlation with robustness-and-STS accuracy performance, we calculate the Pearson correlation between (row "STS accuracy" + row "Robustness") / 2 and "SynTextBench". To calculate SynTextBench correlation with robustness-and-Transfer accuracy performance, we calculate the Pearson correlation between (row "Transfer accuracy" + row "Robustness") / 2 and "SynTextBench". We note that in all our results prior to Table B.21, we always infer the correlation in individual runs before we take an average over all trials. Different from that, the rankings from Val loss, MDL, SDL, ϵSC, and SynTextBench in Table B.21, are inferred from the average metric results over 3 trails for an easier illustration. Therefore, the ranking correlation suggested by the table might have some deviation from what is shown in Table 6.3.

Bibliography

[1] Alberto Acerbi and Joseph M Stubbersfield. Large language models show human-like content biases in transmission chain experiments. In *Proceedings of the National Academy of Science*, 2023.

[2] Pranjal Awasthi, Natalie Frank, and Mehryar Mohri. On the existence of the adversarial bayes classifier. *Advances in Neural Information Processing Systems*, 34:2978–2990, 2021.

[3] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA).

[4] Arjun Nitin Bhagoji, Daniel Cullina, and Prateek Mittal. Lower bounds on adversarial robustness from optimal transport. *Advances in Neural Information Processing Systems*, 32, 2019.

[5] Kush Bhatia, Avanika Narayan, Christopher De Sa, and Christopher Ré. Tart: A plug-and-play transformer module for task-agnostic reasoning. *arXiv preprint arXiv:2306.07536*, 2023.

[6] Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raf. Emergent and predictable memorization in large language models. *arXiv preprint arXiv:2304.11158*, 2023.

[7] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402, 2013.

[8] Léonard Blier and Yann Ollivier. The description length of deep learning models. *Advances in Neural Information Processing Systems*, 31, 2018.

[9] Su Lin Blodgett and Michael Madaio. Risks of ai foundation models in education. *arXiv preprint arXiv:2110.10024*, 2021.

[10] Avrim Blum, Travis Dick, Naren Manoj, and Hongyang Zhang. Random smoothing might be unable to certify l_∞ robustness for high-dimensional images. *arXiv preprint arXiv:2002.03517*, 2020.

[11] Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. In *ICML*, pages 517–526. PMLR, 2017.

[12] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[13] Margaret M Bradley and Peter J Lang. Affective norms for english words (anew): Instruction manual and affective ratings. Technical report, Technical report C-1, the center for research in psychophysiology ..., 1999.

[14] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *SP*, 2017.

[15] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security Symposium*, volume 267, 2019.

[16] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *USENIX Security Symposium*, volume 6, 2021.

[17] Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 1–7, 2018.

[18] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019.

[19] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[20] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[21] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In Hal Daumé III

and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR, 13–18 Jul 2020.

[22] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: elastic-net attacks to deep neural networks via adversarial examples. *AAAI*, 2018.

[23] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607, Virtual, 13–18 Jul 2020. PMLR.

[24] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020.

[25] Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations. In *International Conference on Learning Representations*, 2021.

[26] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297, 2020.

[27] Cheng-Han Chiang and Hung-yi Lee. Pre-training a language model without human language. *arXiv preprint arXiv:2012.11995*, 2020.

[28] Cheng-Han Chiang and Hung-yi Lee. On the transferability of pre-trained language models: A study from artificial datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10518–10525, 2022.

[29] Ching-Yao Chuang, Joshua Robinson, Lin Yen-Chen, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *arXiv preprint arXiv:2007.00224*, 2020.

[30] Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljačić, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass. Diffcse: Difference-based contrastive learning for sentence embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4207–4218, 2022.

[31] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *ICML*, 2019.

[32] Alexis Conneau and Douwe Kiela. Senteval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[33] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, 2017.

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

[35] A Dadhich and B Thankachan. Opinion classification of product reviews using naïve bayes, logistic regression and sentiwordnet: Challenges and survey. In *IOP Conference Series: Materials Science and Engineering*, volume 1099, page 012071. IOP Publishing, 2021.

[36] Chen Dan, Yuting Wei, and Pradeep Ravikumar. Sharp statistical guaratees for adversarially robust gaussian classification. In *International Conference on Machine Learning*, pages 2345–2355. PMLR, 2020.

[37] Kerstin Denecke. Using sentiwordnet for multilingual sentiment analysis. In *2008 IEEE 24th international conference on data engineering workshop*, pages 507–512. IEEE, 2008.

[38] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[39] Edgar Dobriban, Hamed Hassani, David Hong, and Alexander Robey. Provable tradeoffs in adversarially robust classification. *arXiv preprint arXiv:2006.05161*, 2020.

[40] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[41] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1734–1747, 2015.

[42] Gamaleldin Elsayed, Shreya Shankar, Brian Cheung, Nicolas Papernot, Alexey Kurakin, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial examples that fool both computer vision and time-limited humans. In *NeurIPS*, 2018.

[43] Andrea Esuli. Automatic generation of lexical resources for opinion mining: models, algorithms and applications. In *Acm sigir forum*, volume 42, pages 105–106. ACM New York, NY, USA, 2008.

[44] Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, 2006.

[45] Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: a high-coverage lexical resource for opinion mining. *Evaluation*, 2007.

[46] Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, 2019.

[47] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. Robust physical-world attacks on deep learning visual classification. In *CVPR*, pages 1625–1634, 2018.

[48] Antonella Falini. A review on the selection criteria for the truncated svd in data science applications. *Journal of Computational Mathematics and Data Science*, page 100064, 2022.

[49] Lijie Fan, Sijia Liu, Pin-Yu Chen, Gaoyuan Zhang, and Chuang Gan. When does contrastive learning preserve adversarial robustness from pretraining to finetuning?, 2021.

[50] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023.

[51] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2020.

[52] Rebecca LA Frost, Andrew Jessop, Samantha Durrant, Michelle S Peter, Amy Bidgood, Julian M Pine, Caroline F Rowland, and Padraic Monaghan. Non-adjacent dependency learning in infancy, and its link to language development. *Cognitive Psychology*, 120:101291, 2020.

[53] Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*, 2019.

[54] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, 2021.

[55] Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and William B Dolan. Dialogue response ranking training with large-scale human feedback data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 386–395, 2020.

[56] T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. Vechev. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *IEEE Symposium on Security and Privacy (SP)*, volume 00, pages 948–963, 2018.

[57] John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. Declutr: Deep contrastive learning for unsupervised textual representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 879–895, 2021.

[58] Shahriar Golchin and Mihai Surdeanu. Time travel in LLMs: Tracing data contamination in large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[59] Jacob Goldberger, Geoffrey E Hinton, Sam Roweis, and Russ R Salakhutdinov. Neighbourhood components analysis. *NeurIPS*, 17:513–520, 2004.

[60] I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

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

[62] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In *NeurIPS*, volume 33, pages 21271–21284, 2020.

[63] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

[64] Grzegorz Głuch and Rüdiger Urbanke. Constructing a provably adversarially-robust classifier from a high accuracy one, 2019.

[65] Jeff Z HaoChen and Tengyu Ma. A theoretical study of inductive biases in contrastive learning. *arXiv preprint arXiv:2211.14699*, 2022.

[66] Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *arXiv preprint arXiv:2106.04156*, 2021.

[67] Marc D Hauser, Noam Chomsky, and W Tecumseh Fitch. The faculty of language: what is it, who has it, and how did it evolve? *science*, 298(5598):1569–1579, 2002.

[68] Monson H Hayes. *Statistical digital signal processing and modeling*. John Wiley & Sons, 1996.

[69] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, June 2020.

[70] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[71] David Heckerman. Bayesian networks for data mining. *Data mining and knowledge discovery*, 1:79–119, 1997.

[72] Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *NeurIPS*, 2017.

[73] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[74] Sorami Hisamoto, Matt Post, and Kevin Duh. Membership inference attacks on sequence-to-sequence models: Is my data in your machine translation system? *Transactions of the Association for Computational Linguistics*, 8:49–63, 2020.

[75] Chih-Hui Ho and Nuno Vasconcelos. Contrastive learning with adversarial examples. *arXiv preprint arXiv:2010.12050*, 2020.

[76] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[77] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, 2004.

[78] Mujtaba Husnain, Malik Muhammad Saad Missen, Nadeem Akhtar, Mickaël Coustaty, Shahzad Mumtaz, and VB Surya Prasath. A systematic study on the role of sentiwordnet in opinion mining. *Frontiers of Computer Science*, 15(4):154614, 2021.

[79] Geng Ji, Michael C Hughes, and Erik B Sudderth. From patches to images: a nonparametric generative model. In *International Conference on Machine Learning*, pages 1675–1683. PMLR, 2017.

[80] Jinyuan Jia, Xiaoyu Cao, Binghui Wang, and Neil Zhenqiang Gong. Certified robustness for top-k predictions against adversarial perturbations via randomized smoothing. In *International Conference on Learning Representations*, 2019.

[81] Richard Arnold Johnson, Dean W Wichern, et al. *Applied multivariate statistical analysis*. Prentice hall Upper Saddle River, NJ, 2002.

[82] Matt Jordan, Justin Lewis, and Alexandros G Dimakis. Provable certificates for adversarial examples: Fitting a ball in the union of polytopes. In *NeurIPS*, 2019.

[83] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016.

[84] Rudolph Emil Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960.

[85] Guy Katz, Clark Barrett, David L Dill, et al. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017.

[86] Pei Ke, Haozhe Ji, Siyang Liu, Xiaoyan Zhu, and Minlie Huang. SentiLARE: Sentiment-aware language representation learning with linguistic knowledge. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6975–6988, Online, November 2020. Association for Computational Linguistics.

[87] Farhan Hassan Khan, Usman Qamar, and Saba Bashir. Sentimi: Introducing point-wise mutual information with sentiwordnet to improve sentiment polarity detection. *Applied Soft Computing*, 39:140–153, 2016.

[88] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

[89] Minseon Kim, Jihoon Tack, and Sung Ju Hwang. Adversarial self-supervised contrastive learning. *arXiv preprint arXiv:2006.07589*, 2020.

[90] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[91] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. *Advances in neural information processing systems*, 28, 2015.

[92] Kundan Krishna, Jeffrey P Bigham, and Zachary C Lipton. Does pretraining for summarization require knowledge transfer? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3178–3189, 2021.

[93] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[94] Aounon Kumar, Alexander Levine, Tom Goldstein, and Soheil Feizi. Curse of dimensionality on randomized smoothing for certifiable robustness. *arXiv preprint arXiv:2002.03239*, 2020.

[95] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *ICLR Workshop*, 2017.

[96] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[97] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, 2017.

[98] Yann LeCun. The mnist database of handwritten digits, 1998.

[99] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 656–672. IEEE, 2019.

[100] Guang-He Lee, Yang Yuan, Shiyu Chang, and Tommi S Jaakkola. Tight certificates of adversarial robustness for randomly smoothed classifiers. *arXiv preprint arXiv:1906.04948*, 2019.

[101] Kibok Lee, Yian Zhu, Kihyuk Sohn, Chun-Liang Li, Jinwoo Shin, and Honglak Lee. i-mix: A domain-agnostic strategy for contrastive representation learning. In *ICLR*, 2021.

[102] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive data sets*. Cambridge university press, 2020.

[103] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive noise. In *NeurIPS*, 2019.

[104] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, 2020.

[105] Haoran Li, Mingshi Xu, and Yangqiu Song. Sentence embedding leaks more information than you expect: Generative embedding inversion attack to recover the whole sentence. *arXiv preprint arXiv:2305.03010*, 2023.

[106] Linyi Li, Xiangyu Qi, Tao Xie, and Bo Li. Sok: Certified robustness for deep neural networks. *arXiv preprint arXiv:2009.04131*, 2020.

[107] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022.

[108] Yucheng Li. Estimating contamination via perplexity: Quantifying memorisation in language model evaluation. *arXiv e-prints*, pages arXiv–2309, 2023.

[109] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

[110] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[111] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 369–385, 2018.

[112] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[113] Bruno Loureiro, Gabriele Sicuro, Cédric Gerbelot, Alessandro Pacco, Florent Krzakala, and Lenka Zdeborová. Learning gaussian mixtures with generalized linear models: Precise asymptotics in high-dimensions. *Advances in Neural Information Processing Systems*, 34:10144–10157, 2021.

[114] Zhaoyang Lyu, Ching-Yun Ko, Zhifeng Kong, Ngai Wong, Dahua Lin, and Luca Daniel. Fastened crown: Tightened neural network robustness certificates. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5037–5044, 2020.

[115] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[116] Prasanta Chandra Mahalanobis. On the generalised distance in statistics. In *Proceedings of the National Institute of Science of India*, volume 12, pages 49–55, 1936.

[117] Christopher D Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054, 2020.

[118] Francesca Mignacco, Florent Krzakala, Yue Lu, Pierfrancesco Urbani, and Lenka Zdeborova. The role of regularization in classification of high-dimensional noisy gaussian mixture. In *International conference on machine learning*, pages 6874–6883. PMLR, 2020.

[119] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[120] Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. Quantifying privacy risks of masked language models using membership inference attacks. *arXiv preprint arXiv:2203.03929*, 2022.

[121] Andriy Mnih and Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models. In *ICML*, 2012.

[122] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, 2016.

[123] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? *Advances in neural information processing systems*, 33:512–523, 2020.

[124] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, pages 427–436, 2015.

[125] Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. Sentence-t5: Scalable sentence encoders from pretrained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, 2022.

[126] Finn Årup Nielsen. A new anew: Evaluation of a word list for sentiment analysis in microblogs. In *Workshop on'Making Sense of Microposts: Big things come in small packages*, pages 93–98, 2011.

[127] Bruno Ohana and Brendan Tierney. Sentiment classification of reviews using sentiwordnet. *Proceedings of IT&T*, 8, 2009.

[128] Gabriel Iluebe Okolo, Stamos Katsigiannis, and Naeem Ramzan. Ievit: An enhanced vision transformer architecture for chest x-ray image classification. *Computer Methods and Programs in Biomedicine*, 226:107141, 2022.

[129] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

[130] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[131] Yonatan Oren, Nicole Meister, Niladri S. Chatterji, Faisal Ladhak, and Tatsunori Hashimoto. Proving test set contamination for black-box language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[132] Isabel Papadimitriou and Dan Jurafsky. Learning music helps you read: Using transfer to study linguistic structure in language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6829–6839, 2020.

[133] Isabel Papadimitriou and Dan Jurafsky. Pretrain on just structure: Understanding linguistic inductive biases using transfer learning. *arXiv preprint arXiv:2304.13060*, 2023.

[134] Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2071–2081, 2022.

[135] Luca Pesce, Florent Krzakala, Bruno Loureiro, and Ludovic Stephan. Are gaussian data all you need? extents and limits of universality in high-dimensional generalized linear estimation. *arXiv preprint arXiv:2302.08923*, 2023.

[136] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[137] Sergios Petridis and Stavros J Perantonis. On the relation between discriminant analysis and mutual information for supervised linear feature extraction. *Pattern Recognition*, 37(5):857–874, 2004.

[138] Tomaso Poggio and Federico Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[139] Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel Bowman.

Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5231–5247, 2020.

[140] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12556–12565, 2020.

[141] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[142] Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

[143] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[144] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. *ICLR*, 2018.

[145] Jagath C Rajapakse, Jay N Giedd, and Judith L Rapoport. Statistical approach to segmentation of single-channel cerebral mr images. *IEEE transactions on medical imaging*, 16(2):176–186, 1997.

[146] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 784–789, 2018.

[147] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016.

[148] Vijjini Anvesh Rao, Kaveri Anuranjana, and Radhika Mamidi. A sentiwordnet strategy for curriculum learning in sentiment analysis. In *Natural Language Processing and Information Systems: 25th International Conference on Applications of Natural Language to Information Systems, NLDB 2020, Saarbrücken, Germany, June 24–26, 2020, Proceedings 25*, pages 170–178. Springer, 2020.

[149] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[150] Maria Refinetti, Sebastian Goldt, Florent Krzakala, and Lenka Zdeborová. Classifying high-dimensional gaussian mixtures: Where kernel methods fail and neural networks succeed. In *International Conference on Machine Learning*, pages 8936–8947. PMLR, 2021.

[151] Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy, July 2019. Association for Computational Linguistics.

[152] Rezvaneh Rezapour, Saumil H Shah, and Jana Diesner. Enhancing the measurement of social effects by capturing morality. In *Proceedings of the tenth workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 35–45, 2019.

[153] Ryokan Ri and Yoshimasa Tsuruoka. Pretraining with artificial language: Studying transferable knowledge in language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7302–7315, 2022.

[154] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *ICLR*, 2021.

[155] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[156] Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems*, pages 11289–11300, 2019.

[157] S Sanjay-Gopal and Thomas J Hebert. Bayesian pixel classification using spatially variant finite mixtures and the generalized em algorithm. *IEEE Transactions on Image Processing*, 7(7):1014–1028, 1998.

[158] Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. A theoretical analysis of contrastive unsupervised representation learning. In *ICML*, pages 5628–5637, 2019.

[159] Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? *arXiv preprint arXiv:2304.15004*, 2023.

[160] Samira Shaikh, Kit Cho, Tomek Strzalkowski, Laurie Feldman, John Lien, Ting Liu, and George Aaron Broadwell. Anew+: Automatic expansion and validation of affective norms of words lexicons in multiple languages. In *Proceedings of*

the Tenth International Conference on Language Resources and Evaluation (LREC'16), pages 1127–1132, 2016.

[161] Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. On the adversarial robustness of vision transformers. *arXiv preprint arXiv:2103.15670*, 2021.

[162] Wenqi Shao, Xun Zhao, Yixiao Ge, Zhaoyang Zhang, Lei Yang, Xiaogang Wang, Ying Shan, and Ping Luo. Not all models are equal: predicting model transferability in a self-challenging fisher space. In *European Conference on Computer Vision*, pages 286–302. Springer, 2022.

[163] Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[164] Yuge Shi, Imant Daunhawer, Julia E. Vogt, Philip H. S. Torr, and Amartya Sanyal. How robust is unsupervised representation learning to distribution shift?, 2022.

[165] Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin Vechev. Fast and effective robustness certification. In *NeurIPS*, 2018.

[166] Aman Sinha, Hongseok Namkoong, and John Duchi. Certifying some distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018.

[167] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models, 2022.

[168] Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models. *arXiv preprint arXiv:1808.01688*, 2018.

[169] Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*, 2021.

[170] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *ICLR*, 2014.

[171] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[172] Daniel Tarlow, Kevin Swersky, Laurent Charlin, Ilya Sutskever, and Rich Zemel. Stochastic k-neighborhood selection for supervised and unsupervised learning. In *ICML*, 2013.

[173] Jiaye Teng, Guang-He Lee, and Yang YuanTeng J. $\ell 1$ adversarial robustness certificates: a randomized smoothing approach, 2019.

[174] Anja Thieme, Aditya Nori, Marzyeh Ghassemi, Rishi Bommasani, Tariq Osman Andersen, and Ewa Luger. Foundation models in healthcare: Opportunities, risks & strategies forward. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–4, 2023.

[175] Aleena Thomas, David Ifeoluwa Adelani, Ali Davody, Aditya Mogadala, and Dietrich Klakow. Investigating the impact of pre-trained word embeddings on memorization in neural networks. In *Text, Speech, and Dialogue: 23rd International Conference, TSD 2020, Brno, Czech Republic, September 8–11, 2020, Proceedings 23*, pages 273–281. Springer, 2020.

[176] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

[177] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In *NeurIPS*, volume 33, pages 6827–6839, 2020.

[178] Vincent Tjeng, Kai Y Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. In *International Conference on Learning Representations*, 2018.

[179] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[180] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[181] Dustin Tran, Jeremiah Liu, Michael W Dusenberry, Du Phan, Mark Collier, Jie Ren, Kehang Han, Zi Wang, Zelda Mariet, Huiyi Hu, et al. Plex: Towards reliability using pretrained large model extensions. *arXiv preprint arXiv:2207.07411*, 2022.

[182] Sudhakar Tummala, Seifedine Kadry, Syed Ahmad Chan Bukhari, and Hafiz Tayyab Rauf. Classification of brain tumor from magnetic resonance

imaging using vision transformers ensembling. *Current Oncology*, 29(10):7498–7511, 2022.

[183] Zoltán Tüske, Muhammad Ali Tahir, Ralf Schlüter, and Hermann Ney. Integrating gaussian mixtures into deep neural networks: Softmax layer with hidden variables. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4285–4289. IEEE, 2015.

[184] Vikas Verma, Thang Luong, Kenji Kawaguchi, Hieu Pham, and Quoc Le. Towards domain-agnostic contrastive learning. In *International Conference on Machine Learning*, pages 10530–10541. PMLR, 2021.

[185] Elena Voita and Ivan Titov. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, 2020.

[186] Vaclav Voracek and Matthias Hein. Improving l1-certified robustness via randomized smoothing by leveraging box constraints. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 35198–35222. PMLR, 23–29 Jul 2023.

[187] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

[188] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2018.

[189] Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. Efficient formal safety analysis of neural networks. In *NeurIPS*, 2018.

[190] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 9929–9939, Virtual, 13–18 Jul 2020. PMLR.

[191] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2015.

[192] Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45:1191–1207, 2013.

[193] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022.

[194] Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Duane Boning, Inderjit S Dhillon, and Luca Daniel. Towards fast computation of certified robustness for relu networks. *ICML*, 2018.

[195] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, et al. Evaluating the robustness of neural networks: An extreme value theory approach. *ICLR*, 2018.

[196] William F Whitney, Min Jae Song, David Brandfonbrener, Jaan Altosaar, and Kyunghyun Cho. Evaluating representations by the complexity of learning low-loss predictors. *arXiv preprint arXiv:2009.07368*, 2020.

[197] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

[198] Benjamin Wilson, Michelle Spierings, Andrea Ravignani, Jutta L Mueller, Toben H Mintz, Frank Wijnen, Anne Van der Kant, Kenny Smith, and Arnaud Rey. Non-adjacent dependency learning in humans and other animals. *Topics in cognitive science*, 12(3):843–858, 2020.

[199] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *ICML*, 2018.

[200] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2020.

[201] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pages 23965–23998. PMLR, 2022.

[202] Xiongwei Wu, Xin Fu, Ying Liu, Ee-Peng Lim, Steven CH Hoi, and Qianru Sun. A large-scale benchmark for food image segmentation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 506–515, 2021.

[203] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018.

[204] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.

[205] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.

[206] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020.

[207] Greg Yang, Tony Duan, J Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes. In *International Conference on Machine Learning*, pages 10693–10705. PMLR, 2020.

[208] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[209] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6210–6219, 2019.

[210] Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. Logme: Practical assessment of pre-trained models for transfer learning. In *International Conference on Machine Learning*, pages 12133–12143. PMLR, 2021.

[211] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

[212] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. Swag: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, 2018.

[213] Huimin Zeng, Chen Zhu, Tom Goldstein, and Furong Huang. Are adversarial examples created equal? a learnable weighted minimax risk for robustness under non-uniform attacks. *arXiv preprint arXiv:2010.12989*, 2020.

[214] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[215] Dinghuai Zhang*, Mao Ye*, Chengyue Gong*, Zhanxing Zhu, and Qiang Liu. Filling the soap bubbles: Efficient black-box adversarial certification with non-gaussian smoothing, 2020.

[216] Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Comput. Surv.*, 56(3), oct 2023.

[217] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[218] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *NeurIPS*, 2018.

[219] Michael Zhang and Christopher Ré. Contrastive adapters for foundation model group robustness. *arXiv preprint arXiv:2207.07180*, 2022.

[220] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[221] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018.

[222] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *2011 international conference on computer vision*, pages 479–486. IEEE, 2011.

[223] Daniel Zoran and Yair Weiss. Natural images, gaussian mixtures and dead leaves. *Advances in Neural Information Processing Systems*, 25, 2012.