Towards Understanding Self-Supervised Representation Learning

Nikunj Saunshi

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Computer Science
Adviser: Sanjeev Arora

September 2022

Abstract

While supervised learning sparked the deep learning boom, it has some critical shortcomings: (1) it requires an abundance of expensive labeled data, and (2) it solves tasks from scratch rather than the human-like approach of leveraging knowledge and skills acquired from prior experiences. Pre-training has emerged as an alternative and effective paradigm, to overcome these shortcomings, whereby a model is first trained using easily acquirable data, and later used to solve downstream tasks of interest with much fewer labeled data than supervised learning. Pre-training using unlabeled data, a.k.a. self-supervised learning, has been especially revolutionary, with successes in diverse domains: text, vision, speech, etc. This raises an interesting and challenging question: why should pre-training on unlabeled data help with seemingly unrelated downstream tasks?

In this thesis we present works that initiate and build a theoretical framework to study why self-supervised learning is beneficial for downstream tasks. The framework is applied to methods like contrastive learning, auto-regressive language modeling and self-prediction based methods. Central to the framework is the idea that pre-training helps learn low-dimensional representations of data, that subsequently help solve downstream tasks of interest with linear classifiers, requiring fewer labeled data. A common theme is to formalize what are desirable properties of the unlabeled data distribution that is used to construct the self-supervised learning task. Under appropriate formalizations, it can be shown that approximately minimizing the right pre-training objectives can extract the downstream signal that is implicitly encoded in the unlabeled data distribution. Finally it is shown that this signal can be decoded from the learned representations using linear classifiers, thus providing a formalization for transference of "skills and knowledge" across tasks.

Acknowledgements

Writing this section has gladly taken me down many memory lanes over the last 6 years. Firstly I would like to express my utmost gratitude to my adviser, Prof. Sanjeev Arora, whose guidance and mentorship has been invaluable in my research and grad school journey. His research philosophy of diving into challenging unexplored problems, seeking to find the simplest and "truest" explanations, and emphasis on crisp and effective communication of complex ideas[1] have had a great influence on me as a researcher. He has been very generous with his time and it has been a lot of fun, and a privilege, to pick his brain on various topics, within and outside of research. I have been very fortunate to have him as my adviser.

I am very grateful to Akshay Krishnamurthy, Elad Hazan, Jason Lee, Sham Kakade, Karthik Narasimhan, Danqi Chen and Chi Jin for their valuable mentorship and feedback in various stages of my PhD. Special thanks to Pravesh Kothari and Samory Kpotufe for encouraging me to apply to PhD programs.

I had the pleasure and fortune of collaborating with many brilliant researchers from whom I learned a ton: Sanjeev Arora, Jordan T. Ash, Simon S. Du, Surbhi Goel, Arushi Gupta, Wei Hu, Sham Kakade, Hrishi Khandeparkar, Misha Khodak, Akshay Krishnamurthy, Jason Lee, Qi Lei, Yingyu Liang, Yuping Luo, Kaifeng Lyu, Tengyu Ma, Sadhika Malladi, Dipendra Misra, Orestis Plevrakis, Brandon Stewart, Kiran Vodrahalli, Dingli Yu, Cyril Zhang, Yi Zhang, Jiacheng Zhuo. I am also grateful for the insightful and fun conversations with Brian Bullins, Xinyi Chen, Matheus Ferreira, Holden Lee, Zhiyuan Li, Edgar Minasyan, Abhishek Panigrahi, Andrej Risteski and Haoyu Zhao. I learned a lot through AlgML, Gems, theory talk series and am also thankful for them violating the no free lunch principle. I would also like to thank the administrative staff in Davis IC and the Computer Science department, especially Mitra Kelly and Nicki Mahler. The work in this thesis is supported by Sanjeev Arora's grants from NSF, ONR, the Simons Foundation, the Schmidt Foundation, Mozilla Research, Amazon Research, DARPA, and SRC, and also a Siebel scholarship.

A special thanks to Hrishi, Orestis, Wei and Yi for all the insightful and fun discussions in the office, during meals, while having coffee, while making coffee, etc. I am grateful for the friendship and collaborations with Misha and Kiran, through the pre and post Boffini eras. I would also like to thank Cyril, Surbhi and Yi for all the discussion (research, philosophical, random) and the valuable help through job search.

A crucial aspect of my life at Princeton has been the sports clubs and I am grateful to the Princeton Badminton Club for the fun overnight tourneys with the team, the team spirit of the Cricket Club team and

[1] Not to forget, appreciation for good espresso

the constancy of GradFC.

The grad school journey was made fun and exciting by the many friends here: Akash, Akshay K, Akshay Y, Arjun, Gopi, Karan, Naman, Niran, Sanjay, Sravya, Sumegha, Visu, Yatin. I have been extremely lucky to have gotten closer to the 45 group[2] during the pandemic: Divya, Meera, Nivedita, Pranav and Vivek. Special thanks to Divya and Sravya for all the dinner sessions and bearing with my jokes, Vivek and Pranav for all the magical discussions, and Meera for the projections. Last but not the least, I am grateful to Nivedita for all the bits, constant entertainment and the unfinished duets that we hope to finish in the coming years.

My family has been a constant source of love and support, including the SUN and NASA groups, with the newest member Ayu being the reason for many smiles. I would like to dedicate this thesis to my parents, Nirmala Saunshi and Umesh Saunshi, and to my sister Shrutika Saunshi, because this would have been impossible if not for their support, the values they instilled in me, and all the sacrifices they have made.

[2]in need of unhexing

Contents

Chapter 1

Introduction

In the quest to design intelligent agents and data-driven solutions to problems, the field of machine learning and AI has made tremendous advances in the last decade. Sparked with initial successes on challenging supervised learning benchmarks like ImageNet [Deng et al., 2009], innovations in deep learning have subsequently led to models with super-human performances on many such benchmarks across various domains. Training such task-specific models is certainly impressive and has immense utility. However it suffers from an important limitation of requiring large labeled or annotated datasets, which is often expensive to obtain. Additionally, from the standpoint of intelligence, one would hope for more general purpose models which, like humans [Ahn and Brewer, 1993], can learn from prior experiences, summarize them into skills or concepts and leverage those to solve new tasks with very few or no demonstrations. After all babies learn a lot through observations and interactions in the world without explicit supervision. These limitations have inspired the alternative paradigm of *pre-training*.

The focus on this thesis is on pre-training using *unlabeled data*, which is often available in abundance. The idea of using unlabeled data has always been of interest in machine learning, specifically through unsupervised learning and semi-supervised learning. The modern adaptation of this using deep learning is popularly termed as *self-supervised learning* (SSL) and has begun to change the landscape of machine learning and AI through ideas like contrastive learning and language modeling. The idea of self-supervised learning is to construct certain tasks using just unlabeled data, and train a model to do well on the constructed tasks. Such tasks often require the model to encode structural properties of the data through predicting unobserved

or hidden parts (or properties) of the input from the observed or retained parts [LeCun and Misra, 2021]. Self-supervised learning has demonstrated versatility and utility on many *downstream tasks* of interest, often with better sample efficiency than solving the tasks from scratch, thus bringing us a step closer to the goal of general purpose intelligent agents. In fact more recently, large language models like GPT-3 [Brown et al., 2020] and others have demonstrated fascinating "emergent behaviors" that arise at scale, fueling more interest in the idea of self-supervised pre-training.

Although self-supervised learning has enjoyed empirical successes and continues to show great promise, a good theoretical understanding of why it works, besides rough intuitions, is still lacking. These impressive successes raise intriguing questions, since it is a priori unclear why a model trained on one task should help with a different and seemingly unrelated task, i.e. why training on *task A* should help with *task B*. While a complete theoretical understanding of SSL (and deep learning in general) is challenging and elusive, understanding such phenomena at any level of abstraction could be helpful in developing more principled algorithms. This thesis is motivated by the following questions:

Why does training on a **self-supervised learning task** *(using abundant unlabeled data) help with solving a data-scarce* **downstream task***? How does one formalize transferring of "knowledge & skills"?*

While there is extensive and rich literature on supervised learning, the generalization from an SSL task $\rightarrow$ downstream task is fundamentally different from the generalization from train set $\rightarrow$ test set in supervised learning. For supervised learning on a downstream task of classification, for instance, a model trained on the train set of input-label pairs, sampled from an unknown distribution, can be directly used for evaluation on the unseen test set sampled from the same distribution. This underlying distribution is what establishes a connection from the train set $\rightarrow$ test set. However the conceptual connection from an SSL task $\rightarrow$ downstream task is much less clear, since the *unlabeled data used in the SSL task has no explicit signal about the downstream labels.* An implication of this is that a model pre-trained on an SSL task (e.g. predicting part of an input from the rest of it) cannot be directly used for a downstream task (e.g. predicting class label from an input). Hence the *transferring of "knowledge and skills" requires additional training step using some labeled data*, ideally lesser than what supervised learning from scratch would require. Any theoretical understanding of the generalization from an SSL task $\rightarrow$ downstream task will need to address these questions: "what is the inherent role of unlabeled data?" and "how to use a pre-trained model for the downstream task?" In this thesis, we study these questions, for the downstream task of classification, by making distributional assumptions about the unlabeled data and leveraging the idea of representation learning as follows:

(a) (Distributional assumptions) Unlabeled data distribution *implicitly* contains information about the downstream classification task(s) of interest.

(b) (Representation learning) Models pre-trained on an *appropriate SSL task* can encode this signal through *learned representations* that can subsequently solve downstream classification tasks with *linear classifiers*.

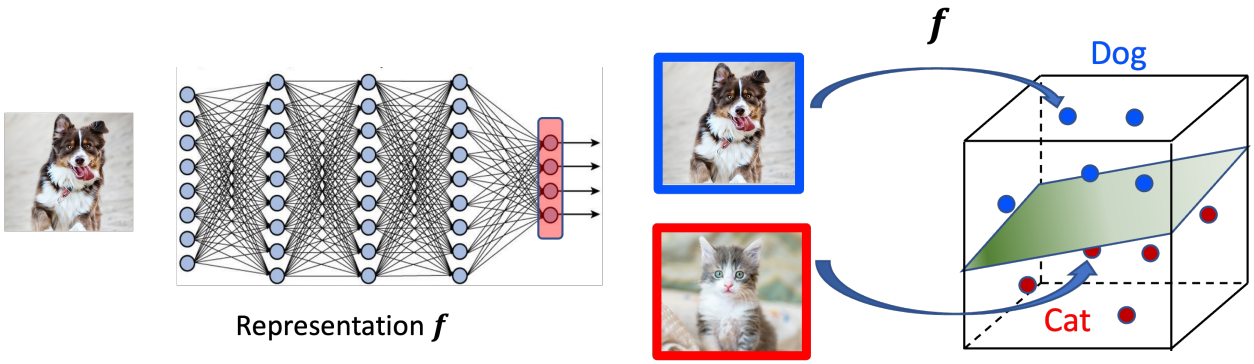
Point (a) suggests that certain structure properties of the unlabeled implicitly provide us hints about subsequent downstream tasks, and self-supervised learning can help tease out this signal from the data. Point (b) suggests a simple and empirically effective way to use a pre-trained model that leverages the model's learned representations. In this thesis we identify and mathematically quantify distributional properties of the unlabeled data, for different SSL methods like contrastive learning, language modeling and self-prediction, that can provably lead to learning good representations. In the next section we delve deeper into idea of representation learning and our formalization of why self-supervised learning helps with downstream tasks.

1.1 Overarching theoretical formalization

We first describe the key components of our theoretical formalization. The term *pre-training* is used to denote training a model on a some task or tasks for potential use in subsequent more interesting tasks, a.k.a. *downstream tasks*. The downstream task (e.g. clasification) is typically assumed to be revealed after pre-training is performed, and is often expensive to obtain labeled data for. Self-supervised learning (SSL) is a particular kind of pre-training that only uses unlabeled data. We now describe the idea of representation learning that is central to our theoretical formalization.

1.1.1 Representation learning

A recurring theme in self-supervised learning, and deep learning in general, is the idea of *representation learning* [Bengio et al., 2013]. The goal is to learn a mapping from inputs (e.g. text, images) to vectors, such that simple operations on these vectors reveal interesting properties of the input, e.g. inner products between representations of two inputs can encode some notion of semantic similarity. Learning such representations can help a learner transfer "knowledge and skills" from a pre-training task to the downstream task. In practice, low-dimensional representations extracted from some layer(s) of a model pre-trained with self-supervised learning (see Figure 1.1a) can often solve downstream tasks by just learning linear classifiers on top of them. The *low-dimensionality* of these representations often helps solving the downstream task with very few labeled



(a) Representation function (b) Linear classification

Figure 1.1: Representations from the final (or an intermediate) layer can be extracted from a network that is pre-trained using self-supervised learning. This can be used to solve a downstream classification task by using the available labeled data to learn a linear classifier separating representations from different classes. This idea, also referred to as head tuning, leads to good performance on many downstream tasks.

samples. The key question that arises here in the context of self-supervised learning is:

Why are low-dimensional **representations**, *learned by solving a self-supervised learning task, able to solve downstream tasks with* **linear classifiers**?

We denote a representation function f as a mapping from inputs x from any domain $\mathcal{X}$ to d-dimensional vectors, i.e. $f : \mathcal{X} \to \mathbb{R}^d$. The quality of a learned representation is evaluated by its performance on a certain downstream tasks of interest. In this thesis we focus our attention on the downstream task of classification.

1.1.2 Downstream classification task

The goal of a classification task is to map an input x to one of many classes. For a given classification task $\mathcal{T}$, we evaluate a representation through the minimimum classification error achievable by a (learned) linear classifier on top of the representation f. This is roughly defined[1] as follows:

$$L_{\text{clf}}(f; \mathcal{T}) \coloneqq \min_{W} \; L_{\text{clf}}(W \circ f; \mathcal{T}) \qquad \text{(Downstream linear classification)}$$

The linear classifier is typically learned by using the little labeled data that is available for the task $\mathcal{T}$, as denoted in Figure 1.1. The goal of self-supervised learning, in the context of this thesis, is to learn a representation f using unlabeled data that can help with the above metric L_{clf}.

[1]Precise definitions and notations are deferred to individual chapters.

1.1.3 Self-supervised learning task

We formalize the goal of a self-supervised learning (SSL) task as learning a representation function f. As is ubiquitous in deep learning, we set up this task as the minimization of an objective function as follows:

$$\text{minimize}_{f \in \mathcal{F}} \; L_{\text{ssl}}(f) \quad \text{using algorithm } \mathcal{A} \qquad \text{(Self-supervised learning)}$$

where $\mathcal{F}$ is a class of representation functions accounting for model architecture (e.g. ResNet-18 [He et al., 2016]) and $\mathcal{A}$ denotes a training algorithm (e.g. stochastic gradient descent, Adam) that is used to learn f. The SSL methods that we consider can all be interpreted as some objective minimization. For instance, one way to phrase a self-prediction task of predicting one part of an input (x_2) from the rest of it (x_1) is through the following objective function:

$$L_{\text{ssl}}(f) = \mathop{\mathbb{E}}_{(x_1, x_2)} \left[\|x_2 - f(x_1)\|_2^2 \right] \tag{1.1}$$

Given any formalization of SSL, an important question to ask is, "What is a good SSL task?" We use the intuitive characterization that an SSL task is "good" if doing very well on the SSL task can guarantee doing well on a downstream task of interest. Given the above objective minimization formalization of an SSL task, we distill the non-trivial question about why SSL is beneficial as follows:

Why should approximately minimizing L_{ssl} (using $\mathcal{F}$ & $\mathcal{A}$) lead to good performance on L_{clf}?

We now discuss our theoretical guarantees in the context of the above question.

1.1.4 Assumptions and guarantees

Since the unlabeled data used in L_{ssl} has no access to downstream labels, it is a priori unclear why minimizing L_{ssl} should help with the evaluation metric of L_{clf} that uses labels. In fact answering the question in full generality may not even be possible. For instance, there is no apparent reason why achieving fluency in reading books in English should help with a downstream task of excelling at badminton. However it is intuitively clear that working on fitness, muscle strength and reflexes can indeed help with learning badminton quicker. Thus it is essential to establish some connection between the SSL task and the downstream task in order to have any hope of theoretically studying the benefit of SSL. As alluded to earlier, we do so by making assumptions about the unlabeled data distribution and the downstream task. Motivated by this idea, we

present an informal version of the results that are shown for various SSL methods.

Informal Theorem 1.1.1. *Under appropriate assumptions on the unlabeled data distribution and the downstream task, we show that an SSL task is useful through one or both the following results:*

(a) Any approximate minimizer of L_{ssl} is good for L_{clf}, i.e.

$$L_{ssl}(f) \leq \inf_{f^\star} L_{ssl}(f^\star) + \epsilon \implies L_{clf}(f) \leq g(\epsilon), \quad \forall f \tag{1.2}$$

(b) Any approximate minimizer of L_{ssl}, within the class $\mathcal{F}$, is good for L_{clf}, i.e.

$$L_{ssl}(f) \leq \inf_{f^\star \in \mathcal{F}} L_{ssl}(f^\star) + \epsilon \implies L_{clf}(f) \leq g_{\mathcal{F}}(\epsilon), \quad \forall f \in \mathcal{F} \tag{1.3}$$

where g and $g_{\mathcal{F}}$ are non-decreasing functions with a small value at $\epsilon = 0$.

The above result suggests that under an appropriate connection between the unlabeled data distribution and the downstream task (similar to the badminton example from above), doing well on the SSL objective L_{ssl} is indeed indicative of doing well on the downstream evaluation L_{clf}. The function $g(\epsilon)$ (or $g_{\mathcal{F}}(\epsilon)$) is monotonic and captures the effect of sub-optimality in L_{ssl} to L_{clf}. Quantifying such robustness is important because in practice we can never hope to learn an exact minimizer of L_{ssl}, due to use of finite amount of labeled data and optimization imperfections. Result (b) differs from (a) in that it considers representation function class into account as well, rather than treating f as a black-box, and this can sometimes be crucial to show non-vacuous guarantees.

The above formalization forms the basis of our study of various SSL methods. The goal, for various SSL methods, is to identify key properties of the unlabeled data distribution and the objective function L_{ssl} that can provably lead to SSL being useful for a downstream task. We note that the precise assumptions required to mathematically study this question may not be perfectly satisfied in practice. However the hope is that a deeper mathematical understanding can help provide a new set of insights to develop better self-supervised learning methods. A quote, attributed to the statistician George Box [Box, 1976], that best captures this philosophy is, "all models are wrong, but some are useful". With this backdrop, we provide an overview of the various parts and chapters in the rest of the thesis.

1.2 Overview of contributions

The thesis is divided into three parts, corresponding to different self-supervised learning methods that we study: contrastive learning, self-prediction and language modeling.

1.2.1 Contrastive learning

In Part I we study the method of contrastive learning, where idea is to learn representations using unlabeled data that can encode some notion of semantic similarity through operations like inner products. The data for semantic similarity is directly obtained from unlabeled data, without access to downstream labels. More precisely, the goal is to learn representation f that can contrast *similar pairs* of points (x, x^+) from random pairs of points (x, x^-). A simple objective function that achieves this goal is the following:

$$L_{\text{ssl}}(f) = \mathop{\mathbb{E}}_{(x,x^+),x^-} \left[\log \left(1 + e^{-f(x)^\top f(x^+) + f(x)^\top f(x^-)} \right) \right] \tag{1.4}$$

where $f(x)^\top f(x^+)$ is encouraged to be higher than $f(x)^\top f(x^-)$. Such representations turn out to be very effective at solving classification tasks with linear classifiers, with utility in many domains like text, images, speech, graph data, etc.

This part contains two chapters, motivated by different strategies to obtain similar pairs (x, x^+) from unlabeled data. The two different settings highlight the role of two important factors for the success of contrastive learning: unlabeled data distribution and inductive biases of the function class $\mathcal{F}$.

Role of data distributions. One idea to generate similar pairs is to leverage temporal similarity, e.g. consecutive words/sentences in a text corpus or nearby frames in a video can be treated as similar. This motivates our study of contrastive learning in Chapter 2, based on our published paper [Arora et al., 2019]. In this chapter, we provide a formalization for the notion of semantic similarity that implicitly connects it to the downstream classes of interest. More precisely, we do it through a conditional independence assumption: similar pairs from unlabeled data are assumed to be conditionally independent on the latent classes of interest. Under this assumption, we show a guarantee similar to Equation (1.2) where for any representation f, $L_{\text{clf}}(f) = \mathcal{O}(L_{\text{ssl}}(f))$. This result provides the first formalization, to the best of our knowledge, of when and why an SSL method like contrastive learning can help with downstream tasks.

Role of inductive biases of function class. Chapter 3 delves into contrastive learning with data augmentations, where a similar pair of data points is obtained by sampling two transformations (or augmentations) of the same input. For this setting, we discuss why existing theoretical analyses fail to capture the full power of contrastive learning. In particular, we argue that the earlier assumptions required on the augmentations are neither necessary nor realistic for many data augmentations used in practice. We identify a *disjoint augmentation setting* where any analysis that has the form of Equation (1.2) (which subsumes existing analyses) will provably fail, despite the practical success of contrastive learning in many such settings. Instead we argue that it is important to incorporate the inductive biases of the function class into the analysis, akin to Equation (1.3) to get non-vacuous guarantees. This chapter is based on [Saunshi et al., 2022].

1.2.2 Self-prediction methods

In Chapter 4 we study a class of methods that are broadly termed as self-prediction based methods, based on the paper [Lee et al., 2021]. The idea is to use unlabeled data to set up a task that requires a model to predict part of an input from the rest of it. Examples of this include predicting missing words in a sentence from the rest of it, or predicting a missing patch in an image from the rest of it. Equation (1.1) provides a simple formalization of this idea as an objective function. In our analysis, we show that an approximate conditional independence assumption on the observed and unobserved components can lead to representations that can help with good downstream linear classification performance. We show guarantees like Equation (1.2) that treat the representation as a black-box, and also show guarantees like Equation (1.3) for a linear representation function class. This work not only helps us relax the conditional independence assumption from earlier, but also allows us to study a broader class of SSL methods.

1.2.3 Language modeling

Chapter 5 explores the idea of auto-regressive language modeling and its impressive successes on solving downstream NLP tasks. The goal of language modeling is to model the distribution of natural language by solving the task of next word prediction, i.e. given a context (partial sentence), predict a distribution over the possible next words that form meaningful completions for the context. This simple idea of next word prediction leads to models with very impressive performances on many downstream NLP tasks. The intuitive reasoning for this success is that next word prediction is a difficult task and requires a good degree of language understanding, which can potentially help with solving downstream tasks. We mathematically formalize this intuition for the downstream task of sentence classification. More precisely, we argue and verify

experimentally, that many classification tasks of interest can be rephrased as sentence completion problems through the use of *prompts*. We mathematically formalize such tasks as *natural classification tasks* and show how an near optimal language model can lead to representations with good downstream performance, showing guarantees like Equation (1.2) for arbitary language models and guarantees like Equation (1.3) for softmax-based language models. This work is based on the paper [Saunshi et al., 2021]

1.3 Previously published work

Chapter 2 contains joint work with Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak and Orestis Plevrakis, which was previously published in ICML 2019 [Arora et al., 2019].

Chapter 3 contains joint work with Sanjeev Arora, Jordan T. Ash, Surbhi Goel, Sham Kakade, Akshay Krishnamurthy, Dipendra Misra and Cyril Zhang, which was previously published in ICML 2022 Saunshi et al. [2022].

Chapter 4 contains joint work with Jason Lee, Qi Lei and Jiacheng Zhou, which was previously published in NeurIPS 2021 Lee et al. [2021].

Chapter 5 contains joint work with Sanjeev Arora and Sadhika Malladi, which was previously published in ICLR 2021 Saunshi et al. [2021].

Part I

Contrastive Learning

Chapter 2

A Theoretical Analysis of Contrastive Unsupervised Representation Learning

In this chapter, we study the class of self-supervised learning methods that are reminiscent of the well-known word2vec embedding algorithm: leveraging availability of pairs of semantically "similar" data points and "negative samples," the learner forces the inner product of representations of similar pairs with each other to be higher on average than with negative samples. We use the term *contrastive learning* for such algorithms and present a theoretical framework for analyzing them by introducing *latent classes* and hypothesizing that semantically similar points are sampled from the same latent class. This framework allows us to show provable guarantees on the performance of the learned representations on the average classification task that is comprised of a subset of the same set of latent classes. Our generalization bound also shows that learned representations can reduce (labeled) sample complexity on downstream tasks. We conduct controlled experiments in both the text and image domains to support the theory. This chapter is based on previously published work [Arora et al., 2019].

2.1 Introduction

Good quality representations for data are ubiquitous in machine learning. In natural language processing (NLP), low-dimensional representations of text – called *text embeddings* – have been computed with unlabeled data Peters et al. [2018], Devlin et al. [2019]. Often the embedding function is trained by using the embedding of a piece of text to predict the surrounding text Kiros et al. [2015], Logeswaran and Lee [2018], Pagliardini et al. [2018]. Similar methods that leverage similarity in nearby frames in a video clip have had some success for images as well Wang and Gupta [2015].

Many of these algorithms are related: they assume access to pairs or tuples (in the form of co-occurrences) of text/images that are more *semantically similar* than randomly sampled text/images, and their objective forces representations to respect this similarity on average. For instance, in order to learn a representation function f for sentences, a simplified version of what Logeswaran and Lee [2018] minimize is the following loss function

$$\mathop{\mathbb{E}}_{x,x^+,x^-}\left[-\log\left(\frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)}+e^{f(x)^T f(x^-)}}\right)\right]$$

where (x, x^+) are a similar pair and x^- is presumably dissimilar to x (often chosen to be a random point) and typically referred to as a *negative sample*. Though reminiscent of past ideas – e.g. kernel learning, metric learning, co-training Cortes et al. [2010], Bellet et al. [2013], Blum and Mitchell [1998] – these algorithms lack a theoretical framework *quantifying* when and why they work. While it seems intuitive that minimizing such loss functions should lead to representations that capture 'similarity,' formally it is unclear why the learned representations should do well on downstream *linear classification tasks* – their somewhat mysterious success is often treated as an obvious consequence. To analyze this success, a framework must connect 'similarity' in unlabeled data with the semantic information that is implicitly present in downstream tasks.

We propose the term *Contrastive Learning* for such methods and provide a new conceptual framework with minimal assumptions[1]. Our main contributions are the following:

1. We formalize the notion of semantic similarity by introducing *latent classes*. Similar pairs are assumed to be drawn from the same latent class. A downstream task is comprised of a subset of these latent classes.

2. Under this formalization, we prove that a representation function f learned from a function class

[1]The alternative would be to make assumptions about generative models of data. This is difficult for images and text.

$\mathcal{F}$ by contrastive learning has low *average* linear classification loss if $\mathcal{F}$ contains a function with low unsupervised loss. Additionally, we show a generalization bound for contrastive learning that depends on the Rademacher complexity of $\mathcal{F}$. After highlighting inherent limitations of negative sampling, we show sufficient properties of $\mathcal{F}$ which allow us to overcome these limitations.

3. Using insights from the above framework, we provide a novel extension of the algorithm that can leverage larger blocks of similar points than pairs, has better theoretical guarantees, and performs better in practice.

Ideally, one would like to show that contrastive learning always gives representations that *compete* with those learned from the same function class with plentiful labeled data. Our formal framework allows a rigorous study of such questions: we show a simple counterexample that prevents such a blanket statement without further assumptions. However, if the representations are well-concentrated and the mean classifier (Definition 2.2.1) has good performance, we can show a weaker version of the ideal result (Corollary 2.5.2). Sections 2.2 and 2.3 give an overview of the framework and the results, and subsequent sections deal with the analysis. Related work is discussed in Section 2.7 and Section 2.8 describes experimental verification and support for our framework.

2.2 Framework for contrastive learning

We first set up notation and describe the framework for unlabeled data and classification tasks that will be essential for our analysis. Let $\mathcal{X}$ denote the set of all possible data points. Contrastive learning assumes access to *similar* data in the form of pairs (x, x^+) that come from a distribution $\mathcal{D}_{sim}$ as well as k i.i.d. *negative samples* $x_1^-, x_2^-, \ldots, x_k^-$ from a distribution $\mathcal{D}_{neg}$ that are presumably unrelated to x. Learning is done over $\mathcal{F}$, a class of *representation functions* $f : \mathcal{X} \to \mathbb{R}^d$, such that $\|f(\cdot)\| \leq R$ for some $R > 0$, where we use $\|\cdot\|$ to denote the ℓ_2 norm $\|\cdot\|_2$ unless specified otherwise.

Latent classes

To formalize the notion of semantically similar pairs (x, x^+), we introduce the concept of *latent classes*.

Let $\mathcal{C}$ denote the set of all latent classes. Associated with each class $c \in \mathcal{C}$ is a probability distribution $\mathcal{D}_c$ over $\mathcal{X}$.

Roughly, $\mathcal{D}_c(x)$ captures how relevant x is to class c. For example, $\mathcal{X}$ could be natural images and c the class "dog" whose associated $\mathcal{D}_c$ assigns high probability to images containing dogs and low/zero probabilities to other images. Classes can overlap arbitrarily.[2] Finally, we assume a distribution ρ over the classes that characterizes how these classes naturally occur in the unlabeled data. Note that we make no assumption about the functional form of $\mathcal{D}_c$ or ρ.

Semantic similarity

To formalize similarity, we assume similar data points x, x^+ are i.i.d. draws from the same class distribution $\mathcal{D}_c$ for some class c picked randomly according to measure ρ. Negative samples are drawn from the marginal of $\mathcal{D}_{sim}$:

$$\mathcal{D}_{sim}(x, x^+) = \mathop{\mathbb{E}}_{c\sim\rho} \mathcal{D}_c(x)\mathcal{D}_c(x^+) \tag{2.1}$$

$$\mathcal{D}_{neg}(x^-) = \mathop{\mathbb{E}}_{c\sim\rho} \mathcal{D}_c(x^-) \tag{2.2}$$

Since classes are allowed to overlap and/or be fine-grained, this is a plausible formalization of "similarity." As the identity of the class in not revealed, we call it unlabeled data. Currently empirical works heuristically identify such similar pairs from co-occurring image or text data.

Supervised tasks

We now characterize the tasks that a representation function f will be tested on. A $(k+1)$-way[3] supervised task $\mathcal{T}$ consists of distinct classes $\{c_1, \ldots, c_{k+1}\} \subseteq \mathcal{C}$. The labeled dataset for the task $\mathcal{T}$ consists of m i.i.d. draws from the following process:

A label $c \in \{c_1, ..., c_{k+1}\}$ is picked according to a distribution $\mathcal{D}_{\mathcal{T}}$. Then, a sample x is drawn from $\mathcal{D}_c$. Together they form a labeled pair (x, c) with distribution

$$\mathcal{D}_{\mathcal{T}}(x, c) = \mathcal{D}_c(x)\mathcal{D}_{\mathcal{T}}(c) \tag{2.3}$$

A key subtlety in this formulation is that the classes in downstream tasks and their associated data distributions $\mathcal{D}_c$ are the same as in the unlabeled data. This provides a path to formalizing how capturing similarity in

[2] An image of a dog by a tree can appear in both $\mathcal{D}_{dog}$ & $\mathcal{D}_{tree}$.
[3] We use k as the number of negative samples later.

unlabeled data can lead to quantitative guarantees on downstream tasks. $\mathcal{D}_{\mathcal{T}}$ is assumed to be uniform[4] for theorems in the main paper.

Evaluation metric for representations

The quality of the representation function f is evaluated by its performance on a multi-class classification task $\mathcal{T}$ using *linear classification*. For this subsection, we fix a task $\mathcal{T} = \{c_1, ..., c_{k+1}\}$. A multi-class classifier for $\mathcal{T}$ is a function $g : \mathcal{X} \to \mathbb{R}^{k+1}$ whose output coordinates are indexed by the classes c in task $\mathcal{T}$.

The loss incurred by g on point $(x, y) \in \mathcal{X} \times \mathcal{T}$ is defined as $\ell(\{g(x)_y - g(x)_{y'}\}_{y' \neq y})$, which is a function of a k-dimensional vector of differences in the coordinates. The two losses we will consider in this work are the standard hinge loss $\ell(\boldsymbol{v}) = \max\{0, 1 + \max_i\{-\boldsymbol{v}_i\}\}$ and the logistic loss $\ell(\boldsymbol{v}) = \log_2\left(1 + \sum_i \exp(-\boldsymbol{v}_i)\right)$ for $\boldsymbol{v} \in \mathbb{R}^k$. Then the supervised loss of the classifier g is

$$L_{sup}(\mathcal{T}, g) \coloneqq \mathop{\mathbb{E}}_{(x,c)\sim\mathcal{D}_{\mathcal{T}}} \left[\ell\big(\{g(x)_c - g(x)_{c'}\}_{c' \neq c}\big)\right]$$

To use a representation function f with a linear classifier, a matrix $W \in \mathbb{R}^{(k+1)\times d}$ is trained and $g(x) = Wf(x)$ is used to evaluate classification loss on tasks. Since the best W can be found by fixing f and training a linear classifier, we abuse notation and define the *supervised loss* of f on $\mathcal{T}$ to be the loss when the best W is chosen for f:

$$L_{sup}(\mathcal{T}, f) = \inf_{W \in \mathbb{R}^{(k+1)\times d}} L_{sup}(\mathcal{T}, Wf) \tag{2.4}$$

Crucial to our results and experiments will be a specific W where the rows are the means of the representations of each class which we define below.

Definition 2.2.1 (Mean Classifier)**.** *For a function f and task $\mathcal{T} = (c_1, \ldots, c_{k+1})$, the mean classifier is W^μ whose c^{th} row is the mean μ_c of representations of inputs with label c: $\mu_c \coloneqq \mathop{\mathbb{E}}_{x\sim\mathcal{D}_c}[f(x)]$. We use $L^\mu_{sup}(\mathcal{T}, f) \coloneqq L_{sup}(\mathcal{T}, W^\mu f)$ as shorthand for its loss.*

Since contrastive learning has access to data with latent class distribution ρ, it is natural to have better guarantees for tasks involving classes that have higher probability in ρ.

[4]We state and prove the general case in the Appendix.

Definition 2.2.2 (Average Supervised Loss)**.** *Average loss for a function f on $(k+1)$-way tasks is defined as*

$$L_{sup}(f) \coloneqq \mathop{\mathbb{E}}_{\{c_i\}_{i=1}^{k+1} \sim \rho^{k+1}} \left[L_{sup}(\{c_i\}_{i=1}^{k+1}, f) \mid c_i \neq c_j \right]$$

The average supervised loss of its mean classifier *is*

$$L_{sup}^{\mu}(f) \coloneqq \mathop{\mathbb{E}}_{\{c_i\}_{i=1}^{k+1} \sim \rho^{k+1}} \left[L_{sup}^{\mu}(\{c_i\}_{i=1}^{k+1}, f) \mid c_i \neq c_j \right]$$

Contrastive learning algorithm

We describe the training objective for contrastive learning: the choice of loss function is dictated by the ℓ used in the supervised evaluation, and k denotes number of negative samples used for training. Let $(x, x^+) \sim \mathcal{D}_{sim}$, $(x_1^-, .., x_k^-) \sim \mathcal{D}_{neg}^k$ as defined in Equations (2.1) and (2.2).

Definition 2.2.3 (Unsupervised Loss)**.** *The population loss is*

$$L_{un}(f) \coloneqq \mathbb{E}\left[\ell\left(\left\{ f(x)^T \left(f(x^+) - f(x_i^-) \right) \right\}_{i=1}^{k} \right) \right] \tag{2.5}$$

and its empirical counterpart with M samples $(x_j, x_j^+, x_{j1}^-, ..., x_{jk}^-)_{j=1}^M$ from $\mathcal{D}_{sim} \times \mathcal{D}_{neg}^k$ is

$$\widehat{L}_{un}(f) = \frac{1}{M} \sum_{j=1}^{M} \ell\left(\left\{ f(x_j)^T \left(f(x_j^+) - f(x_{ji}^-) \right) \right\}_{i=1}^{k} \right) \tag{2.6}$$

Note that, by the assumptions of the framework described above, we can now express the unsupervised loss as

$$L_{un}(f) \quad = \mathop{\mathbb{E}}_{\substack{c^+, c_i^- \\ \sim \rho^{k+1}}} \mathop{\mathbb{E}}_{\substack{x, x^+ \sim \mathcal{D}_{c^+}^2 \\ x_i^- \sim \mathcal{D}_{c_i^-}}} \left[\ell\left(\left\{ f(x)^T \left(f(x^+) - f(x_i^-) \right) \right\} \right) \right]$$

The algorithm to learn a representation function from $\mathcal{F}$ is to find a function $\widehat{f} \in \arg\min_{f \in \mathcal{F}} \widehat{L}_{un}(f)$ that minimizes the empirical unsupervised loss. This function $\widehat{f}$ can be subsequently used for supervised linear classification tasks. In the following section we proceed to give an overview of our results that stem from this framework.

2.3 Overview of analysis and results

What can one *provably* say about the performance of $\widehat{f}$? As a first step we show that L_{un} is like a "surrogate" for L_{sup} by showing that $L_{sup}(f) \leq \alpha L_{un}(f), \forall f \in \mathcal{F}$, suggesting that minimizing L_{un} makes sense. This lets us show a bound on the supervised performance $L_{sup}(\widehat{f})$ of the representation learned by the algorithm. For instance, when training with one negative sample, the performance on average binary classification has the following guarantee:

Theorem 2.4.1 *(Informal binary version).*

$$L_{sup}(\widehat{f}) \leq \alpha L_{un}(f) + \eta \; Gen_M + \delta \qquad \forall f \in \mathcal{F}$$

where α, η, δ are constants depending on the distribution ρ and $Gen_M \to 0$ as $M \to \infty$. When ρ is uniform and $|\mathcal{C}| \to \infty$, we have that $\alpha, \eta \to 1, \; \delta \to 0$.

At first glance, this bound seems to offer a somewhat complete picture: *When the number of classes is large, if the unsupervised loss can be made small by $\mathcal{F}$, then the supervised loss of $\widehat{f}$, learned using finite samples, is small.*

While encouraging, this result still leaves open the question: Can $L_{un}(f)$ indeed be made small on reasonable datasets using function classes $\mathcal{F}$ of interest, even though the similar pair and negative sample can come from the same latent class? We shed light on this by upper-bounding $L_{un}(f)$ by two components: (a) the loss $L_{un}^{\neq}(f)$ for the case where the positive and negative samples are from different classes; (b) a notion of deviation $s(f)$, within each class.

Theorem 2.4.6 *(Informal binary version).*

$$L_{sup}(\widehat{f}) \leq L_{un}^{\neq}(f) + \beta s(f) + \eta \; Gen_M \qquad \forall f \in \mathcal{F}$$

for constants β, η that depend on the distribution ρ. Again, when ρ is uniform and $|\mathcal{C}| \to \infty$ we have $\beta \to 0, \eta \to 1$.

This bound lets us infer the following: *if the class $\mathcal{F}$ is rich enough to contain a function f for which $L_{un}^{\neq}(f) + \beta s(f)$ is low, then $\widehat{f}$ has high supervised performance.* Both $L_{un}^{\neq}(f)$ and $s(f)$ can potentially be made small for rich enough $\mathcal{F}$.

Ideally, however, one would want to show that $\widehat{f}$ can compete on classification tasks with every $f \in \mathcal{F}$

$$(\textit{Ideal Result}): \quad L_{sup}(\widehat{f}) \leq \alpha L_{sup}(f) + \eta \; Gen_M \tag{2.7}$$

Unfortunately, we show in Section 2.5.1 that the algorithm can pick something far from the optimal f. However, we extend Theorem 2.4.6 to a bound similar to Equation (2.7) (where the classification is done using the mean classifier) under assumptions about the intraclass concentration of f and about its mean classifier having high margin.

Sections 2.6.1 and 2.6.2 extend our results to the more complicated setting where the algorithm uses k negative samples Equation (2.5) and note an interesting behavior: increasing the number of negative samples beyond a threshold can hurt the performance. In Section 2.6.3 we show a novel extension of the algorithm that utilizes larger blocks of similar points. Finally, we perform controlled experiments in Section 2.8 to validate components of our framework and corroborate our suspicion that the mean classifier of representations learned using labeled data has good classification performance.

2.4 Guaranteed average binary classification

To provide the main insights, we prove the algorithm's guarantee when we use only 1 negative sample ($k = 1$). For this section, let $L_{sup}(f)$ and $L^{\mu}_{sup}(f)$ be as in Definition 2.2.2 for binary tasks. We will refer to the two classes in the supervised task as well as the unsupervised loss as c^+, c^-. Let $\mathcal{S} = \{x_j, x_j^+, x_j^-\}_{j=1}^M$ be our training set sampled from the distribution $\mathcal{D}_{sim} \times \mathcal{D}_{neg}$ and $\widehat{f} \in \arg\min_{f \in \mathcal{F}} \widehat{L}_{un}(f)$.

2.4.1 Upper bound using unsupervised loss

Let $f_{|\mathcal{S}} = \big(f_t(x_j), f_t(x_j^+), f_t(x_j^-)\big)_{j \in [M], t \in [d]} \in \mathbb{R}^{3dM}$ be the restriction on $\mathcal{S}$ for any $f \in \mathcal{F}$. Then, the statistical complexity measure relevant to the estimation of the representations is the following Rademacher average

$$\mathcal{R}_{\mathcal{S}}(\mathcal{F}) = \mathop{\mathbb{E}}_{\sigma \sim \{\pm 1\}^{3dM}} \big[\sup_{f \in \mathcal{F}} \langle \sigma, f_{|\mathcal{S}} \rangle \big]$$

Let $\tau = \mathop{\mathbb{E}}_{c, c' \sim \rho^2} \mathbf{1}\{c = c'\}$ be the probability that two classes sampled independently from ρ are the same.

Theorem 2.4.1. *With probability at least $1-\delta$, for all $f \in \mathcal{F}$*

$$L^{\mu}_{sup}(\widehat{f}) \leq \frac{1}{(1-\tau)}(L_{un}(f)-\tau) + \frac{1}{(1-\tau)}Gen_M$$

where

$$Gen_M = O\left(R\frac{\mathcal{R}_{\mathcal{S}}(\mathcal{F})}{M} + R^2\sqrt{\frac{\log\frac{1}{\delta}}{M}}\right)$$

Remark 2.4.2. *The complexity measure $\mathcal{R}_S(\mathcal{F})$ is tightly related to the labeled sample complexity of the classification tasks. For the function class $\mathcal{G} = \{w^T f(\cdot) | f \in \mathcal{F},\ \|w\| \leq 1\}$ that one would use to solve a binary task from scratch using labeled data, it can be shown that $\mathcal{R}_{\mathcal{S}}(\mathcal{F}) \leq d\mathcal{R}_{\mathcal{S}}(\mathcal{G})$, where $\mathcal{R}_{\mathcal{S}}(\mathcal{G})$ is the usual Rademacher complexity of $\mathcal{G}$ on $\mathcal{S}$ (Definition 3.1 from Mohri et al. [2018]).*

We state two key lemmas needed to prove the theorem.

Lemma 2.4.3. *With probability at least $1-\delta$ over the training set $\mathcal{S}$, for all $f \in \mathcal{F}$*

$$L_{un}(\widehat{f}) \leq L_{un}(f) + Gen_M$$

We prove Lemma 2.4.3 in Section 2.10.3.

Lemma 2.4.4. *For all $f \in \mathcal{F}$*

$$L^{\mu}_{sup}(f) \leq \frac{1}{(1-\tau)}\left(L_{un}(f)-\tau\right)$$

Proof. The key idea in the proof is the use of Jensen's inequality. Unlike the unsupervised loss which uses a random point from a class as a classifier, using the mean of the class as the classifier should only make the loss lower. Let $\mu_c = \mathop{\mathbb{E}}_{x\sim\mathcal{D}_c} f(x)$ be the mean of the class c.

$$\begin{aligned}
L_{un}(f) &= \mathop{\mathbb{E}}_{\substack{(x,x^+)\sim\mathcal{D}_{sim}\\ x^-\sim\mathcal{D}_{neg}}}\left[\ell(f(x)^T(f(x^+)-f(x^-)))\right] \\
&=^{(a)} \mathop{\mathbb{E}}_{\substack{c^+,c^-\sim\rho^2\\ x\sim\mathcal{D}_{c^+}}} \mathop{\mathbb{E}}_{\substack{x^+\sim\mathcal{D}_{c^+}\\ x^-\sim\mathcal{D}_{c^-}}}\left[\ell(f(x)^T(f(x^+)-f(x^-)))\right] \\
&\geq^{(b)} \mathop{\mathbb{E}}_{c^+,c^-\sim\rho^2} \mathop{\mathbb{E}}_{x\sim\mathcal{D}_{c^+}}\left[\ell(f(x)^T(\mu_{c^+}-\mu_{c^-}))\right]
\end{aligned}$$

$$
\begin{aligned}
&=^{(c)} (1-\tau) \mathop{\mathbb{E}}_{c^+,c^-\sim\rho^2}[L^{\mu}_{sup}(\{c^+,c^-\},f)|c^+\neq c^-] + \tau \\
&=^{(d)} (1-\tau) L^{\mu}_{sup}(f) + \tau
\end{aligned}
$$

where (a) follows from the definitions in Equations (2.1) and (2.2), (b) follows from the convexity of ℓ and Jensen's inequality by taking the expectation over x^+, x^- inside the function, (c) follows by splitting the expectation into the cases $c^+ = c^-$ and $c^+ \neq c^-$, from symmetry in c^+ and c^- in sampling and since classes in tasks are uniformly distributed (general distributions are handled in Section 2.11.1). Rearranging terms completes the proof. □

Proof of Theorem 2.4.1. The result follows directly by applying Lemma 2.4.4 for $\widehat{f}$ and finishing up with Lemma 2.4.3. □

One could argue that if $\mathcal{F}$ is rich enough such that L_{un} can be made small, then Theorem 2.4.1 suffices. However, in the next section we explain that unless $\tau \ll 1$, this may not always be possible and we show one way to alleviate this.

2.4.2 Price of negative sampling: class collision

Note first that the unsupervised loss can be decomposed as

$$
L_{un}(f) = \tau L^{=}_{un}(f) + (1-\tau) L^{\neq}_{un}(f) \tag{2.8}
$$

where $L^{\neq}_{un}(f)$ is the loss suffered when the similar pair and the negative sample come from different classes.

$$
L^{\neq}_{un}(f) = \mathop{\mathbb{E}}_{\substack{c^+,c^-\sim\rho^2 \\ x,x^+\sim\mathcal{D}^2_{c^+} \\ x^-\sim\mathcal{D}_{c^-}}} \left[\ell(f(x)^T(f(x^+)-f(x^-)))|c^+\neq c^-\right]
$$

and $L^{=}_{un}(f)$ is when they come from the *same class*. Let ν be a distribution over $\mathcal{C}$ with $\nu(c) \propto \rho^2(c)$, then

$$
L^{=}_{un}(f) = \mathop{\mathbb{E}}_{\substack{c\sim\nu \\ x,x^+,x^-\sim\mathcal{D}^3_c}} \left[\ell(f(x)^T(f(x^+)-f(x^-)))\right]
$$

$$\geq \mathop{\mathbb{E}}_{c\sim\nu, x\sim\mathcal{D}_c}\left[\ell(f(x)^T(\mu_c - \mu_c))\right] = 1$$

by Jensen's inequality again, which implies $L_{un}(f) \geq \tau$. In general, without any further assumptions on f, $L_{un}(f)$ can be far from τ, rendering the bound in Theorem 2.4.1 useless. However, as we will show, the magnitude of $L^{=}_{un}(f)$ can be controlled by the intraclass deviation of f. Let $\Sigma(f, c)$ the covariance matrix of $f(x)$ when $x \sim \mathcal{D}_c$. We define a notion of intraclass deviation as follows:

$$s(f) \coloneqq \mathop{\mathbb{E}}_{c\sim\nu}\left[\sqrt{\|\Sigma(f,c)\|_2}\mathop{\mathbb{E}}_{x\sim\mathcal{D}_c}\|f(x)\|\right] \tag{2.9}$$

Lemma 2.4.5. *For all $f \in \mathcal{F}$,*

$$L^{=}_{un}(f) - 1 \leq c' s(f)$$

where c' is a positive constant.

We prove Lemma 2.4.5 in Lemma 2.10.1. Theorem 2.4.1 combined with Equation (2.8) and Lemma 2.4.5 gives the following result.

Theorem 2.4.6. *With probability at least $1 - \delta$, $\forall f \in \mathcal{F}$*

$$L_{sup}(\widehat{f}) \leq L^{\mu}_{sup}(\widehat{f}) \leq L^{\neq}_{un}(f) + \beta\ s(f) + \eta\ Gen_M$$

where $\beta = c' \frac{\tau}{1-\tau}$, $\eta = \frac{1}{1-\tau}$ and c' is a constant.

The above bound highlights two sufficient properties of the function class for unsupervised learning to work: when the function class $\mathcal{F}$ is rich enough to contain *some* f with low $\beta s(f)$ as well as low $L^{\neq}_{un}(f)$ then $\widehat{f}$, the empirical minimizer of the unsupervised loss – learned using sufficiently large number of samples – will have good performance on supervised tasks (low $L_{sup}(\widehat{f})$).

2.5 Towards competitive guarantees

We provide intuition and counter-examples for why contrastive learning does not always pick the best supervised representation $f \in \mathcal{F}$ and show how our bound captures these. Under additional assumptions, we show a competitive bound where classification is done using the mean classifier.

2.5.1 Limitations of contrastive learning

The bound provided in Theorem 2.4.6 might not appear as the most natural guarantee for the algorithm. Ideally one would like to show a bound like the following: for all $f \in \mathcal{F}$,

$$\textit{(Ideal 1):} \quad L_{sup}(\widehat{f}) \leq \alpha L_{sup}(f) + \eta \; Gen_M \tag{2.10}$$

for constants α, η and generalization error Gen_M. This guarantees that $\widehat{f}$ is competitive against the *best* f on the average binary classification task. However, the bound we prove has the following form: for all $f \in \mathcal{F}$,

$$L^{\mu}_{sup}(\widehat{f}) \leq \alpha L^{\neq}_{un}(f) + \beta s(f) + \eta \; Gen_M$$

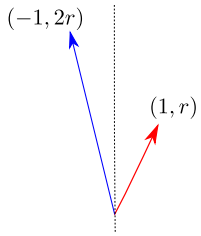
To show that this discrepancy is not an artifact of our analysis but rather stems from limitations of the algorithm, we present two examples in Figure 2.1. Our bound appropriately captures these two issues individually owing to the large values of $L^{\neq}(f)$ or $s(f)$ in each case, for the optimal f.

In Figure 2.1a, we see that there is a direction on which f_1 can be projected to perfectly separate the classes. Since the algorithm takes inner products between the representations, it inevitably considers the spurious components along the orthogonal directions. This issue manifests in our bound as the term $L^{\neq}_{un}(f_1)$ being high even when $s(f_1) = 0$. Hence, contrastive learning will not always work when the only guarantee we have is that $\mathcal{F}$ can make L_{sup} small.

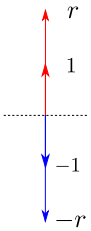
This should not be too surprising, since we show a relatively strong guarantee – a bound on L^{μ}_{sup} for the *mean classifier* of $\widehat{f}$. This suggests a natural stronger assumption that $\mathcal{F}$ can make L^{μ}_{sup} small (which is observed experimentally in Section 2.8 for function classes of interest) and raises the question of showing a bound that looks like the following: for all $f \in \mathcal{F}$,

$$\textit{(Ideal 2):} \quad L^{\mu}_{sup}(\widehat{f}) \leq \alpha L^{\mu}_{sup}(f) + \eta Gen_M \tag{2.11}$$

without accounting for any intraclass deviation – recall that $s(f)$ captures a notion of this deviation in our bound. However this is not true: high intraclass deviation may not imply high $L^{\mu}_{sup}(f)$, but can make $L^{=}_{un}(f)$ (and thus $L_{un}(f)$) high, resulting in the failure of the algorithm. Consequently, the term $s(f)$ also increases while $L^{\neq}_{un}$ does not necessarily have to. This issue, apparent in Figure 2.1b, shows that a guarantee like



(a) Mean is bad



(b) High intraclass variance

Figure 2.1: In both examples we have uniform distribution over classes $\mathcal{C} = \{c_1, c_2\}$, blue and red points are in c_1 and c_2 respectively and $\mathcal{D}_{c_i}$ is uniform over the points of c_i. In the first figure we have one point per class, while in the second we have two points per class. Let $\mathcal{F} = \{f_0, f_1\}$ where f_0 maps all points to $(0, 0)$ and f_1 is defined in the figure. In both cases, using the hinge loss, $L_{sup}(f_1) = 0$, $L_{sup}(f_0) = 1$ and in the second case $L^{\mu}_{sup}(f_1) = 0$. However, in both examples the algorithm will pick f_0 since $L_{un}(f_0) = 1$ but $L_{un}(f_1) = \Omega(r^2)$.

Equation (2.11) cannot be shown without further assumptions.

2.5.2 Competitive bound via intraclass concentration

We saw that $L^{\mu}_{sup}(f)$ being small does not imply low $L^{\mu}_{sup}(\widehat{f})$, if f is not concentrated within the classes. In this section we show that when there is an f that has intraclass concentration in a strong sense (sub-Gaussianity) and can separate classes with high margin (on average) with the mean classifier, then $L^{\mu}_{sup}(\widehat{f})$ will be low.

Let $\ell_\gamma(x) = (1 - \frac{x}{\gamma})_+$ be the hinge loss with margin γ and $L^{\mu}_{\gamma,sup}(f)$ be $L^{\mu}_{sup}(f)$ with loss function ℓ_γ.

Lemma 2.5.1. *For $f \in \mathcal{F}$, if the random variable $f(X)$, where $X \sim D_c$, is σ^2-sub-Gaussian in every direction for every class c and has maximum norm $R = max_{x \in \mathcal{X}} \|f(x)\|$, then for all $\epsilon > 0$,*

$$L^{\neq}_{un}(f) \leq \gamma L^{\mu}_{\gamma,sup}(f) + \epsilon$$

where $\gamma = 1 + c' R\sigma\sqrt{\log \frac{R}{\epsilon}}$ and c' is some constant.

The proof of Lemma 2.5.1 is provided in the Section 2.10.2. Using Lemma 2.5.1 and Theorem 2.4.6, we get the following:

Corollary 2.5.2. *For all $\epsilon > 0$, with probability at least $1 - \delta$, for all $f \in \mathcal{F}$,*

$$L^{\mu}_{sup}(\widehat{f}) \leq \gamma(f) L^{\mu}_{\gamma(f),sup}(f) + \beta s(f) + \eta Gen_M + \epsilon$$

where $\gamma(f)$ is as defined in Lemma 2.5.1, $\beta = c'\frac{\tau}{1-\tau}$, $\eta = \frac{\tau}{1-\tau}$ and c' is a constant.

2.6 Multiple negative samples and block similarity

In this section we explore two extensions to our analysis. First, in Section 2.6.1, inspired by empirical works like Logeswaran and Lee [2018] that often use more than one negative sample for every similar pair, we show provable guarantees for this case by careful handling of class collision. Additionally, in Section 2.6.2 we show simple examples where increasing negative samples beyond a certain threshold can hurt contrastive learning. Second, in Section 2.6.3, we explore a modified algorithm that leverages access to *blocks* of similar data, rather than just pairs and show that it has stronger guarantees as well as performs better in practice.

2.6.1 Guarantees for k negative samples

Here the algorithm utilizes k negative samples $x_1^-, ..., x_k^-$ drawn i.i.d. from $\mathcal{D}_{neg}$ for every positive sample pair x, x^+ drawn from $\mathcal{D}_{sim}$ and minimizes Equation (2.6). As in Section 2.4, we prove a bound for $\widehat{f}$ of the following form:

Theorem 2.6.1. *(Informal version) For all $f \in \mathcal{F}$*

$$\mathcal{L}_{sup}(\widehat{f}) \leq \mathcal{L}^{\mu}_{sup}(\widehat{f}) \leq \alpha L^{\neq}_{un}(f) + \beta s(f) + \eta\ Gen_M$$

where $L^{\neq}_{un}(f)$ and Gen_M are extensions of the corresponding terms from Section 2.4 and $s(f)$ remains unchanged. The formal statement of the theorem and its proof appears in Section 2.11.1. The key differences from Theorem 2.4.6 are β and the distribution of tasks in $\mathcal{L}_{sup}$ that we describe below. The *coefficient* β of $s(f)$ increases with k, e.g. when ρ is uniform and $k \ll |\mathcal{C}|$, $\beta \approx \frac{k}{|\mathcal{C}|}$.

The *average supervised loss* that we bound is

$$\mathcal{L}_{sup}(\widehat{f}) := \mathop{\mathbb{E}}_{\mathcal{T}\sim\mathcal{D}}\left[L_{sup}(\mathcal{T}, \widehat{f})\right]$$

where $\mathcal{D}$ is a distribution over tasks, defined as follows: sample $k+1$ classes $c^+, c_1^-, \ldots, c_k^- \sim \rho^{k+1}$, conditioned on the event that c^+ does not also appear as a negative sample. Then, set $\mathcal{T}$ to be the set of distinct classes in $\{c^+, c_1^-, \ldots, c_k^-\}$. $\mathcal{L}^{\mu}_{sup}(\widehat{f})$ is defined by using $L^{\mu}_{sup}(\mathcal{T}, \widehat{f})$.

Remark 2.6.2. *Bounding $\mathcal{L}_{sup}(\widehat{f})$ directly gives a bound for average $(k+1)$-wise classification loss $L_{sup}(\widehat{f})$ from Definition 2.2.2, since $L_{sup}(\widehat{f}) \leq \mathcal{L}_{sup}(\widehat{f})/p$, where p is the probability that the $k+1$ sampled classes are distinct. For $k \ll |\mathcal{C}|$ and $\rho \approx$ uniform, these metrics are almost equal.*

We also extend our competitive bound from Section 2.5.2 for the above $\widehat{f}$ in Section 2.11.2.

2.6.2 Effect of excessive negative sampling

The standard belief is that increasing the number of negative samples always helps, at the cost of increased computational costs. In fact for Noise Contrastive Estimation (NCE) Gutmann and Hyvärinen [2010], which is invoked to explain the success of negative sampling, increasing negative samples has shown to provably improve the asymptotic variance of the learned parameters. However, we find that such a phenomenon does not always hold for contrastive learning – larger k can hurt performance for the same inherent reasons highlighted in Section 2.5.1, as we illustrate next.

When ρ is close to uniform and the number of negative samples is $k = \Omega(|\mathcal{C}|)$, frequent class collisions can prevent the unsupervised algorithm from learning the representation $f \in \mathcal{F}$ that is optimal for the supervised problem. In this case, owing to the contribution of $s(f)$ being high, a large number of negative samples could hurt. This problem, in fact, can arise even when the number of negative samples is much smaller than the number of classes. For instance, if the best representation function $f \in \mathcal{F}$ groups classes into t "clusters",[5] such that f cannot contrast well between classes from the same cluster, then $L_{un}^{\neq}$ will contribute to the unsupervised loss being high even when $k = \Omega(t)$. We illustrate, by examples, how these issues can lead to picking suboptimal $\widehat{f}$ in Section 2.12. Experimental results in Figures 2.2a and 2.2b also suggest that larger negative samples hurt performance beyond a threshold, confirming our suspicions.

2.6.3 Blocks of similar points

Often a dataset consists of *blocks* of similar data instead of just pairs: a block consists of $x_0, x_1, \ldots x_b$ that are i.i.d. draws from a class distribution D_c for a class $c \sim \rho$. In text, for instance, paragraphs can be thought of as a *block* of sentences sampled from the same latent class. How can an algorithm leverage this additional structure?

We propose an algorithm that uses two blocks: one for positive samples $x, x_1^+, \ldots, x_b^+$ that are i.i.d. samples

[5] This can happen when $\mathcal{F}$ is not rich enough.

from $c^+ \sim \rho$ and another one of negative samples $x_1^-, \ldots x_b^-$ that are i.i.d. samples from $c^- \sim \rho$. Our proposed algorithm then minimizes the following loss:

$$L_{un}^{block}(f) := \mathbb{E}\left[\ell\left(f(x)^T\left(\frac{\sum_i f(x_i^+)}{b} - \frac{\sum_i f(x_i^-)}{b}\right)\right)\right] \tag{2.12}$$

To understand why this loss function make sense, recall that the connection between L_{sup}^{μ} and L_{un} was made in Lemma 2.4.4 by applying Jensen's inequality. Thus, the algorithm that uses the average of the positive and negative samples in blocks as a proxy for the classifier instead of just one point each should have a strictly better bound owing to the Jensen's inequality getting tighter. We formalize this intuition below. Let τ be as defined in Section 2.4.

Proposition 2.6.3. $\forall f \in \mathcal{F}$

$$L_{sup}(f) \leq \frac{1}{1-\tau}\left(L_{un}^{block}(f) - \tau\right) \leq \frac{1}{1-\tau}\left(L_{un}(f) - \tau\right)$$

This bound tells us that L_{un}^{block} is a better surrogate for L_{sup}, making it a more attractive choice than L_{un} when larger blocks are available.[6]. The algorithm can be extended, analogously to Equation (2.5), to handle more than one negative block. Experimentally we find that minimizing L_{un}^{block} instead of L_{un} can lead to better performance and our results are summarized in Section 2.8.2. We defer the proof of Proposition 2.6.3 to Section 2.10.4.

2.7 Related work

The contrastive learning framework is inspired by several empirical works, some of which were mentioned in the introduction. The use of co-occurring words as semantically similar points and negative sampling for learning word embeddings was introduced in Mikolov et al. [2013b]. Subsequently, similar ideas have been used by Logeswaran and Lee [2018] and Pagliardini et al. [2018] for sentences representations and by Wang and Gupta [2015] for images. Notably the sentence representations learned by the *quick thoughts (QT)* method in Logeswaran and Lee [2018] that we analyze has state-of-the-art results on many text classification tasks. Previous attempts have been made to explain negative sampling Dyer [2014] using the idea of Noise Contrastive Estimation (NCE) Gutmann and Hyvärinen [2010] which relies on the assumption that the

[6]Rigorous comparison of the generalization errors is left for future work.

data distribution belongs to some known parametric family. This assumption enables them to consider a broader class of distributions for negative sampling. The mean classifier that appears in our guarantees is of significance in meta-learning and is a core component of ProtoNets Snell et al. [2017].

Our data model for similarity is reminiscent of the one in *co-training* Blum and Mitchell [1998]. They assume access to pairs of "views" with the same label that are conditionally independent given the label. Our unlabeled data model can be seen as a special case of theirs, where the two views have the same conditional distributions. However, they additionally assume access to some labeled data (semi-supervised), while we learn representations using only unlabeled data, which can be subsequently used for classification when labeled data is presented. *Two-stage kernel learning* Cortes et al. [2010], Kumar et al. [2012] is similar in this sense: in the first stage, a positive linear combination of some base kernels is learned and is then used for classification in the second stage; they assume access to labels in both stages. *Similarity/metric learning* Bellet et al. [2012, 2013] learns a linear feature map that gives low distance to similar points and high to dissimilar. While they identify dissimilar pairs using labels, due to lack of labels we resort to negative sampling and pay the price of class collision. While these works analyze linear function classes, we can handle arbitrarily powerful representations. Learning of representations that are broadly useful on a distribution of tasks is done in *multitask learning*, specifically in the *learning-to-learn model* Maurer et al. [2016] but using labeled data.

Recently Hazan and Ma [2016] proposed "assumption-free" methods for representation learning via MDL/compression arguments, but do not obtain any guarantees comparable to ours on downstream classification tasks. As noted by Arora and Risteski [2017], this compression approach has to preserve *all* input information (e.g. preserve every pixel of the image) which seems suboptimal.

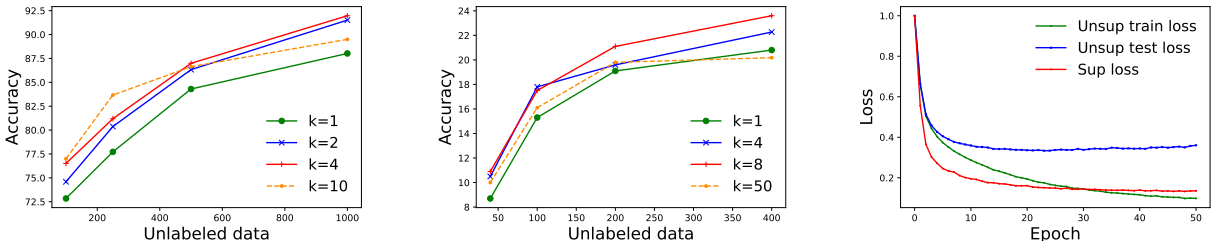
2.8 Experimental results

We report experiments in text and vision domains supporting our theory. Since contrastive learning has already shown to obtain state-of-the-art results on text classification by *quick thoughts* (QT) in Logeswaran and Lee [2018], most of our experiments are conducted to corroborate our theoretical analysis. We also show that our extension to similarity blocks in Section 2.6.3 can improve QT on a real-world task.

Datasets: Two datasets were used in the controlled experiments. (1) The CIFAR-100 dataset Krizhevsky [2009] consisting of 32x32 images categorized into 100 classes with a 50000/10000 train/test split. (2) Lacking

Table 2.1: Performance of supervised and unsupervised representations on average k-wise classification tasks (AVG-k) and for comparison, on full multiclass (TOP-R) which is not covered by our theory. Classifier can have a trained output layer (TR), or the mean classifier (μ) of Definition 2.2.1, with μ-5 indicating the mean was computed using only 5 labeled examples.

		SUPERVISED			UNSUPERVISED		
		TR	μ	μ-5	TR	μ	μ-5
WIKI-3029	AVG-2	97.8	97.7	97.0	97.3	97.7	96.9
	AVG-10	89.1	87.2	83.1	88.4	87.4	83.5
	TOP-10	67.4	59.0	48.2	64.7	59.0	45.8
	TOP-1	43.2	33.2	21.7	38.7	30.4	17.0
CIFAR-100	AVG-2	97.2	95.9	95.8	93.2	92.0	90.6
	AVG-5	92.7	89.8	89.4	80.9	79.4	75.7
	TOP-5	88.9	83.5	82.5	70.4	65.6	59.0
	TOP-1	72.1	69.9	67.3	36.9	31.8	25.0



(a) CIFAR-100 (b) Wiki-3029 (c) Wiki-3029

Figure 2.2: Effect of amount of unlabeled data and # of negative samples on unsupervised representations, measured on binary classification for CIFAR100 in (a) and on top-1 performance on Wiki-3029 in Fig (b) (top-1 performance is used because avg binary was same for all k). Fig. (c) shows the dynamics of train/test loss; supervised loss roughly tracks unsupervised test loss, as suggested by Theorem 2.4.1

an appropriate NLP dataset with large number of classes, we create the Wiki-3029 dataset, consisting of 3029 Wikipedia articles as the classes and 200 sentences from each article as samples. The train/dev/test split is 70%/10%/20%. To test our method on a more standard task, we also use the unsupervised part of the IMDb review corpus Maas et al. [2011], which consists of 560K sentences from 50K movie reviews. Representations trained using this corpus are evaluated on the supervised IMDb binary classification task, consisting of training and testing set with 25K reviews each.

2.8.1 Controlled experiments

To simulate the data generation process described in Section 2.2, we generate similar pairs (blocks) of data points by sampling from the same class. Dissimilar pairs (negative samples) are selected randomly. Contrastive learning was done using our objectives Equation (2.5), and compared to performance of standard

Table 2.2: Effect of larger block size on representations. For CIFAR-100 and Wiki-3029 we measure the average binary classification accuracy. IMDb representations are tested on IMDb supervised task. CURL is our large block size contrastive method, QT is the algorithm from Logeswaran and Lee [2018]. For larger block sizes, QT uses all pairs within a block as similar pairs. We use the same GRU architecture for both CURL and QT for a fair comparison.

Dataset	Method	$b = 2$	$b = 5$	$b = 10$
CIFAR-100	CURL	88.1	89.6	89.7
Wiki-3029	CURL	96.6	97.5	97.7
IMDb	CURL	89.2	89.6	89.7
	QT	86.5	87.7	86.7

supervised training, with both using the *same architecture* for representation f. For CIFAR-100 we use VGG-16 Simonyan and Zisserman [2014] with an additional 512x100 linear layer added at the end to make the final representations 100 dimensional, while for Wiki-3029 we use a Gated Recurrent Network (GRU) Chung et al. [2015] with output dimension 300 and fix the word embedding layer with pretrained GloVe embeddings Pennington et al. [2014]. The unsupervised model for CIFAR-100 is trained with 500 blocks of size 2 with 4 negative samples, and for Wiki-3029 we use 20 blocks of size 10 with 8 negative samples. We test (1) learned representations on average tasks by using the mean classifier and compare to representations trained using labeled data; (2) the effect of various parameters like amount of unlabeled data (N)[7], number of negative samples (k) and block size (b) on representation quality; (3) whether the supervised loss tracks the unsupervised loss as suggested by Theorem 2.4.1; (4) performance of the mean classifier of the supervised model.

Results: These appear in Table 2.1. For Wiki-3029 the unsupervised performance is very close to the supervised performance in all respects, while for CIFAR-100 the *avg-k* performance is respectable, rising to good for binary classification. One surprise is that the mean classifier, central to our analysis of unsupervised learning, performs well also with representations learned by supervised training on CIFAR-100. Even the mean computed by just 5 labeled samples performs well, getting within 2% accuracy of the 500 sample mean classifier on CIFAR-100. This suggests that representations learnt by standard supervised deep learning are actually quite concentrated. We also notice that the supervised representations have fairly low unsupervised training loss (as low as 0.4), even though the optimization is minimizing a different objective.

To measure the sample complexity benefit provided by contrastive learning, we train the supervised model on

[7]If we used M similar blocks of size b and k negative blocks for each similar block, $N = Mb(k + 1)$. In practice, however, we reuse the blocks for negative sampling and lose the factor of $k + 1$.

just 10% fraction of the dataset and compare it with an unsupervised model trained on unlabeled data whose mean classifiers are computed using the same amount of labeled data. We find that the unsupervised model beats the supervised model by almost 4% on the 100-way task and by 5% on the average binary task when only 50 labeled samples are used.

Figure 2.2 highlights the positive effect of increasing number of negative samples as well as amount of data used by unsupervised algorithm. In both cases, using a lot of negative examples stops helping after a point, confirming our suspicions in Section 2.6.2. We also demonstrate how the supervised loss tracks unsupervised test loss in Figure 2.2c.

2.8.2 Effect of block size

As suggested in Section 2.6.3, a natural extension to the model would be access to blocks of similar points. We refer to our method of minimizing the loss in Equation (2.12) as *CURL* for *Contrastive Unsupervised Representation Learning* and perform experiments on CIFAR-100, Wiki-3029, and IMDb. In Table 2.2 we see that for CIFAR-100 and Wiki-3029, increasing block size yields an improvement in classification accuracy. For IMDb, as is evident in Table 2.2, using larger blocks provides a clear benefit and the method does better than QT, which has state-of-the-art performance on many tasks. A thorough evaluation of CURL and its variants on other unlabeled datasets is left for future work.

2.9 Conclusion

Contrastive learning methods have been empirically successful at learning useful feature representations. We provide a new conceptual framework for thinking about this form of learning, which also allows us to formally treat issues such as guarantees on the quality of the learned representations. The framework gives fresh insights into what guarantees are possible and impossible, and shapes the search for new assumptions to add to the framework that allow tighter guarantees. The framework currently ignores issues of efficient minimization of various loss functions, and instead studies the interrelationships of their minimizers as well as sample complexity requirements for training to generalize, while clarifying what generalization means in this setting. Our approach should be viewed as a first cut; possible extensions include allowing tree structure – more generally metric structure – among the latent classes. Connections to meta-learning and transfer learning may arise.

We use experiments primarily to illustrate and support the new framework. But one experiment on sentence embeddings already illustrates how fresh insights derived from our framework can lead to improvements upon state-of-the-art models in this active area. We hope that further progress will follow, and that our theoretical insights will begin to influence practice, including design of new heuristics to identify semantically similar/dissimilar pairs.

2.10 Deferred proofs

2.10.1 Class collision lemma

We prove a general Lemma, from which Lemma 2.4.5 can be derived directly.

Lemma 2.10.1. *Let $c \in \mathcal{C}$ and $\ell : \mathbb{R}^t \to \mathbb{R}$ be either the t-way hinge loss or t-way logistic loss, as defined in Section 2.2. Let $x, x^+, x_1^-, ..., x_t^-$ be iid draws from $\mathcal{D}_c$. For all $f \in \mathcal{F}$, let*

$$L^{=}_{un,c}(f) = \mathop{\mathbb{E}}_{x,x^+,x_i^-}\left[\ell\left(\left\{f(x)^T\left(f(x^+) - f(x_i^-)\right)\right\}_{i=1}^t\right)\right]$$

Then

$$L^{=}_{un,c}(f) - \ell(\vec{0}) \leq c' t\sqrt{\|\Sigma(f,c)\|_2}\mathop{\mathbb{E}}_{x\sim\mathcal{D}_c}[\|f(x)\|] \tag{2.13}$$

where c' is a positive constant.

Lemma 2.4.5 is a direct consequence of the above Lemma, by setting $t = 1$ (which makes $\ell(0) = 1$), taking an expectation over $c \sim \nu$ in Equation (2.13) and noting that $\mathbb{E}_{c\sim\nu}[L^{=}_{un,c}(f)] = L^{=}_{un}(f)$.

Proof of Lemma 2.10.1. Fix an $f \in \mathcal{F}$ and let $z_i = f(x)^T\left(f(x_i^-) - f(x^+)\right)$ and $z = \max_{i\in[t]} z_i$. First, we show that $L^{=}_{un,c}(f) - \ell(\vec{0}) \leq c'\mathbb{E}[|z|]$, for some constant c'. Note that $\mathbb{E}[|z|] = \mathbb{P}[z \geq 0]\mathbb{E}[z|z \geq 0] + \mathbb{P}[z \leq 0]\mathbb{E}[-z|z \leq 0] \geq \mathbb{P}[z \geq 0]\mathbb{E}[z|z \geq 0]$.

t-way hinge loss: By definition $\ell(\boldsymbol{v}) = \max\{0, 1 + \max_{i\in[t]}\{-\boldsymbol{v}_i\}\}$. Here, $L^{=}_{un,c}(f) = \mathbb{E}[(1+z)_+] \leq \mathbb{E}[\max\{1+z, 1\}] = 1 + \mathbb{P}[z \geq 0]\mathbb{E}[z|z \geq 0] \leq 1 + \mathbb{E}[|z|]$.

t-way logistic loss: By definition $\ell(\boldsymbol{v}) = \log_2(1 + \sum_{i=1}^t e^{-\boldsymbol{v}_i})$, we have $L^{=}_{un,c}(f) = \mathbb{E}[\log_2(1 + \sum_i e^{z_i})] \leq \mathbb{E}[\log_2(1 + te^z)] \leq \max\{\frac{z}{\log 2} + \log_2(1+t), \log_2(1+t)\} = \frac{\mathbb{P}[z\geq 0]\mathbb{E}[z|z\geq 0]}{\log 2} + \log_2(1+t) \leq \frac{\mathbb{E}[|z|]}{\log 2} + \log_2(1+t)$.

Finally, $\mathbb{E}[|z|] \leq \mathbb{E}[\max_{i\in[t]}|z_i|] \leq t\mathbb{E}[|z_1|]$. But,

$$\begin{aligned}\mathbb{E}[|z_1|] &= \mathbb{E}_{x,x^+,x_1^-}\left[\ \left|f(x)^T\left(f(x_1^-) - f(x^+)\right)\right|\ \right] \\ &\leq \mathbb{E}_x\left[\|f(x)\|\sqrt{\mathbb{E}_{x^+,x_1^-}\left[\left(\frac{f(x)^T}{\|f(x)\|}\left(f(x_1^-) - f(x^+)\right)\right)^2\right]}\right] \leq \sqrt{2}\ \sqrt{\|\Sigma(f,c)\|_2}\mathop{\mathbb{E}}_{x\sim\mathcal{D}_c}[\|f(x)\|]\end{aligned}$$

□

2.10.2 Proof of Lemma 2.5.1

Fix an $f \in \mathcal{F}$ and suppose that within each class c, f is σ^2-subgaussian in every direction. [8] Let $\mu_c = \mathop{\mathbb{E}}_{x\sim\mathcal{D}_c}[f(x)]$. This means that for all $c \in \mathcal{C}$ and unit vectors v, for $x \sim D_c$, we have that $v^T(f(x) - \mu_c)$ is σ^2-subgaussian. Let $\epsilon > 0$ and $\gamma = 1 + 2R\sigma\sqrt{2\log R + \log 3/\epsilon}$. [9] Consider fixed c^+, c^-, x and let $f(x)^T(f(x^-) - f(x^+)) = \mu + z$, where

$$\mu = f(x)^T(\mu_{c^-} - \mu_{c^+}) \qquad \text{and} \qquad z = f(x)^T\left(f(x^-) - \mu_{c^-}\right) - f(x)^T\left(f(x^+) - \mu_{c^+}\right)$$

For $x^+ \sim \mathcal{D}_c^+$, $x^- \sim \mathcal{D}_c^-$ independently, z is the sum of two independent $R^2\sigma^2$-subgaussians (x is fixed), so z is $2R^2\sigma^2$-subgaussian and thus $p = \Pr[z \geq \gamma - 1] \leq e^{-\frac{4R^2\sigma^2(2\log R + \log 3/\epsilon)}{4R^2\sigma^2}} = \frac{\epsilon}{3R^2}$. So, $\mathbb{E}_z[(1 + \mu + z)_+] \leq (1-p)(\gamma + \mu)_+ + p(2R^2 + 1) \leq \gamma(1 + \frac{\mu}{\gamma})_+ + \epsilon$ (where we used that $\mu + z \leq 2R^2$). By taking expectation over $c^+, c^- \sim \rho^2$, $x \sim \mathcal{D}_{c^+}$ we have

$$\begin{aligned}
L_{un}^{\neq}(f) &\leq \mathop{\mathbb{E}}_{\substack{c^+,c^-\sim\rho^2 \\ x\sim\mathcal{D}_{c^+}}}\left[\gamma\left(1 + \frac{f(x)^T(\mu_{c^-} - \mu_{c^+})}{\gamma}\right)_+ \middle| c^+ \neq c^-\right] + \epsilon \\
&= \gamma \mathop{\mathbb{E}}_{c^+,c^-\sim\rho^2}\left[\frac{1}{2}\mathop{\mathbb{E}}_{x\sim\mathcal{D}_{c^+}}\left[\left(1 + \frac{f(x)^T(\mu_{c^-} - \mu_{c^+})}{\gamma}\right)_+\right] + \frac{1}{2}\mathop{\mathbb{E}}_{x\sim\mathcal{D}_{c^-}}\left[\left(1 + \frac{f(x)^T(\mu_{c^+} - \mu_{c^-})}{\gamma}\right)_+\right] \middle| c^+ \neq c^-\right] + \epsilon \\
&= \gamma \mathop{\mathbb{E}}_{c^+,c^-\sim\rho^2}\left[L_{\gamma,sup}^{\mu}(\{c^+, c^-\}, f) \middle| c^+ \neq c^-\right] + \epsilon
\end{aligned} \tag{2.14}$$

where $L_{\gamma,sup}^{\mu}(\{c^+, c^-\}, f)$ is $L_{sup}^{\mu}(\{c^+, c^-\}, f)$ when $\ell_\gamma(x) = (1 - x/\gamma)_+$ is the loss function. Observe that in Equation (2.14) we used that $\mathcal{D}_{\mathcal{T}}$ are uniform for binary $\mathcal{T}$, which is an assumption we work with in section 4, but we remove it in section 5. The proof finishes by observing that the last line in Equation (2.14) is equal to $\gamma L_{\gamma,sup}^{\mu}(f) + \epsilon$.

□

2.10.3 Generalization bound

We first state the following general Lemma in order to bound the generalization error of the function class $\mathcal{F}$ on the unsupervised loss function $L_{un}(\cdot)$. Lemma 2.4.3 can be directly derived from it.

[8] A random variable X is called σ^2-subgaussian if $\mathbb{E}[e^{\lambda(X - \mathbb{E}[X])}] \leq e^{\lambda^2\sigma^2/2}, \forall\lambda \in \mathbb{R}$. A random vector $V \in \mathbb{R}^d$ is σ^2-subgaussian in every direction, if $\forall u \in \mathbb{R}^d, ||u|| = 1$, the random variable $\langle u, V\rangle$ is σ^2-subgaussian.

[9] We implicitly assume here that $R \geq 1$, but for $R < 1$, we just set $\gamma = 1 + 2R\sigma\sqrt{\log 3/\epsilon}$ and the same argument holds.

Lemma 2.10.2. *Let $\ell : \mathbb{R}^k \to \mathbb{R}$ be η-Lipschitz and bounded by B. Then with probability at least $1-\delta$ over the training set $\mathcal{S} = \{(x_j, x_j^+, x_{j1}^-, \ldots, x_{jk}^-)\}_{j=1}^M$, for all $f \in \mathcal{F}$*

$$L_{un}(\hat{f}) \leq L_{un}(f) + O\left(\frac{\eta R\sqrt{k}\mathcal{R}_{\mathcal{S}}(\mathcal{F})}{M} + B\sqrt{\frac{\log\frac{1}{\delta}}{M}}\right) \tag{2.15}$$

where

$$\mathcal{R}_{\mathcal{S}}(\mathcal{F}) = \mathop{\mathbb{E}}_{\sigma\sim\{\pm 1\}^{(k+2)dM}}\left[\sup_{f\in\mathcal{F}}\langle\sigma, f_{|\mathcal{S}}\rangle\right] \tag{2.16}$$

and $f_{|\mathcal{S}} = \Big(f_t(x_j), f_t(x_j^+), f_t(x_{j1}^-), \ldots, , f_t(x_{jk}^-)\Big)_{\substack{j\in[M]\\ t\in[d]}}$

Note that for $k+1$-way classification, for hinge loss we have $\eta = 1$ and $B = O(R^2)$, while for logistic loss $\eta = 1$ and $B = O(R^2 + \log k)$. Setting $k = 1$, we get Lemma 2.4.3. We now prove Lemma 2.10.2.

Proof of Lemma 2.10.2. First, we use the classical bound for the generalization error in terms of the Rademacher complexity of the function class (see Mohri et al. [2018] Theorem 3.1). For a real function class G whose functions map from a set Z to $[0,1]$ and for any $\delta > 0$, if $\mathcal{S}$ is a training set composed by M iid samples $\{z_j\}_{j=1}^M$, then with probability at least $1 - \frac{\delta}{2}$, for all $g \in G$

$$\mathbb{E}[g(z)] \leq \frac{1}{M}\sum_{j=1}^{M} g(z_i) + \frac{2\mathcal{R}_{\mathcal{S}}(G)}{M} + 3\sqrt{\frac{\log\frac{4}{\delta}}{2M}} \tag{2.17}$$

where $\mathcal{R}_{\mathcal{S}}(G)$ is the usual Rademacher complexity. We apply this bound to our case by setting $Z = \mathcal{X}^{k+2}$, $\mathcal{S}$ is our training set and the function class is

$$G = \left\{ g_f(x, x^+, x_1^-, ..., x_k^-) = \frac{1}{B}\ell\left(\{f(x)^T\big(f(x^+) - f(x_i^-)\big)\}_{i=1}^k\right) \Big| f \in \mathcal{F}\right\} \tag{2.18}$$

We will show that for some universal constant c, $\mathcal{R}_{\mathcal{S}}(G) \leq c\frac{\eta R\sqrt{k}}{B}\mathcal{R}_{\mathcal{S}}(\mathcal{F})$ or equivalently

$$\mathop{\mathbb{E}}_{\sigma\sim\{\pm 1\}^M}\left[\sup_{f\in\mathcal{F}}\left\langle\sigma, (g_f)_{|\mathcal{S}}\right\rangle\right] \leq c\frac{\eta R\sqrt{k}}{B}\mathop{\mathbb{E}}_{\sigma\sim\{\pm 1\}^{d(k+2)M}}\left[\sup_{f\in\mathcal{F}}\left\langle\sigma, f_{|\mathcal{S}}\right\rangle\right] \tag{2.19}$$

where $(g_f)_{|\mathcal{S}} = \{g_f(x_j, x_j^+, x_{j1}^-, ..., x_{jk}^-)\}_{j=1}^M$. To do that we will use the following vector-contraction inequality.

Theorem 2.10.3. *[Corollary 4 in Maurer [2016]] Let Z be any set, and $\mathcal{S} = \{z_j\}_{j=1}^M \in Z^M$. Let $\widetilde{\mathcal{F}}$ be a class of functions $\tilde{f} : Z \to \mathbb{R}^n$ and $h : \mathbb{R}^n \to \mathbb{R}$ be L-Lipschitz. For all $\tilde{f} \in \widetilde{\mathcal{F}}$, let $g_{\tilde{f}} = h \circ \tilde{f}$. Then*

$$\mathop{\mathbb{E}}_{\sigma\sim\{\pm1\}^M}\left[\sup_{\tilde{f}\in\widetilde{\mathcal{F}}}\left\langle\sigma,(g_{\tilde{f}})_{|\mathcal{S}}\right\rangle\right] \leq \sqrt{2}L \mathop{\mathbb{E}}_{\sigma\sim\{\pm1\}^{nM}}\left[\sup_{\tilde{f}\in\widetilde{\mathcal{F}}}\left\langle\sigma,\tilde{f}_{|\mathcal{S}}\right\rangle\right]$$

where $\tilde{f}_{|\mathcal{S}} = \left(\tilde{f}_t(z_j)\right)_{t\in[n],j\in[M]}$.

We apply Theorem 2.10.3 to our case by setting $Z = \mathcal{X}^{k+2}$, $n = d(k+2)$ and

$$\widetilde{\mathcal{F}} = \left\{\tilde{f}(x, x^+, x_{j1}^-, ..., x_{jk}^-) = (f(x), f(x^+), f(x_{j1}^-), ..., f(x_{jk}^-))|f \in \mathcal{F}\right\}$$

We also use $g_{\tilde{f}} = g_f$ where $\tilde{f}$ is derived from f as in the definition of $\widetilde{F}$. Observe that now Theorem 2.10.3 is exactly in the form of Equation (2.19) and we need to show that $L \leq \frac{c}{\sqrt{2}}\frac{\eta R\sqrt{k}}{B}$ for some constant c. But, for $z = (x, x^+, x_1^-, ..., x_k^-)$, we have $g_{\tilde{f}}(z) = \frac{1}{B}\ell(\phi(\tilde{f}(z)))$ where $\phi : \mathbb{R}^{(k+2)d} \to \mathbb{R}^k$ and $\phi\left((v_t, v_t^+, v_{t1}^-, ..., v_{tk}^-)_{t\in[d]}\right) = \left(\sum_t v_t(v_t^+ - v_{ti}^-)\right)_{i\in[k]}$. Thus, we may use $h = \frac{1}{B}\ell \circ \phi$ to apply Theorem 2.10.3.

Now, we see that ϕ is $\sqrt{6k}R$-Lipschitz when $\sum_t v_t^2, \sum_t (v_t^+)^2, \sum_t (v_{tj}^-)^2 \leq R^2$ by computing its Jacobian. Indeed, for all $i, j \in [k]$ and $t \in [d]$, we have $\frac{\partial\phi_i}{\partial v_t} = v_t^+ - v_{ti}^-$, $\frac{\partial\phi_i}{\partial v_t^+} = v_t$ and $\frac{\partial\phi_i}{\partial v_{tj}^-} = -v_t 1\{i = j\}$. From triangle inequaltiy, the Frobenius norm of the Jacobian J of ϕ is

$$||J||_F = \sqrt{\sum_{i,t}(v_t^+ - v_{ti}^-)^2 + 2k\sum_t v_t^2} \leq \sqrt{4kR^2 + 2kR^2} = \sqrt{6k}R$$

Now, taking into account that $||J||_2 \leq ||J||_F$, we have that ϕ is $\sqrt{6k}R$-Lipschitz on its domain and since ℓ is η-Lipschitz, we have $L \leq \sqrt{6}\frac{\eta R\sqrt{k}}{B}$.

Now, we have that with probability at least $1 - \frac{\delta}{2}$

$$L_{un}(\hat{f}) \leq \widehat{L}_{un}(\hat{f}) + O\left(\frac{\eta R\sqrt{k}\mathcal{R}_{\mathcal{S}}(\mathcal{F})}{M} + B\sqrt{\frac{\log\frac{1}{\delta}}{M}}\right) \tag{2.20}$$

Let $f^* \in \arg\min_{f\in\mathcal{F}} L_{un}(f)$. With probability at least $1 - \frac{\delta}{2}$, we have that $\widehat{L}_{un}(f^*) \leq L_{un}(f^*) + 3B\sqrt{\frac{\log\frac{2}{\delta}}{2M}}$ (Hoeffding's inequality). Combining this with Equation (2.20), the fact that $\widehat{L}_{un}(\hat{f}) \leq \widehat{L}_{un}(f^*)$ and applying

a union bound, finishes the proof. □

2.10.4 Proof of Proposition 2.6.3

By convexity of ℓ,

$$\ell\left(f(x)^T\left(\frac{\sum_i f(x_i^+)}{b}-\frac{\sum_i f(x_i^-)}{b}\right)\right)=\ell\left(\frac{1}{b}\sum_i f(x)^T\left(f(x_i^+)-f(x_i^-)\right)\right)\leq\frac{1}{b}\sum_i\ell\left(f(x)^T\left(f(x_i^+)-f(x_i^-)\right)\right)$$

Thus,

$$L_{un}^{block}(f)=\mathop{\mathbb{E}}_{\substack{x,x_i^+\\x_i^-}}\left[\ell\left(f(x)^T\left(\frac{\sum_i f(x_i^+)}{b}-\frac{\sum_i f(x_i^-)}{b}\right)\right)\right]\leq\mathop{\mathbb{E}}_{\substack{x,x_i^+\\x_i^-}}\left[\frac{1}{b}\sum_i\ell\left(f(x)^T\left(f(x_i^+)-f(x_i^-)\right)\right)\right]=L_{un}(f)$$

The proof of the lower bound is analogous to that of Lemma 2.4.4.

□

2.11 Results for k negative samples

2.11.1 Formal theorem statement and proof

We now present Theorem 2.11.1 as the formal statement of Theorem 2.6.1 and prove it. First we define some necessary quantities.

Let $(c^+, c_1^-, \ldots, c_k^-)$ be $k+1$ not necessarily distinct classes. We define $Q(c^+, c_1^-, \ldots, c_k^-)$ to be the set of distinct classes in this tuple. We also define $I^+(c_1^-, ..., c_k^-) = \{i \in [k] \mid c_i^- = c^+\}$ to be the set of indices where c^+ reappears in the negative samples. We will abuse notation and just write Q, I^+ when the tuple is clear from the context.

To define $L_{un}^{\neq}(f)$ consider the following tweak in the way the latent classes are sampled: sample $c^+, c_1^-, \ldots, c_k^- \sim \rho^{k+1}$ conditioning on $|I^+| < k$ and then remove all c_i^-, $i \in I^+$. The datapoints are then sampled as usual:

$x, x^+ \sim \mathcal{D}_{c^+}^2$ and $x_i^- \sim \mathcal{D}_{c_i^-}$, $i \in [k]$, independently.

$$L_{un}^{\neq}(f) \coloneqq \mathop{\mathbb{E}}_{\substack{c^+, c_i^- \\ x, x^+, x_i^-}} \left[\ell \left(\left\{ f(x)^T \left(f(x^+) - f(x_i^-) \right) \right\}_{i \notin I^+} \right) \middle| |I^+| < k \right]$$

which always contrasts points from different classes, since it only considers the negative samples that are not from c^+.

The generalization error is [10]

$$Gen_M = O\left(R\sqrt{k} \frac{\mathcal{R}_{\mathcal{S}}(\mathcal{F})}{M} + (R^2 + \log k) \sqrt{\frac{\log \frac{1}{\delta}}{M}} \right)$$

were $\mathcal{R}_{\mathcal{S}}(\mathcal{F})$ is the extension of the definition in Section 2.4: $\mathcal{R}_{\mathcal{S}}(\mathcal{F}) = \mathop{\mathbb{E}}_{\sigma \sim \{\pm 1\}^{(k+2)dM}} \left[\sup_{f \in \mathcal{F}} \langle \sigma, f_{|\mathcal{S}} \rangle \right]$, where $f_{|\mathcal{S}} = \left(f_t(x_j), f_t(x_j^+), f_t(x_{j1}^-), \ldots, , f_t(x_{jk}^-) \right)_{j \in [M], t \in [d]}$.

For $c^+, c_1^-, ..., c_k^- \sim \rho^{k+1}$, let $\tau_k = \mathbb{P}[I^+ \neq \emptyset]$ and $\tau' = \mathbb{P}[c^+ = c_i^-, \forall i]$. Observe that τ_1, as defined in Section 2.4, is $\mathbb{P}[c^+ = c_1^-]$. Let $p_{max}(\mathcal{T}) = \max_c \mathcal{D}_{\mathcal{T}}(c)$ and

$$\rho_{min}^+(\mathcal{T}) = \min_{c \in \mathcal{T}} \mathbb{P}_{c^+, c_i^- \sim \rho^{k+1}} \left(c^+ = c | Q = \mathcal{T}, I^+ = \emptyset \right)$$

In Theorem 2.11.1 we will upper bound the following quantity: $\mathop{\mathbb{E}}_{\mathcal{T} \sim \mathcal{D}} \left[\frac{\rho_{min}^+(\mathcal{T})}{p_{max}(\mathcal{T})} \, L_{sup}^{\mu}(\mathcal{T}, \hat{f}) \right]$ ($\mathcal{D}$ was defined in Section 2.6.1).

Theorem 2.11.1. *Let $\hat{f} \in \arg\min_{f \in \mathcal{F}} \widehat{L}_{un}(f)$. With probability at least $1 - \delta$, for all $f \in \mathcal{F}$*

$$\mathop{\mathbb{E}}_{\mathcal{T} \sim \mathcal{D}} \left[\frac{\rho_{min}^+(\mathcal{T})}{p_{max}(\mathcal{T})} \, L_{sup}^{\mu}(\mathcal{T}, \hat{f}) \right] \leq \frac{1 - \tau'}{1 - \tau_k} L_{un}^{\neq}(f) + c' k \frac{\tau_1}{1 - \tau_k} s(f) + \frac{1}{1 - \tau_k} Gen_M$$

where c' is a constant.

Note that the definition of $s(f)$ used here is defined in Section 2.4

Proof. First, we note that both hinge and logistic loss satisfy the following property: $\forall I_1, I_2$ such that

[10] The $\log k$ term can be made $O(1)$ for the hinge loss.

$I_1 \cup I_2 = [t]$ we have that

$$\ell(\{\boldsymbol{v}_i\}_{i\in I_1}) \leq \ell(\{\boldsymbol{v}_i\}_{i\in[t]}) \leq \ell(\{\boldsymbol{v}_i\}_{i\in I_1}) + \ell(\{\boldsymbol{v}_i\}_{i\in I_2}) \tag{2.21}$$

We now prove the Theorem in 3 steps. First, we leverage the convexity of ℓ to upper bound a supervised-type loss with the unsupervised loss $L_{un}(f)$ of any $f \in \mathcal{F}$. We call it supervised-type loss because it also includes degenerate tasks: $|\mathcal{T}| = 1$. Then, we decompose the supervised-type loss into an average loss over a distribution of supervised tasks, as defined in the Theorem, plus a degenerate/constant term. Finally, we upper bound the unsupervised loss $L_{un}(f)$ with two terms: $L_{un}^{\neq}(f)$ that measures how well f contrasts points from different classes and an intraclass deviation penalty, corresponding to $s(f)$.

Step 1 (convexity): When the class c is clear from context, we write $\hat{\mu}_c = \mathop{\mathbb{E}}_{x\sim c}[\hat{f}(x)]$. Recall that the sampling procedure for unsupervised data is as follows: sample $c^+, c_1^-, ..., c_k^- \sim \rho^{k+1}$ and then $x, x^+ \sim \mathcal{D}_{c^+}^2$ and $x_i^- \sim \mathcal{D}_{c_i^-}$, $i \in [k]$. So, we have

$$\begin{aligned}
L_{un}(\hat{f}) &= \mathop{\mathbb{E}}_{\substack{c^+,c_i^-\sim\rho^{k+1}\\ x,x^+\sim\mathcal{D}_{c^+}^2\\ x_i^-\sim\mathcal{D}_{c_i^-}}} \left[\ell\left(\left\{\hat{f}(x)^T\left(\hat{f}(x^+) - \hat{f}(x_i^-)\right)\right\}_{i=1}^k\right)\right] \\
&= \mathop{\mathbb{E}}_{\substack{c^+,c_i^-\sim\rho^{k+1}\\ x\sim\mathcal{D}_{c^+}}} \mathop{\mathbb{E}}_{\substack{x^+\sim\mathcal{C}_{c^+}\\ x_i^-\sim\mathcal{D}_{c_i^-}}} \left[\ell\left(\left\{\hat{f}(x)^T\left(\hat{f}(x^+) - \hat{f}(x_i^-)\right)\right\}_{i=1}^k\right)\right] \geq \mathop{\mathbb{E}}_{\substack{c^+,c_i^-\sim\rho^{k+1}\\ x\sim\mathcal{D}_{c^+}}} \left[\ell\left(\left\{\hat{f}(x)^T\left(\hat{\mu}_{c^+} - \hat{\mu}_{c_i^-}\right)\right\}_{i=1}^k\right)\right]
\end{aligned} \tag{2.22}$$

where the last inequality follows by applying the usual Jensen's inequality and the convexity of ℓ. Note that in the upper bounded quantity, the $c^+, c_1^-, ..., c_k^-$ don't have to be distinct and so the tuple does not necessarily form a task.

Step 2 (decomposing into supervised tasks) We now decompose the above quantity to handle repeated classes.

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\substack{c^+,c_i^-\sim\rho^{k+1}\\ x\sim\mathcal{D}_{c^+}}} \left[\ell\left(\left\{\hat{f}(x)^T\left(\hat{\mu}_{c^+} - \hat{\mu}_{c_i^-}\right)\right\}_{i=1}^k\right)\right] \\
&\qquad \geq (1-\tau_k) \mathop{\mathbb{E}}_{\substack{c^+,c_i^-\sim\rho^{k+1}\\ x\sim\mathcal{D}_{c^+}}} \left[\ell\left(\left\{\hat{f}(x)^T\left(\hat{\mu}_{c^+} - \hat{\mu}_{c_i^-}\right)\right\}_{i=1}^k\right) \middle| I^+ = \emptyset\right] + \tau_k \mathop{\mathbb{E}}_{c^+,c_i^-\sim\rho^{k+1}} [\ell(\underbrace{0,...,0}_{|I^+| \text{ times}})|I^+ \neq \emptyset]
\end{aligned}$$

$$\geq (1-\tau_k) \mathop{\mathbb{E}}_{\substack{c^+,c_i^-\sim\rho^{k+1}\\ x\sim\mathcal{D}_{c^+}}} \left[\ell\left(\left\{ \hat{f}(x)^T \left(\hat{\mu}_{c^+} - \hat{\mu}_c\right) \right\}_{\substack{c\in Q\\ c\neq c^+}} \right) \middle| I^+ = \emptyset \right] + \tau_k \mathop{\mathbb{E}}_{c^+,c_i^-\sim\rho^{k+1}} \left[\ell_{|I^+|}(\vec{0}) \;\middle|\; I^+ \neq \emptyset \right] \tag{2.23}$$

where $\ell_t(\vec{0}) = \ell(0,\ldots,0)$ (t times). Both inequalities follow from the LHS of Equation (2.21). Now we are closer to our goal of lower bounding an average supervised loss, since the first expectation in the RHS has a loss which is over a set of distinct classes. However, notice that this loss is for separating c^+ from $Q(c^+, c_1^-, ..., c_k^-) \setminus \{c^+\}$. We now proceed to a symmetrization of this term to alleviate this issue.

Recall that in the main paper, sampling $\mathcal{T}$ from $\mathcal{D}$ is defined as sampling the (k+1)-tuple from ρ^{k+1} conditioned on $I^+ = \emptyset$ and setting $\mathcal{T} = Q$. Based on this definition, by the tower property of expectation, we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\substack{c^+,c_i^-\sim\rho^{k+1}\\ x\sim\mathcal{D}_{c^+}}} \left[\ell\left(\left\{ \hat{f}(x)^T \left(\hat{\mu}_{c^+} - \hat{\mu}_c\right) \right\}_{\substack{c\in Q\\ c\neq c^+}} \right) \middle| I^+ = \emptyset \right] \\
&= \mathop{\mathbb{E}}_{\mathcal{T}\sim\mathcal{D}} \mathop{\mathbb{E}}_{\substack{c^+,c_i^-\sim\rho^{k+1}\\ x\sim\mathcal{D}_{c^+}}} \left[\ell\Big(\Big\{ \hat{f}(x)^T \big(\hat{\mu}_{c^+} - \hat{\mu}_c\big) \Big\}_{\substack{c\in Q\\ c\neq c^+}} \Big) \Big| Q = \mathcal{T}, I^+ = \emptyset \right] \\
&= \mathop{\mathbb{E}}_{\mathcal{T}\sim\mathcal{D}} \mathop{\mathbb{E}}_{\substack{c^+\sim\rho^+(\mathcal{T})\\ x\sim\mathcal{D}_{c^+}}} \left[\ell\Big(\Big\{ \hat{f}(x)^T \big(\hat{\mu}_{c^+} - \hat{\mu}_c\big) \Big\}_{\substack{c\in\mathcal{T}\\ c\neq c^+}} \Big) \right]
\end{aligned} \tag{2.24}$$

where $\rho^+(\mathcal{T})$ is the distribution of c^+ when $(c^+, c_1^-, ..., c_k^-)$ are sampled from ρ^{k+1} conditioned on $Q = \mathcal{T}$ and $I^+ = \emptyset$. Recall that $\rho^+_{min}(\mathcal{T})$ from the theorem's statement is exactly the minimum out of these $|\mathcal{T}|$ probabilities. Now, to lower bound the last quantity with the LHS in the theorem statement, we just need to observe that for all tasks $\mathcal{T}$

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\substack{c^+\sim\rho^+(\mathcal{T})\\ x\sim\mathcal{D}_{c^+}}} \left[\ell\Big(\Big\{ \hat{f}(x)^T \big(\hat{\mu}_{c^+} - \hat{\mu}_c\big) \Big\}_{\substack{c\in\mathcal{T}\\ c\neq c^+}} \Big) \right] \\
&\geq \frac{\rho^+_{min}(\mathcal{T})}{p_{max}(\mathcal{T})} \mathop{\mathbb{E}}_{\substack{c^+\sim\mathcal{D}_{\mathcal{T}}\\ x\sim\mathcal{D}_{c^+}}} \left[\ell\Big(\Big\{ \hat{f}(x)^T \big(\hat{\mu}_{c^+} - \hat{\mu}_c\big) \Big\}_{\substack{c\in\mathcal{T}\\ c\neq c^+}} \Big) \right] \\
&= \frac{\rho^+_{min}(\mathcal{T})}{p_{max}(\mathcal{T})} L_{sup}(\mathcal{T}, \hat{f})
\end{aligned} \tag{2.25}$$

By combining this with Equations (2.22), (2.23) and (2.25) we get

$$(1-\tau_k)\mathop{\mathbb{E}}_{\mathcal{T}\sim\mathcal{D}}\left[\frac{\rho^+_{min}(T)}{p_{max}(T)}L_{sup}(\mathcal{T},\hat{f})\right] \leq L_{un}(\hat{f}) - \tau_k \mathop{\mathbb{E}}_{c^+,c_i^-\sim\rho^{k+1}}\left[\ell_{|I^+|}(\vec{0}) \;\middle|\; I^+\neq\emptyset\right] \tag{2.26}$$

Now, by applying Lemma 2.10.2, we bound the generalization error: with probability at least $1-\delta$, $\forall f\in\mathcal{F}$

$$L_{un}(\hat{f}) \leq L_{un}(f) + Gen_M \tag{2.27}$$

However, $L_{un}(f)$ cannot be made arbitrarily small. One can see that for all $f\in\mathcal{F}$, $L_{un}(f)$ is lower bounded by the second term in Equation (2.22), which cannot be made arbitrarily small as $\tau_k > 0$.

$$L_{un}(f) \geq \mathop{\mathbb{E}}_{\substack{c^+,c_i^-\sim\rho^{k+1}\\ x,x^+\sim\mathcal{D}_{c^+}\\ x_i^-\sim\mathcal{D}_{c_i^-}}}\left[\ell\left(\left\{f(x)^T\left(f(x^+)-f(x_i^-)\right)\right\}_{i\in I^+}\right)\right] \geq \tau \mathop{\mathbb{E}}_{c^+,c_i^-\sim\rho^{k+1}}\left[\ell_{|I^+|}(\vec{0}) \;\middle|\; I^+\neq\emptyset\right] \tag{2.28}$$

where we applied Jensen's inequality. Since τ_k is not 0, the above quantity can never be arbitrarily close to 0 (no matter how rich $\mathcal{F}$ is).

Step 3 (L_{un} decomposition) Now, we decompose $L_{un}(f)$ by applying the RHS of Equation (2.21)

$$\mathcal{L}_{un}(f) \leq \mathop{\mathbb{E}}_{\substack{c^+,c_i^-\sim\rho^{k+1}\\ x,x^+\sim\mathcal{D}^2_{c^+}\\ x_i^-\sim\mathcal{D}_{c_i^-}}}\left[\ell\Big(\Big\{f(x)^T\big(f(x^+)-f(x_i^-)\big)\Big\}_{i\notin I^+}\Big) + \ell\Big(\Big\{f(x)^T\big(f(x^+)-f(x_i^-)\big)\Big\}_{i\in I^+}\Big)\right] \tag{2.29}$$

$$= \mathop{\mathbb{E}}_{\substack{c^+,c_i^-\sim\rho^{k+1}\\ x,x^+\sim\mathcal{D}^2_{c^+}\\ x_i^-\sim\mathcal{D}_{c_i^-},\ i\notin I^+}}\left[\ell\Big(\Big\{f(x)^T\big(f(x^+)-f(x_i^-)\big)\Big\}_{i\notin I^+}\Big)\right] + \mathop{\mathbb{E}}_{\substack{c^+,c_i^-\sim\rho^{k+1}\\ x,x^+\sim\mathcal{D}^2_{c^+}\\ x_i^-\sim\mathcal{D}_{c_i^-},\ i\in I^+}}\left[\ell\left(\left\{f(x)^T\left(f(x^+)-f(x_i^-)\right)\right\}_{i\in I^+}\right)\right] \tag{2.30}$$

$$\begin{aligned} &= (1-\tau') \mathop{\mathbb{E}}_{\substack{c^+,c_i^-\sim\rho^{k+1}\\ x,x^+\sim\mathcal{D}^2_{c^+}\\ x_i^-\sim\mathcal{D}_{c_i^-},i\notin I^+}}\left[\ell\Big(\Big\{f(x)^T\big(f(x^+)-f(x_i^-)\big)\Big\}_{i\notin I^+}\Big)\Big||I^+|<k\right] \\ &\quad +\tau_k \mathop{\mathbb{E}}_{\substack{c^+,c_i^-\sim\rho^{k+1}\\ x,x^+\sim\mathcal{D}^2_{c^+}\\ x_i^-\sim\mathcal{D}_{c_i^-},\ i\in I^+}}\left[\ell\left(\left\{f(x)^T\left(f(x^+)-f(x_i^-)\right)\right\}_{i\in I^+}\right)\Big|I^+\neq\emptyset\right] \end{aligned} \tag{2.31}$$

Observe that the first term is exactly $(1-\tau')L^{\neq}_{un}(f)$. Thus, combining Equations (2.26), (2.27) and (2.31) we get

$$\begin{aligned}(1-\tau_k)\mathop{\mathbb{E}}_{\mathcal{T}\sim\mathcal{D}}\left[\frac{\rho^+_{min}(T)}{p_{max}(T)}L_{sup}(\mathcal{T},\hat{f})\right] &\leq (1-\tau')L^{\neq}_{un}(f)+Gen_M \\ &+\tau_k\underbrace{\mathop{\mathbb{E}}_{c^+,c_i^-\sim\rho^{k+1}}\left[\mathop{\mathbb{E}}_{\substack{x,x^+\sim\mathcal{D}^2_{c^+}\\ x_i^-\sim\mathcal{D}_{c_i^-},\ i\in I^+}}\left[\ell\Big(\Big\{f(x)^T\big(f(x^+)-f(x_i^-)\big)\Big\}_{i\in I^+}\Big)\right]-\ell_{|I^+|}(\vec{0})\ \middle|\ I^+\neq\emptyset\right]}_{\Delta(f)}\end{aligned} \tag{2.32}$$

From the definition of I^+, $c_i^- = c^+$, $\forall i \in I^+$. Thus, from Lemma 2.10.1, we get that

$$\Delta(f) \leq c' \mathop{\mathbb{E}}_{c^+,c_i^-\sim\rho^{k+1}}\left[|I^+|\sqrt{\|\Sigma(f,c)\|_2}\mathop{\mathbb{E}}_{x\sim\mathcal{D}_c}[\|f(x)\|]\ \middle|\ I^+\neq\emptyset\right] \tag{2.33}$$

for some constant c'.

Let u be a distribution over classes with $u(c) = \mathbb{P}_{c^+,c_i^-\sim\rho^{k+1}}[c^+ = c|I^+\neq\emptyset]$ and it is easy to see that $u(c)\propto\rho(c)\big(1-(1-\rho(c))^k\big)$ By applying the tower property to Equation (2.33) we have

$$\Delta(f) \leq c' \mathop{\mathbb{E}}_{c\sim u}\left[\mathop{\mathbb{E}}_{c^+,c_i^-\sim\rho^{k+1}}\big[|I^+|\big|c^+=c,I^+\neq\emptyset\big]\ \sqrt{\|\Sigma(f,c)\|_2}\mathop{\mathbb{E}}_{x\sim\mathcal{D}_c}[\|f(x)\|]\right] \tag{2.34}$$

But,

$$\begin{aligned}\mathop{\mathbb{E}}_{c^+,c_i^-\sim\rho^{k+1}}\big[|I^+|\big|c^+=c,I^+\neq\emptyset\big] &= \sum_{i=1}^{k}\mathbb{P}_{c^+,c_i^-\sim\rho^{k+1}}\big(c_i^-=c^+\big|c^+=c,I^+\neq\emptyset\big)\\ &= k\mathbb{P}_{c^+,c_i^-\sim\rho^{k+1}}\big(c_1^-=c^+\big|c^+=c,I^+\neq\emptyset\big)\\ &= k\frac{\mathbb{P}_{c^+,c_i^-\sim\rho^{k+1}}\big(c_1^-=c^+=c\big)}{\mathbb{P}_{c^+,c_i^-\sim\rho^{k+1}}\big(c^+=c,I^+\neq\emptyset\big)}\\ &= k\frac{\rho^2(c)}{\rho(c)\big(1-(1-\rho(c))^k\big)} = k\frac{\rho(c)}{1-(1-\rho(c))^k}\end{aligned} \tag{2.35}$$

Now, using the fact that $\tau_k = 1 - \sum_{c'} \rho(c')(1-\rho(c'))^k = \sum_{c'} \rho(c') \left(1 - (1-\rho(c'))^k\right)$ and $\tau_1 = \sum_c \rho^2(c)$,

$$\begin{aligned}
\frac{\tau_k}{1-\tau_k}\Delta(f) &\leq \frac{\tau_k}{1-\tau_k} c' \mathop{\mathbb{E}}_{c\sim u}\left[k\frac{\rho(c)}{1-(1-\rho(c))^k}\sqrt{\|\Sigma(f,c)\|_2}\mathop{\mathbb{E}}_{x\sim\mathcal{D}_c}[\|f(x)\|]\right] \\
&= c'k\frac{\tau_k}{1-\tau_k}\sum_c \frac{\rho^2(c)}{\sum_{c'}\rho(c')\left(1-(1-\rho(c'))^k\right)}\sqrt{\|\Sigma(f,c)\|_2}\mathop{\mathbb{E}}_{x\sim\mathcal{D}_c}[\|f(x)\|] \qquad (2.36)\\
&= c'k\frac{\tau_1}{1-\tau_k}\mathop{\mathbb{E}}_{c\sim\nu}\left[\sqrt{\|\Sigma(f,c)\|_2}\mathop{\mathbb{E}}_{x\sim\mathcal{D}_c}[\|f(x)\|]\right] = c'k\frac{\tau_1}{1-\tau_k}s(f)
\end{aligned}$$

and we are done. □

2.11.2 Competitive bound

As in Section 2.5.2, we prove a competitive type of bound, under similar assumptions. Let $\ell_\gamma(\boldsymbol{v}) = \max\{0, 1+\max_i\{-\boldsymbol{v}_i\}/\gamma\}$, $\boldsymbol{v}\in\mathbb{R}^k$, be the multiclass hinge loss with margin γ and for any $\mathcal{T}$ let $L^\mu_{\gamma,sup}(\mathcal{T}, f)$ be $L^\mu_{sup}(\mathcal{T}, f)$ when ℓ_γ is used as loss function. For all tasks $\mathcal{T}$, let ${\rho'}^+(\mathcal{T})$ is the distribution of c^+ when $(c^+, c_1^-, ..., c_k^-)$ are sampled from ρ^{k+1} conditioned on $Q = \mathcal{T}$ and $|I^+| < k$. Also, let ${\rho'}^+_{max}(\mathcal{T})$ be the maximum of these $|\mathcal{T}|$ probabilities and $p_{min}(\mathcal{T}) = \min_{c\in\mathcal{T}}\mathcal{D}_\mathcal{T}(c)$.

We will show a competitive bound against the following quantity, for all $f\in\mathcal{F}$: $\mathop{\mathbb{E}}_{\mathcal{T}\sim\mathcal{D}'}\left[\frac{{\rho'}^+_{max}(\mathcal{T})}{p_{min}(\mathcal{T})}\mathcal{L}^\mu_{\gamma,sup}(\mathcal{T},f)\right]$, where $\mathcal{D}'$ is defined as follows: sample $c^+, c_1^-, ..., c_k^- \sim \rho^{k+1}$, conditioned on $|I^+| < k$. Then, set $\mathcal{T} = Q$. Observe that when $I^+ = \emptyset$ with high probability, we have $\mathcal{D}' \approx \mathcal{D}$.

Lemma 2.11.2. *For all $f\in\mathcal{F}$ suppose the random variable $f(X)$, where $X\sim D_c$, is $\sigma^2(f)$-subgaussian in every direction for every class c and has maximum norm $R(f) = max_{x\in\mathcal{X}}\|f(x)\|$. Let $\widehat{f}\in\arg\min_{f\in\mathcal{F}}\widehat{L}_{un}(f)$. Then for all $\epsilon>0$, with probability at least $1-\delta$, for all $f\in\mathcal{F}$*

$$\mathop{\mathbb{E}}_{\mathcal{T}\sim\mathcal{D}}\left[\frac{\rho^+_{min}(\mathcal{T})}{p_{max}(\mathcal{T})}\,L^\mu_{sup}(\mathcal{T},\hat{f})\right] \leq \alpha\gamma(f)\mathop{\mathbb{E}}_{\mathcal{T}\sim\mathcal{D}'}\left[\frac{{\rho'}^+_{max}(\mathcal{T})}{p_{min}(\mathcal{T})}\mathcal{L}^\mu_{\gamma,sup}(\mathcal{T},f)\right] + \beta s(f) + \eta Gen_M + \epsilon$$

where $\gamma(f) = 1 + c'R(f)\sigma(f)(\sqrt{\log k} + \sqrt{\log\frac{R(f)}{\epsilon}})$, c' is some constant, $\alpha = \frac{1-\tau'}{1-\tau_k}$, $\beta = k\frac{\tau_1}{1-\tau_k}$ and $\eta = \frac{1}{1-\tau_k}$.

Proof. We will show that $\forall f\in\mathcal{F}$

$$L^{\neq}_{un}(f) \leq \gamma(f)\mathop{\mathbb{E}}_{\mathcal{T}\sim\mathcal{D}'}\left[\frac{{\rho'}^+_{max}(\mathcal{T})}{p_{min}(\mathcal{T})}\mathcal{L}^\mu_{\gamma,sup}(\mathcal{T},f)\right] \qquad (2.37)$$

and the Lemma follows from Theorem 2.6.1. Now, we fix an $\epsilon>0$, an $f\in\mathcal{F}$ and we drop most of the

arguments f in the rest of the proof. Also, fix $c^+, c_1^- \dots c_k^-, x$ and let $t = k - |I^+|$. We assume without loss of generality, that $c^+ \neq c_i^-$, $\forall i \in [t]$. Now,

$$\max_{i\in[t]} f(x)^T(f(x_i^-) - f(x^+)) \leq \mu + \max_i z_i^- - z^+ \tag{2.38}$$

where $\mu = \max_{i\in[t]} f(x)^T(\mu_{c_i^-} - \mu_{c^+})$, $z_i^- = f(x)^T(f(x_i^-) - \mu_{c_i^-})$ and $z^+ = f(x)^T(f(x^+) - \mu_{c^+})$. z_i are centered σ^2R^2-subgaussian, so from standard properties of subgaussian random variables $\mathbb{P}[\max_i z_i^- \geq \sqrt{2}\sigma R\sqrt{\log t} + \sqrt{2c_1}\sigma R\sqrt{\log R/\epsilon}] \leq (\epsilon/R)^{c_1}$ (again we consider here the case where $R \geq 1$ and for $R < 1$, the same arguments hold but with removing R from the log). z^+ is also centered σ^2R^2-subgaussian, so $\mathbb{P}[z^+ \geq \sqrt{2c_1}\sigma R\sqrt{\log R/\epsilon}] \leq (\epsilon/R)^{c_1}$. Let $\gamma = 1 + c'\sigma R(\sqrt{\log t} + \sqrt{\log R/\epsilon})$ for appropriate constant c'. By union bound, we have $p = \mathbb{P}[\max_i z_i^- - z^+ \geq \gamma - 1] \leq 2(\epsilon/R)^{c_1}$. Thus, $\mathbb{E}_{z^+, z_i^-}[(1 + \mu + \max_i z_i^- - z^+)_+] \leq (1-p)(\mu+\gamma)_+ + p(2R^2+1) \leq \gamma(1+\mu/\gamma)_+ + \epsilon$ (for appropriate constant c_1). By taking expectation over $c^+, c_i^- \sim \rho^{k+1}$, conditioned on $|I^+| < k$, and over $x \sim \mathcal{D}_{c^+}$ we get

$$\begin{aligned}
L_{un}^{\neq}(f) &\leq \gamma \mathop{\mathbb{E}}_{\substack{c^+, c_i^- \sim \rho^{k+1} \\ x\sim\mathcal{D}_{c^+}}} \left[\left(1 + \frac{\max_{c\in Q, c\neq c^+} f(x)^T(\mu_c - \mu_{c^+})}{\gamma}\right)_+ \middle| |I^+| < k \right] \\
&= \gamma \mathop{\mathbb{E}}_{\mathcal{T}\sim\mathcal{D}'} \mathop{\mathbb{E}}_{\substack{c^+, c_i^- \sim \rho^{k+1} \\ x\sim\mathcal{D}_{c^+}}} \left[\left(1 + \frac{\max_{c\in Q, c\neq c^+} f(x)^T(\mu_c - \mu_{c^+})}{\gamma}\right)_+ \middle| Q = \mathcal{T}, |I^+| < k \right] \\
&= \gamma \mathop{\mathbb{E}}_{\mathcal{T}\sim\mathcal{D}'} \mathop{\mathbb{E}}_{\substack{c^+ \sim \rho'^+(\mathcal{T}) \\ x\sim\mathcal{D}_{c^+}}} \left[\left(1 + \frac{\max_{c\in T, c\neq c^+} f(x)^T(\mu_c - \mu_{c^+})}{\gamma}\right)_+ \right] \leq \gamma \mathop{\mathbb{E}}_{\mathcal{T}\sim\mathcal{D}'} \left[\frac{\rho'^+_{max}(\mathcal{T})}{p_{min}(\mathcal{T})} \mathcal{L}^{\mu}_{\gamma,sup}(\mathcal{T}, f) \right]
\end{aligned} \tag{2.39}$$

□

2.12 Examples for Section 2.6.2

Here, we illustrate via examples two ways in which the increase of k can lead to suboptimal $\hat{f}$. We will consider the hinge loss as the loss function, while the examples carry over trivially for logistic loss.

1. The first example is the case where even though there exist representations in $\mathcal{F}$ that can separate every class, the suboptimal representation is picked by the algorithm when $k = \Omega(|\mathcal{C}|)$. Let $\mathcal{C} = \{c_i\}_{i\in[n]}$ where

for each class, D_{c_i} is uniform over two points $\{x_i^1, x_i^2\}$. Let e_i be the indicator vectors in $\mathbb{R}^n$ and let the class $\mathcal{F}$ consists of $\{f_0, f_1\}$ with $f_0, f_1 : \mathcal{X} \mapsto \mathbb{R}^n$ where $f_1(x_i^1) = 3/2re_i$ and $f_1(x_i^2) = 1/2re_i$ for all i, for some $r > 0$, and $f_0 = \vec{0}$. Finally, ρ is uniform over $\mathcal{C}$. Now, when the number of negative samples is $\Omega(n)$, the probability that $\exists j \in [k]$ such that $c^+ = c_j^-$ is constant, and therefore $L_{un}(f) = \Omega(r^2) > 1 = L_{un}(f_0)$ when r is large. This means that despite $L_{sup}(\mathcal{C}, f_1) = 0$, the algorithm will pick f_0 which is a suboptimal representation.

2. We can extend the first example to the case where, even when $k = o(|\mathcal{C}|)$, the algorithm picks suboptimal representations. To do so, we simply 'replicate' the first example to create clusters of classes. Formally, let $\mathcal{C} = \{c_{ij}\}_{i,j\in[n]}$ where for each class, $D_{c_{ij}}$ is uniform over two points $\{x_{ij}^1, x_{ij}^2\}$. Finally, same as above, let $\mathcal{F}$ consist of two functions $\{f_0, f_1\}$. The function f_1 maps $f_1(x_{ij}^1) = 3/2re_i$ and $f_1(x_{ij}^2) = 1/2re_i$ for all i, j and $f_0 = \vec{0}$. ρ is uniform over $\mathcal{C}$. Now, note that f_1 'clutsters' the n^2 classes and their points into n clusters, each along an e_i. Thus, it is only useful for contrasting classes from different clusters. However, note that the probability of intra-cluster collision with k negative samples is $1-(1-1/n)^k$. When $k = o(n)$, we have that $L_{un}(f_1) = o(1) < 1 = L_{un}(f_0)$ so the algorithm will pick f_1. However, when $k = \Omega(n)$, $L_{un}(f) = \Omega(r^2) > 1 = L_{un}(f_0)$ and the algorithm will pick the suboptimal representation f_0. Thus, despite $|\mathcal{C}| = n^2$, having more than n negative samples can hurt performance, since even tough f_1 cannot solve all the tasks, the average supervised loss over t-way tasks, $t = o(n)$, is $L_{sup}(f) \leq O(1 - (1 - 1/n)^{t-1}) = o(1)$.

2.13 Experiment details

2.13.1 Wiki-3029 construction

We use the Wikipedia dump and select articles that have entries in the WordNet, have at least 8 sections and at least 12 sentences of length at least 4 per section. At the end of this filtering we are left with 3029 articles with at least 200 sentences per article. We then sample 200 sentences from each article and do a 70%/10%/20% train/dev/test split.

2.13.2 GRU model

We use a bi-directional GRU with output dimension of 300 trained using dropout 0.3. The input word embeddings are initialized to pretrained CC GloVe vectors and fixed throughout training.

Chapter 3

Understanding Contrastive Learning Requires Incorporating Inductive Biases

This chapter studies contrastive learning with data augmentations, based on previously published work [Saunshi et al., 2022]. In this setting, contrastive learning encourages augmentations (views) of the same input to have more similar representations compared to augmentations of different inputs. Recent attempts to theoretically explain the success of contrastive learning on downstream classification tasks prove guarantees depending on properties of *augmentations* and the value of *contrastive loss* of representations. We demonstrate that such analyses, that ignore *inductive biases* of the function class and training algorithm, cannot adequately explain the success of contrastive learning, even *provably* leading to vacuous guarantees in some settings. Extensive experiments on image and text domains highlight the ubiquity of this problem – different function classes and algorithms behave very differently on downstream tasks, despite having the same augmentations and contrastive losses. Theoretical analysis is presented for the class of linear representations, where incorporating inductive biases of the function class allows contrastive learning to work with less stringent conditions compared to prior analyses.

3.1 Introduction

Recently, representation functions learned via contrastive learning have transformed machine learning. Using unlabeled data, a representation function is learnt by generating simple augmentations of each datapoint and by enforcing, via a suitable loss function, that (1) augmentations of a single datapoint tend to be clustered (2) augmentations of different datapoints tend to be far apart. Such representations give competitive classification performance —via even a linear classifier— on a host of downstream tasks, bringing us closer to the old dream of machine learners capable of generalization across different data distributions and tasks.

We lack a conceptual framework for understanding such phenomena — which is unsurprising, since good quantitative understanding of generalization is lacking even for single task and single data distribution. However, deriving even partial conceptual understanding could help push the field forward, and researchers have begun to grapple with this task [Arora et al., 2019, Tosh et al., 2021a, HaoChen et al., 2021]. The current work seeks to provide guidance for further development of this nascent theory[1] using simple experiments and theoretical analysis. A common thread in these existing theories is the following components: (1) Quantifying how data augmentations *implicitly* encode downstream class labels. (2) Demonstrating how representations with small contrastive loss can uncover this implicit structure and do well on downstream tasks.

Recent works formalize (1) via assumptions that end up implying that the augmentation distributions of inputs from the same class have significant overlap, but there is little overlap for inputs from different classes. For example, distributions of augmentations of different dog images tend to be similar to each other, but their union has little overlap with distributions of augmentations of cat images. Arora et al. [2019] — which predates the recent wave of methods — assume that points in the same class share the same augmentation distribution, and use this to show that the contrastive loss is a surrogate to the downstream performance. Since methods like SimCLR [Chen et al., 2020a] do not appear to satisfy such assumptions, recently HaoChen et al. [2021] gave a more refined analysis under milder assumptions, that requires only some overlap in augmentation distributions, such that the resultant graph of connections due to overlaps within a class is dense. Again, it can be shown that a low-dimensional representation that is near-optimal in the contrastive loss is guaranteed to linearly separate the downstream classes.

Note that properties of the class of representation functions (architecture) or the training algorithm (SGD, Adam etc.) make no appearance in the above analyses; but only properties of the *augmentation distributions*

[1] Our title is a clear allusion to Zhang et al. [2017a], which highlighted a gap between deep learning phenomena and classical ML theory, motivating development of better theoretical understanding.

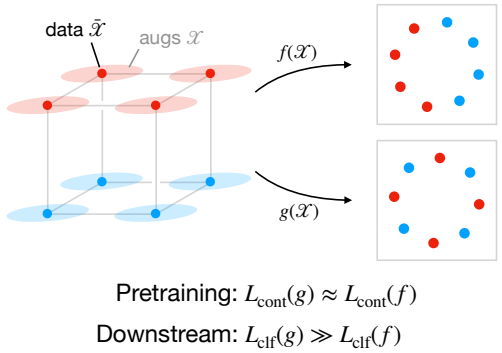


Figure 3.1: Cartoon of our theoretical example. The downstream labels (and thus classification loss L_{clf}) are determined by a few relevant attributes (e.g. cat or dog?), and the augmentations perturb irrelevant attributes (e.g. grayscale, random crop). Without restricting the function class for the contrastive pretraining task, there exist perfect (f) and spurious (g) *augmentation-invariant* representations which both minimize the contrastive loss L_{cont}. However, minimizing using a linear representation class is always guaranteed to succeed with these augmentations (Section 3.3).

and *value of contrastive loss* of representations. This is understandable since currently theory is unable to pinpoint why different real-life architectures differ in their capabilities, or to pinpoint the implicit bias of the training algorithms. Nevertheless it raises interesting questions: *Is the contrastive loss indeed a good indicator of downstream performance? Do augmentations overlap sufficiently enough in practice to explain the success of contrastive learning? Can contrastive learning succeed even when there is little to no overlap?* In a nutshell, the current work suggests via experiments and simple theory that the answers are, respectively: *No, Maybe no, Yes.* In particular, ignoring the architecture and the training algorithm can make the current theoretical analyses of contrastive learning vacuous. We present three key phenomena:

- **Function class sensitivity.** Downstream performance of a representation depends not just on its contrastive loss, but it is also sensitive to the function class (architecture) and training procedure used to learn it.
- **Brittleness of transfer.** Minimizing the contrastive loss to optimality can sometimes have a non-monotonic, deleterious effect on downstream performance, despite the augmentations being effective for some function classes.
- **The disjoint augmentations regime.** When augmentation distributions for inputs do not overlap, it can be shown that any function-class-agnostic analysis (subsumes prior work) provably leads to vacuous guarantees. However on-overlapping augmentations can still be "informative", and contrastive learning with appropriate function classes can succeed, a phenomenon unexplained by existing theory.

Organization. We define the contrastive losses and downstream performance in Section 3.2, and summarize prior theoretical results and how they ignore inductive biases. In Section 3.3 we describe a simple synthetic setting that elucidates all of the aforementioned phenomena. A pictorial depiction in Figure 3.1 demonstrates the existence of bad contrastive solutions, despite the augmentations satisfying intuitive properties. These ideas are grounded through theoretical results in Section 3.4, which includes lower bounds for function class agnostic analyses and upper bounds that are sensitive to the function class of linear representations. Finally we describe various experimental setups in Section 3.5.

3.1.1 Related work

Contrastive learning has been very successful at solving downstream tasks by learning representations from similar pairs of data obtained using temporal information [Wang and Gupta, 2015, Logeswaran and Lee, 2018] or different views or augmentations of inputs [Dosovitskiy et al., 2014, Hjelm et al., 2019, Wu et al., 2018, Bachman et al., 2019, Tian et al., 2020a, He et al., 2020, Chen et al., 2020a, Chen and He, 2021, Gao et al., 2021]. Given its empirical success, there has been significant interest in the theory of contrastive learning, from various perspectives. Most relevant to us are learning theoretic analyses [Arora et al., 2019, Tosh et al., 2021b,a, HaoChen et al., 2021, Wang et al., 2022] and their follow ups [Nozawa and Sato, 2021, Ash et al., 2022]. These study the downstream linear classification performance of learned representation, by making assumptions about the data and augmentation distributions.

Contrastive learning has also been studied (1) from a mutual information maximization view [Oord et al., 2018, Hjelm et al., 2019, Bachman et al., 2019]; [Tschannen et al., 2020] points out certain issues with this view, (2) using an information theoretic framework Tsai et al. [2021]; fails to explain downstream success via simple linear classifiers, (3) through properties like alignment and uniformity on the sphere Wang and Isola [2020], (4) under certain latent variable data generative processes [Zimmermann et al., 2021, Von Kügelgen et al., 2021], and (5) through a causality perspective [Mitrovic et al., 2021]. On the optimization front, [Wen and Li, 2021] study the feature learning process of contrastive learning with gradient dynamics on a two layer network, under a sparse coding model. The theory of noise contrastive estimation [Gutmann and Hyvärinen, 2010] has been a useful motivation for negative sampling based objectives. On the empirical side, there are studies on identifying useful augmentation properties [Tian et al., 2020b].

Non-contrastive methods, with no negative samples, [Chen and He, 2021, Grill et al., 2020] rely on tricks like stop-grad to avoid representation collapse. Dimension collapse of representations has also been studied

[Jing et al., 2022]. Unlike these works, the brittleness of transfer we study is neither due to ill-designed objectives nor due to training degeneracies. It is fundamental to data distributions and arises out of existence of spurious solutions. A related idea of feature suppression [Chen et al., 2021] and shortcut solutions found by contrastive learning was recently studied in Robinson et al. [2021] in certain stylized settings, with a proposed fix through better augmentations strategies. We instead study the role of inductive bias of function classes in avoiding such shortcut solutions. [Abnar et al., 2022] analyze upstream to downstream transfer for supervised pre-training, complementing our experiments for unsupervised pre-training, whereas Wu et al. [2020a] studies negative transfer for multi-task learning. Robinson et al. [2021] also observe negative transfer for contrastive learning in some settings. Finally, there are theoretical works for other types of self-supervised learning [Bansal et al., 2021], including methods like context reconstruction [Lee et al., 2021] and language model [Saunshi et al., 2021], studying their benefits on downstream tasks.

3.2 Preliminaries

Here we formalize the problem of learning useful representations via contrastive learning for downstream classification.

Notation. We use $[n]$ for the set $\{1, \ldots, n\}$. $\mathcal{U}(S)$ denotes uniform distribution over a set S. For a vector $v \in \mathbb{R}^n$, we denote $v_{:i} \in \mathbb{R}^i$ and $v_{i:} \in \mathbb{R}^{n-i}$ to be the sub-vector of first $i \in [n]$ and last i coordinates respectively. For sets P, Q, we use P^Q to denote the set of functions from Q to P.

Augmentations. We use $\bar{\mathcal{X}}$ to denote the set of all (unaugmented) samples and denote their marginal distribution as $\mathcal{D}_{\bar{\mathcal{X}}}$. $\mathcal{X}$ denotes the set of all augmented data. For an input $\bar{x} \in \bar{\mathcal{X}}$, we define the corresponding augmentation distribution over $\mathcal{X}$ as $\mathcal{A}(\cdot \mid \bar{x})$. For instance, augmentations for an image $\bar{x}$ can correspond to applying a sequence of random transformations such as random cropping, Gaussian blur, and color jitter. The distributions $\mathcal{D}_{\bar{\mathcal{X}}}$ and $\mathcal{A}$ together induce a marginal distribution $\mathcal{D}_{\mathcal{X}}$ over augmentations.

Contrastive self-supervised learning. The goal is to learn a representation function $f : \mathcal{X} \to \mathbb{R}^d$ that maps augmentations to d-dimensional vectors by encouraging representations of "similar pairs" of augmentations to be closer to each other, compared to representations of random pairs. A common strategy to pick a similar pair (x, x^+) is to pick two augmentations of the same input. Formally we define this distribution

of similar pairs $\mathcal{D}_{\text{sim}}$ as follows

$$(x, x^+) \sim \mathcal{D}_{\text{sim}} \;\equiv\; \bar{x} \sim \mathcal{D}_{\bar{\mathcal{X}}};\; x, x^+ \sim_{\text{i.i.d.}} \mathcal{A}(\cdot \mid \bar{x})$$

The negative sampling distribution, denoted by $\mathcal{D}_{\text{neg}}$, is picked to be the augmentation marginal distribution $\mathcal{D}_{\mathcal{X}}$. There are several variants of the contrastive loss, a popular one being the *SimCLR loss* [Chen et al., 2020a].

$$L_{\text{SimCLR}}(f) = \mathop{\mathbb{E}}_{(x,x^+)\sim\mathcal{D}_{\text{sim}}, x_{1:n}^-\sim\mathcal{D}_{\text{neg}}^n} \left[-\log \left(\frac{e^{f(x)^\top f(x^+)}}{e^{f(x)^\top f(x^+)} + \sum_{i=1}^n e^{f(x)^\top f(x_i^-)}} \right) \right] \tag{3.1}$$

Intuitively the contrastive loss aims to make $f(x)^\top f(x^+)$ larger compared to $f(x)^\top f(x_i^-)$. Another variant proposed in HaoChen et al. [2021] is the spectral contrastive loss:

$$L_{\text{spec}}(f) = \mathop{\mathbb{E}}_{(x,x^+)\sim\mathcal{D}_{\text{sim}}} \left[-2 f(x)^\top f(x^+) \right] + \mathop{\mathbb{E}}_{x,x^-\sim\mathcal{D}_{\text{neg}}^2} \left[\left(f(x)^\top f(x^-) \right)^2 \right] \tag{3.2}$$

We will use L_{cont} to refer to a generic contrastive loss, either L_{SimCLR} or L_{spec} or something else.

Downstream task. We assume these involve binary classification[2]. If the ground-truth labeling function is $\bar{y}^\star : \bar{\mathcal{X}} \to \{\pm 1\}$, the quality of representation $\bar{f} : \bar{\mathcal{X}} \to \mathbb{R}^d$ is captured by how well it allows *linear classification*:

$$L_{\text{clf}}(\bar{f}; \bar{y}^\star) = \inf_{w \in \mathbb{R}^d} \mathbb{E}_{\bar{x}} \left[\mathbb{1} \left\{ \bar{y}^\star(\bar{x}) \left(\bar{f}(\bar{x})^\top w \right) < 0 \right\} \right] \tag{3.3}$$

Since the representation function is trained to map augmentations to vectors, its behavior on unaugmented inputs can be undefined. We evaluate downstream performance on original inputs $\bar{\mathcal{X}}$ by using the average augmentation representation $f_{\mathcal{A}} : \bar{\mathcal{X}} \to \mathbb{R}^d$, defined as:

$$f_{\mathcal{A}}(\bar{x}) = \mathop{\mathbb{E}}_{x\sim\mathcal{A}(\cdot|\bar{x})} [f(x)],\; L_{\text{clf}}(f; \bar{y}^\star) \coloneqq L_{\text{clf}}(f_{\mathcal{A}}; \bar{y}^\star) \tag{3.4}$$

Experimentally such an average gives better performance than the standard un-averaged approach.

Transfer Bounds. We introduce an abstraction of *transfer function* $\mathcal{T}$ to capture prior analyses [Arora

[2]We consider binary tasks mostly for simplicity. Extensions of our results (lower bounds for function class agnostic analyses and upper bound guarantees for linear representations) to more than two classes are not difficult.

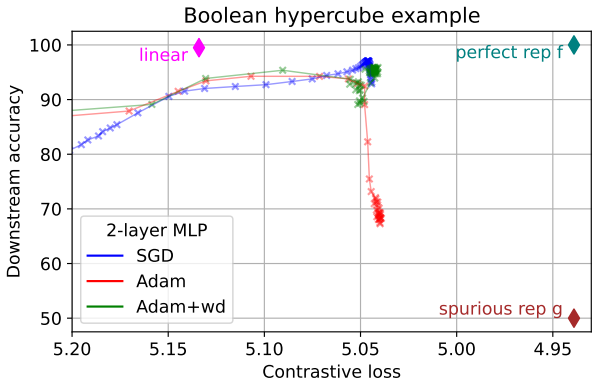


Figure 3.2: Contrastive loss → accuracy transfer plots for the Boolean hypercube example. There exist global minimizers of L_{cont} with perfect (top right) and worst possible (bottom right) downstream classification error L_{clf}. The representations learned by two-layer neural networks are very sensitive to training configuration. With a smaller (linear) function class, the contrastive loss minimizer gives a nearly-perfect downstream classifier.

et al., 2019, Tosh et al., 2021a, HaoChen et al., 2021]. It translates performance on contrastive loss to performance on the downstream task as follows:

$$L_{\text{clf}}(f; \bar{y}^\star) \leq \mathcal{T}(\Gamma, L_{\text{cont}}(f), d), \text{ where } \Gamma = (\mathcal{D}_{\bar{\mathcal{X}}}, \mathcal{A}, \bar{y}^\star, L_{\text{cont}}) \tag{3.5}$$

These guarantees only depend on (1) problem dependent quantities like input marginals $\mathcal{D}_{\bar{\mathcal{X}}}$, properties of augmentations $\mathcal{A}$, downstream label $\bar{y}^\star$, form of contrastive loss L_{cont}, (2) contrastive loss $L_{\text{cont}}(f)$ of the representation f and (3) its dimensionality d. Typically $\mathcal{T}$ is monotone non-decreasing function of $L_{\text{cont}}(f)$ and so the above bound justifies minimizing the contrastive loss. A common property of augmentations and labels that transfer bounds assume is *overlap* between augmentations of images from the same class. For instance, Arora et al. [2019] effectively assume full overlap, that is, all images from the same class have identical augmentation distribution, and these distributions are different for different classes. HaoChen et al. [2021] relax this requirement to a spectral quantity that depends on the ratio of overlap between augmentation distributions of the same class and those of different classes, leading to a bound $L_{\text{clf}}(f) \leq \alpha(L_{\text{cont}}(f) - \min_{f^\star} L_{\text{cont}}(f^\star)) + \beta$, where $f^\star$ is a minimizer of the contrastive loss, and α, β quantify the overlap in augmentations. These bounds place a premium on the value of contrastive loss of f, but are agnostic to any other properties of f, like the representation function class $\mathcal{F}$ it belongs to or how it was trained. We are interested in transfer bounds

Table 3.1: Contrastive loss and downstream accuracy of various representation classes and training procedures used to train on the hypercube example. There exist minimizers of the contrastive loss which transfer to perfect downstream classifiers, and ones which are no better than random guessing. Function class matters (MLP vs. linear vs. any representation), as does the training algorithm.

Representation	Contrastive loss	Accuracy (%)
$\exists f$ (perfect)	4.939	100
$\exists g$ (spurious)	4.939	50
MLP + Adam	5.039 $\pm$ 0.001	74.1 $\pm$ 4.3
MLP + Adam + wd	5.040 $\pm$ 0.002	89.5 $\pm$ 4.9
Linear	5.134 $\pm$ 0.002	99.5 $\pm$ 0.1

that also incorporate these effects. A simple abstraction that incorporates the function class is

$$L_{\text{clf}}(f; \bar{y}^\star) \leq \mathcal{T}(\Gamma, L_{\text{cont}}(f), \mathcal{F}), \text{ where } \Gamma = (\mathcal{D}_{\bar{\mathcal{X}}}, \mathcal{A}, \bar{y}^\star, L_{\text{cont}}) \tag{3.6}$$

A bound like this, unlike the one in Equation (3.5), reflects that the downstream performance at a particular value of contrastive loss depends also on the representation function class, which we also find to be true in many experiments.

Not about generalization. The above bounds only deal with upstream and downstream *population losses*. Thus the role of function class bias is not for guaranteeing good generalization properties with finite samples, as in supervised learning. It pertains to how good performance on a pre-training task can guarantee good downstream performance, as will become evident in the following sections.

3.3 Warm-up: contrastive learning on hyper🧊s

In this section, we present a simple but illustrative example that succinctly highlights brittleness of transfer and the importance of incorporating inductive biases in transfer bounds. Since contrastive learning tries to make representations invariant to augmentation transformations, ideal augmentations are those that retain parts of the input that can predict the downstream label, but modify parts that are less important for the label. We now describe a simple example on the Boolean hypercube that captures these intuitions.

Example 3.3.1. The input set is $\bar{\mathcal{X}} = \{\pm 1\}^D$, the augmentation set is $\mathcal{X} = \mathbb{R}^D$. Downstream label $\bar{y}^\star$ is linear in the first $k \ll D$ coordinates.

$$\bar{y}^\star(\bar{x}) = \text{sign}({w^\star}^\top \bar{x}_{:k}), \; w^\star \in \mathbb{R}^k$$

Augmentation distribution $\mathcal{A}(\cdot \mid \bar{x})$ for input $\bar{x} \in \bar{\mathcal{X}}$ randomly scales down the last k coordinates while keeps the first k coordinates unchanged[3]. Formally it is defined as

$$x \sim \mathcal{A}(\cdot \mid \bar{x}) \;\equiv\; \tau \sim \mathcal{U}((0,1]),\; x_{:k} = \bar{x}_{:k},\; x_{k:} = \tau \bar{x}_{k:}$$

where $\mathcal{U}((0,1])$ is the uniform distribution over $(0,1]$.

We experimentally study this example using two function classes to minimize the contrastive objective: MLP[4], linear. Results from Table 3.1 and Figure 3.2 are summarized below.

Transfer is sensitive to function class and algorithm. Firstly, we notice that despite having much worse (higher) contrastive loss compared to MLP, linear representation has significantly better downstream performance. Secondly, Figure 3.2 suggests that even for the same MLP architecture, the training algorithm (Adam v/s SGD, weight decay or not) can drastically affect the downstream accuracy.

Brittleness of transfer and disjoint augmentations. Although the augmentations in the example seem intuitively helpful, there exists a spurious representation that has much smaller contrastive loss than all architectures, but random guessing performance downstream. This, as we show in later sections, is a consequence of having disjoint augmentation distributions, since the original input can be recovered from an augmentation by simply performing $\bar{x} = \text{sign}(x)$. The existence of a bad minimizer of the contrastive loss also gives us a concrete case where contrastive learning can succeed with an appropriate function class, but the success cannot be explained by any function class agnostic analysis. With this backdrop, we present our theoretical results next.

3.4 Lower bounds and improved analysis

In this section we discuss the role of overlap in augmentations and function class in theoretical guarantees. We first show in the disjoint augmentation regime (augmentation distributions do not overlap), that *any* function class independent analyses will lead to vacuous bounds, which includes many previous analysis. Delving deeper into the most recent results from HaoChen et al. [2021], we discuss reasons for failure, even in approximately disjoint augmentations. Finally we present guarantees for contrastive learning with a linear representation function class that is sensitive to the function class and allows for weaker assumptions

[3]Generalizable to downscaling subsets of $\bar{x}_{k:}$, analogous to downscaling different aspects of images like color, sharpness
[4]CNN behave similarly to MLP.

on augmentations. We instantiate this bound for the hypercube example, provably explaining the good performance of linear representations on disjoint augmentations.

3.4.1 Lower bound for disjoint augmentations

In this section, we prove that brittle transfer exists much more generically whenever the augmentation distributions for different inputs do not overlap, generalizing our observations from the hypercube example in the previous section.

Definition 3.4.1 (Disjoint augmentations)**.** *We say the augmentation distributions are disjoint if for all distinct inputs $\bar{x}_1, \bar{x}_2 \in \bar{\mathcal{X}}$, augmentation distributions $\mathcal{A}(\cdot \mid \bar{x}_1)$ and $\mathcal{A}(\cdot \mid \bar{x}_2)$ have disjoint supports.*

Disjoint augmentations can be problematic because the contrastive loss only encourages separating individual instances, but does not encourage making classes linearly separable. We formalize this argument in the next two lemmas by showing that any representation f can be transformed — by shuffling identities of examples — to a new representation $\hat{f}$ that has lower (or equal) contrastive loss but near-trivial downstream performance. An immediate consequence is that any function class agnostic analysis (including previous analyses) will necessarily leads to vacuous downstream guarantees. We establish this in two settings where the exact choice of contrastive objective is not critical; results hold for both L_{SimCLR} and L_{spec} and we abbreviate these by L_{cont} below. First we consider unconstrained representations.

Lemma 3.4.2. *Let $|\bar{\mathcal{X}}| = N$ and $d = \mathcal{O}(N/\log_2(N))$. Suppose the labeling function is balanced, i.e. $\sum_i y_i^\star = 0$, and let $\mathcal{D}_{\mathcal{X}}$ be uniform over $\bar{\mathcal{X}}$. If the augmentation distribution is disjoint, then for any $f^\star : \mathcal{X} \to \mathbb{R}^d$ there exists a $\hat{f} : \mathcal{X} \to \mathbb{R}^d$ such that:*

$$L_{cont}(\hat{f}) \le L_{cont}(f^\star), \;\; \& \; L_{clf}(\hat{f}) \ge \frac{1}{2} - O\left(\sqrt{\frac{d\log(N)}{N}}\right).$$

Since it is common to use normalized representations in practice (e.g., to have Euclidean norm 1), we also establish a similar result for this case.

Lemma 3.4.3. *In the setup of Lemma 3.4.2, suppose further that representations are constrained (to any given set) and that the augmentation distributions satisfy the following: There exists a fixed source of randomness W and a deterministic map $T : (\bar{x}, w) \mapsto x$ that is invertible in w for any $\bar{x}$ such that $x \sim \mathcal{A}(\cdot \mid \bar{x}) \equiv w \sim W, x = T(\bar{x}, w)$. Then the conclusion of Lemma 3.4.2 holds.*

We prove both statements jointly in Section 3.9. Both lemmas show that when the representation dimension is small relative to the size of the input space (as is typical) and the augmentations are disjoint, there exists a *global minimizer* of the contrastive loss with vacuous transfer to downstream. The extra assumption in Lemma 3.4.3 is that the augmentation generation protocol uses a common source of randomness, which is actually satisfied in many practical scenarios. For instance, the same sequence of transformations like random cropping, color jittering etc. are applied to all images to generate augmentations. The other assumptions, e.g., that the labeling function $\bar{y}^\star$ is balanced and that $\mathcal{D}_{\mathcal{X}}$ is uniform, are technical in nature and can be potentially relaxed.

Proposition 1 in Robinson et al. [2021] discusses a similar lower bound when augmentations are disjoint, arguing that contrastive learning can find "shortcut solutions" that can lead to feature suppression. While the motivation is similar to ours, those results are shown specifically for contrastive loss with normalized representations, in the regime of large number of negative samples and with a specific uniform over sphere assumption on latent variables generating the data. The above results are shown in much more general settings. Wang et al. [2022] also show a lower bound for why the alignment and uniformity properties from Wang and Isola [2020] is insufficient to guarantee good downstream performance, through a non-overlapping augmentation example.

A corollary of these results is that any transfer learning bound that only depends on the value of the contrastive loss cannot be meaningful in the disjoint augmentations setting.

Corollary 3.4.4. *In the setup of Lemma 3.4.2 or Lemma 3.4.3, consider a transfer function $\mathcal{T}$ bounding the downstream performance as $L_{clf}(f; \bar{y}^\star) \le \mathcal{T}(\Gamma, L_{cont}(f), d)$ as in Equation (3.5), where $\Gamma = (\mathcal{D}_{\bar{\mathcal{X}}}, \mathcal{A}, \bar{y}^\star, L_{cont})$ are problem dependent but function class independent quantities. Suppose $\mathcal{T}$ is monotonic in its second argument, then for all $f : \mathcal{X} \to \mathbb{R}^d$:*

$$\mathcal{T}\left(\Gamma, L_{cont}(f), d\right) \ge 1/2 - \tilde{\mathcal{O}}\left(\sqrt{d/|\bar{\mathcal{X}}|}\right)$$

Takeaways. The above lower bounds suggest that previous analyses for contrastive learning are vacuous in the disjoint augmentation setting, due to existence of bad minimizers of the contrastive loss. The brittleness of transfer for disjoint augmentations is also observable in practice, as in the first row of Table 3.1 for the hypercube example. Vision and NLP experiments in Section 3.5 also demonstrate this phenomenon, for more expressive function classes.

3.4.2 Prior theoretical results and failure modes

We briefly discuss the results from HaoChen et al. [2021] and delve deeper into how their function class agnostic nature leads to poor guarantees even for *approximately* disjoint augmentations. Their analysis considers the spectral loss $L_{\text{spec}}(f)$ from Equation (3.2). A key component of their analysis is an *augmentation graph* constructed using $\mathcal{A}$, whose spectral properties characterize how much overlap there is in augmentations. This is a weighted graph on augmentations $\mathcal{X}$ with adjacency matrix $A \in \mathbb{R}^{\mathcal{X}\times\mathcal{X}}$ with entries $A[x, x'] = \mathcal{D}_{\text{sim}}(x, x')$, i.e. similar augmentations have edges. The normalized adjacency matrix, a central object in spectral graph theory, is defined as $A_\circ \in \mathbb{R}^{\mathcal{X}\times\mathcal{X}}$ with entries $A_\circ[x, x'] = \frac{\mathcal{D}_{\text{sim}}(x,x')}{\sqrt{\mathcal{D}_{\mathcal{X}}(x)\mathcal{D}_{\mathcal{X}}(x')}}$.

Canonical results in spectral graph theory connect the eigenvalues $\lambda_1 \le \cdots \le \lambda_{|\mathcal{X}|}$ of the normalized Laplacian $L_\circ = I - A_\circ$ to density of edges in the graph: denser graphs have larger eigenvalues. For representation dimension d, HaoChen et al. [2021] roughly make two key assumptions: (1) any partition of the graph into $\mathcal{O}(d)$ partitions is dense i.e. λ_{d+1} is high, (2) the partition of downstream classes is sparse. The condition (2) is the same as saying augmentations of different classes do not overlap much. Under these assumptions, they show the following transfer bound:

Theorem 3.4.5 (Theorem 4.2 from HaoChen et al. [2021])**.** *If $\lambda_1 \le \cdots \le \lambda_{|\mathcal{X}|}$ are the eigenvalues of the normalized Laplacian $L_\circ = I - A_\circ$ for the augmentation distribution $\mathcal{A}$, and if the augmentations can predict the original input labels with probability $1 - \alpha$, then for any $d' \in [d]$ and representation f we have*

$$L_{clf}(f; \bar{y}^\star) \lesssim c_1 \frac{\alpha}{\lambda_{d'+1}} + c_2 \frac{(L_{spec}(f) - \inf_{f^\star} L_{spec}(f^\star))\, d'}{(\lambda_{d+1} - \lambda_{d'})^2}$$

where $L_{spec}(f) - \inf_{f^\star} L_{spec}(f^\star)$ is the sub-optimality of f.

Firstly we note that the above bound is function class independent and fits the abstraction from Equation (3.5). If augmentations are disjoint, then augmentations of an image $\bar{x}$ will be connected to each other in the augmentation graph, but disconnected from all other input augmentations. Thus the graph A will have $|\bar{\mathcal{X}}|$ connected components, implying that the first $|\bar{\mathcal{X}}|$ eigenvalues of the Laplacian $L_\circ$ are 0, i.e. $\lambda_i = 0$ for $i \in [|\bar{\mathcal{X}}|]$.[5] So any representation dimension $d < |\bar{\mathcal{X}}|$ leads to vacuous bounds in Theorem 3.4.5. This again happens because the global minimizer of L_{spec} is not unique, and some of those could be terrible on downstream, as in our proof for Lemma 3.4.2.

[5]Results in spectral graph theory equate number of connected components to the multiplicity of eigenvalue 0 of the Laplacian.

Approximately disjoint augmentations. We show that the above bound does not scale well even when there is very little overlap in the augmentation distributions. To quantify approximate disjointness, we consider the problem of predicting the original input $\bar{x}$ that could have generated an augmentation x, as a classification problem.

Definition 3.4.6. *Augmentation distribution $\mathcal{A}$ is $1-\tau$ disjoint when the minimum error achievable in the input identification task, i.e. predicting the input $\bar{x}$ that could generate an augmentation x, is at most τ. Formally this means*

$$\inf_{g:\mathcal{X}\to\bar{\mathcal{X}}} \mathop{\mathbb{E}}_{\bar{x}}\left[\mathop{\mathbb{E}}_{x\sim\mathcal{A}(\cdot|\bar{x})}\left[\mathbb{1}\left\{g(x)\neq\bar{x}\right\}\right]\right] \leq \tau \tag{3.7}$$

The augmentation distributions are disjoint if and only if one can perfectly predict $\bar{x}$ from x, i.e. under the disjoint augmentation setting from Definition 3.4.1, it is easy to see that $\mathcal{A}$ is 1-disjoint. The following result shows that the eigenvalues and eigen-gaps in Theorem 3.4.5 will be small if the augmentation classification accuracy is high.

Lemma 3.4.7. *Suppose again that $|\bar{\mathcal{X}}| = N$ and $\mathcal{D}_{\bar{\mathcal{X}}} = \mathcal{U}(\bar{\mathcal{X}})$. If the augmentations are $1-\tau$ disjoint (defined in Definition 3.4.6), i.e. average accuracy of predicting $\bar{x}$ from x is $1-\tau$, then for $d' \in [d]$,*

$$\lambda_{d+1} - \lambda_{d'} \leq \lambda_{d+1} \leq \frac{2\tau}{(1-{}^{d}/_{N})}$$

Thus for a small representation dimension $d \ll N$, the guarantees from Theorem 3.4.5 are non-vacuous only when $L_{\text{cont}}(f) \leq \inf_{f^\star} L_{\text{cont}}(f^\star) + \mathcal{O}(\tau^2)$, which is a stringent condition to satisfy. The proof of this is presented in Section 3.9.1. Experiments, as in Section 3.3 and later in Section 3.5, suggest that contrastive learning can succeed even without augmentation overlap. Given that prior analysis fail, we now proceed to show function class dependent guarantees that can show tighter bounds.

3.4.3 Function class dependent transfer guarantees

We present guarantees for a representation that incorporates the function class in addition to the contrastive loss and augmentations. Results in this section are for the spectral contrastive loss defined in Equation (3.2). For simplicity we assume that the input and augmentation sets are finite.

We consider a representation class that is linear in fixed features $\phi : \mathcal{X} \to \mathbb{R}^D$, defined as

$$\mathcal{F}_\phi = \left\{ f(\cdot) = W^\top \phi(\cdot) \mid W \in \mathbb{R}^{D \times d} \right\} \tag{3.8}$$

A crucial property of the function class $\mathcal{F}_\phi$ is that it is expressive enough to solve the downstream task on *augmentations* well, even if not sample efficiently. To formalize this, we define the following metrics

Definition 3.4.8 (Expressivity)**.** *For any augmentation representations $h : \mathcal{X} \to \mathbb{R}^d$ on augmentation labels $g : \mathcal{X} \to \{\pm 1\}$, the regression loss is defined as*

$$L_{reg}(h; g) = \inf_{w \in \mathbb{R}^d} \mathop{\mathbb{E}}_{x \sim \mathcal{X}} \left[\left(w^\top h(x) - g(x) \right)^2 \right]$$

Definition 3.4.9 (Inconsistency)**.** *We define inconsistency of a labeling function $g \in \{\pm 1\}^{\mathcal{X}}$ on augmentations w.r.t. ground truth labeling $\bar{y}^\star \in \{\pm 1\}^{\bar{\mathcal{X}}}$ on original inputs as*

$$\Delta_{\mathcal{A}}(g, \bar{y}^\star) = \mathop{\mathbb{E}}_{\bar{x}} \left[\mathop{\mathbb{E}}_{x \sim \mathcal{A}(\cdot|\bar{x})} [\mathbb{1}\{g(x) \neq \bar{y}^\star(\bar{x})\}] \right] \tag{3.9}$$

Denote the augmentation mean features as $\phi_{\mathcal{A}} = \mathop{\mathbb{E}}_{x \sim \mathcal{A}(\bar{x})} [\phi(x)]$ and covariance as $\Sigma(\phi) = \mathop{\mathbb{E}}_{x} \left[\phi(x) \phi(x)^\top \right]$. We now present the upper bound result.

Theorem 3.4.10. *Let $\lambda_1, \cdots, \lambda_D$ be the eigenvalues of $I_D - \Sigma(\phi)^{-\frac{1}{2}} \Sigma(\phi_{\mathcal{A}}) \Sigma(\phi)^{-\frac{1}{2}}$ in increasing order. Then for every $d' \in [d]$, a representation $f \in \mathcal{F}_\phi$ will satisfy*

$$L_{clf}(f; \bar{y}^\star) \leq \frac{\min_{g \in \{\pm 1\}^{\mathcal{X}}} 4 \left(2\Delta_{\mathcal{A}}(g, \bar{y}^\star) + \sqrt{L_{reg}(\phi; g)} \right)}{\lambda_{d'+1}} + \frac{2d'(L_{spec}(f) - \inf_{f^\star \in \mathcal{F}_\phi} L_{spec}(f^\star))}{(1 - \lambda_{d'})(\lambda_{d+1} - \lambda_{d'})^2}.$$

Firstly note that the this transfer bound is indeed of the form $L_{\text{clf}}(f; \bar{y}^\star) \leq \mathcal{T}(\Gamma, L_{\text{spec}}(f), \mathcal{F}_\phi)$ as in Equation (3.6), connecting to the function class sensitivity discussed in Section 3.1. This is because the eigenvalues, and the inconsistency and regression metrics in the above bound depend on the features ϕ that defines the function class, unlike the guarantee from HaoChen et al. [2021] in Theorem 3.4.5, where the eigenvalues depend only on the data distributions. We discuss the result in more detail in Section 3.8.4 and present its proof in Section 3.8. The result can in fact recover Theorem 3.4.5, in the special case of ϕ being full rank,

i.e. $D = |\mathcal{X}|$. In this case we have

- $L_{\text{reg}}(\phi; g) = 0$, since a full rank ϕ can express any function in $\mathbb{R}^{\mathcal{X}}$.
- $\inf_{f^\star \in \mathcal{F}_\phi} L_{\text{spec}}(f^\star) = \inf_{f^\star} L_{\text{spec}}(f^\star)$ since $\mathcal{F}_\phi$ can express all d-dimensional representations.
- $\min_g \Delta_{\mathcal{A}}(g, \bar{y}^\star) = \mathcal{O}(\alpha)$, where α (from Theorem 3.4.5) is the minimum error in predicting labels from augmentations. Setting g to be an optimal augmentations to label predictor and plugging into Definition 3.4.9 proves this.
- Finally, the matrix $I_D - \Sigma(\phi)^{-\frac{1}{2}} \Sigma(\phi_{\mathcal{A}}) \Sigma(\phi)^{-\frac{1}{2}}$ is closely related to the normalized Laplacian from Section 3.4.2. Proof of this is presented in Lemma 3.8.12.

However when ϕ is not full rank, we get a function class dependent bound that can potentially provide non-vacuous guarantees under weaker assumptions, as will be evident in the next part.

Revisiting hypercube setting. We provide theoretical explanations for observations from Section 3.3 by instantiating our lower and upper bounds for the hypercube example.

Corollary 3.4.11. *Consider the setting from Example 3.3.1. Suppose the classifier is $w^\star = e_1 \in \mathbb{R}^k$, so the downstream label is $\bar{y}^\star(\bar{x}) = \bar{x}_1$. Furthermore, let the feature map ϕ be an identity mapping, i.e. $\phi(x) = x$. In this setting, the following statements are true:*

(a) Function class-agnostic transfer guarantees are vacuous.
(b) For any $f \in \mathcal{F}_\phi$, we have $L_{clf}(f; \bar{y}^\star) \leq 32k \left(L_{spec}(f) - \inf_{f^\star \in \mathcal{F}_\phi} L_{spec}(f^\star) \right)$.

Result (b) suggests that finding an approximate minimizer of the contrastive loss, within the class $\mathcal{F}_\phi$, is sufficient to guarantee good downstream performance; this explains the good performance of linear representation in Table 3.1. The proof of this part is presented in Section 3.7.1. Result (a) explains the presence of spurious representation in the same table and also why prior analyses fail on this example, and follows from Corollary 3.4.4.

3.5 Experiments

Our theoretical examples and analysis show that prior function class agnostic transfer bounds can be near vacuous, particularly when augmentation distributions are disjoint (Corollary 3.4.4) or near disjoint (Lemma 3.4.7). Furthermore they suggest that meaningful downstream guarantees for contrastive learning

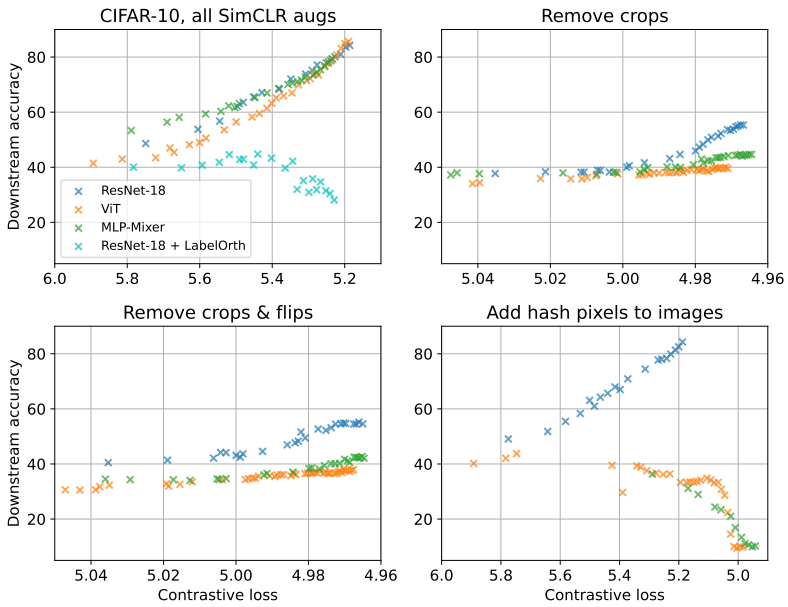


Figure 3.3: Contrastive loss → accuracy transfer plots for CIFAR-10 with `ResNet-18`, `ViT`, and `MLP-Mixer` architectures for different augmentations. **TL:** Full pipeline of augmentations from SimCLR [Chen et al., 2020a]. **TR:** Remove random cropping. **BL:** Remove random cropping and horizontal flip. **BR:** Add "hash pixels" to each image, as described in Section 3.5.1 to ensure that there is no overlap in augmentations. Here, we observe transfer collapse for the `ViT` and `MLP-Mixer` architectures, as they overfit to these uninformative features; `ResNet-18` ignores these pixels.

would need to depend not only on the contrastive loss but also on the representation function class and possibly training algorithm. In this section, we ask, in the context of modern contrastive learning pipelines: (*a*) *how sensitive to the function class is the contrastive loss → downstream accuracy transfer in practice?*, (*b*) *do augmentations sufficiently overlap in standard settings* and (*c*) *can contrastive learning work when there is little to no overlap?*

3.5.1 CIFAR-10 + SimCLR experiments

We consider the setting of CIFAR-10 image classification, where the augmentation distribution for contrastive learning is derived from the popular SimCLR protocol [Chen et al., 2020a]. An augmentation is generated by applying a series of transformations (each with some probability) to an image, like random cropping, horizontal flipping, color jittering, grayscaling and Gaussian blurring (see Section 3.10.2).

We run contrastive learning with standard function classes (architectures): convolutional networks (`ResNet`) [He et al., 2016], Vision Transformers (`ViT`) [Dosovitskiy et al., 2021] and `MLP-Mixer` [Tolstikhin et al., 2021].

Like in the hypercube example, we compare the transfer performance of different function class and algorithmic choices, as contrastive pre-training proceeds, by plotting the trajectories through $(L_{\text{cont}}(f), 1-L_{\text{clf}}(f))$ space of different setups. Figure 3.3 summarizes our findings at a glance and we list the key observations below:

- **Effect of function class.** While standard training using the full pipeline of SimCLR augmentations (top left) displays very similar behaviors for different architectures, removal of certain transformations like random cropping (top right) and horizontal flipping (bottom left), leading to "weaker" augmentations, can accentuate the difference in transfer performances between different architectures.
- **Label-orthogonal training.** All architectures behaving similarly for the full SimCLR pipeline (top left) might superficially suggest that the role of inductive biases is not that significant, and that function class agnostic guarantees are good enough to explain the practical success of contrastive learning for these augmentations. However for the same augmentations, we can find pathological representations that have small contrastive loss but poor downstream performance, by introducing an adversarial modification to the training algorithm and minor tweak to ResNet architecture that is detailed in Section 3.10.2. This suggests that guarantees depending only on the contrastive loss, but not the function class or algorithm, cannot explain the effectiveness of contrastive learning with standard architectures and augmentations.
- **Hash pixels.** The difference in architectures is even more prominent in the hash pixels setting (bottom right). Here we non-destructively[6] modify images and augmentations in order to force the augmentation to be in the disjoint augmentation regime, as defined in Definition 3.4.1. In this case, `ViT` and `MLP-Mixer` representations make the contrastive loss much smaller than `ResNet`, but have close to random guessing downstream performance. `ResNet` training however is unaffected by this hash pixel modification, and it does well on the downstream task, despite being far from minimizing the contrastive loss. This experiment not only highlights the difference in function classes, but also concretely demonstrate a case where *contrastive learning can succeed despite the augmentation distributions being disjoint.* Details on the hash pixel augmentation are in Section 3.10.2.

An important point to note is that the contrastive losses and downstream accuracies in Figure 3.3 are measured on unseen data and are thus reflective of the population versions of these metrics; thus the difference in transfer performance is not an issue of generalization. Further details of experimental setups and hyperparameters are in Section 3.10.2. The next question we tackle is understanding how much overlap there is in augmentations for standard settings.

[6]Only add a small number of pseudorandom pixels in random locations of a 2D image; this kind of noise can be easily removed and do not visually change images by much.

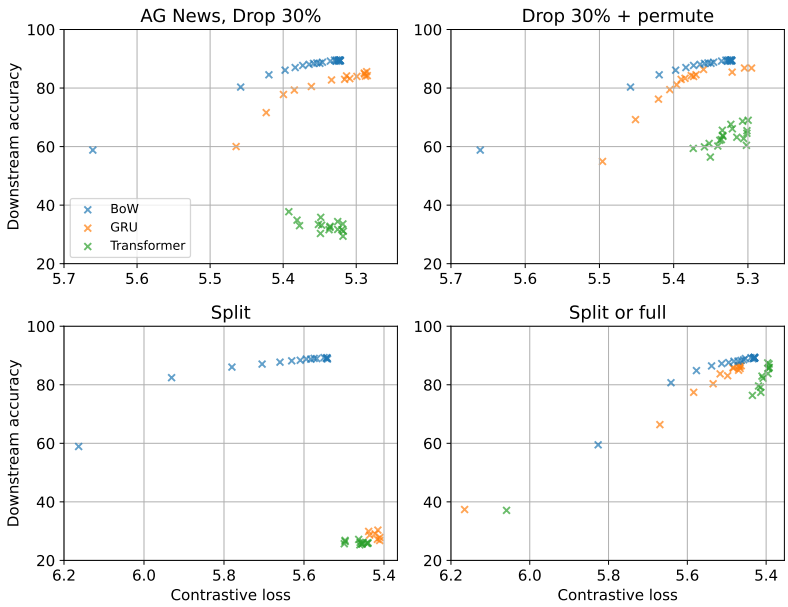


Figure 3.4: Contrastive loss → accuracy transfer plots for AG News with bag-of-words (`BoW`), `GRU` and `Transformer` architectures with representation dimensionality $d = 128$. Augmentations in each case are as follows: **TL:** Drop random 30% of tokens. **TR:** Drop random 30% of tokens and randomly permute the rest. **BL:** Either the first half or second half of the input. **BR:** Either the first half, second half or the full input. In all cases `BoW` representation makes the contrastive loss reasonably small and does quite well downstream ($\sim$ 90%), but either `Transformer` or both `GRU` and `Transformer` demonstrate brittleness of transfer for different augmentations.

3.5.2 Are we in the disjoint augmentation regime?

Central to previous theory is the assumption that there exists overlap between augmentations distributions of data within a class. To get a sense of amount of overlap, we set up a classification task of predicting the image $\bar{x}$ that could have generated an augmentation x, similar to Definition 3.4.6. The standard `ResNet`-18 architecture is modified to have 5000 output classes, one for each image in a CIFAR-10 class. We train the model to take augmentations from a fixed CIFAR-10 class and predict index of the original image generating it. Performance is measured on *unseen* augmented data from inputs from the same class, by evaluating the accuracy of predicting the original input. The results of these experiments in Figure 3.7 (averaged over the 10 classes) suggest that with extremely high accuracy, the trained model is able to identify the image given an augmentation, with accuracies higher than 99.5% for different augmentation types. This suggests that we may be closer to the disjoint augmentation setting than we think. An important note is that the overlap here was only measured on images in the training set, not on the full population distribution as assumed in prior work [HaoChen et al., 2021]. It is still an open question whether sufficient overlap exists in the

population distribution or not. The experiment on the training set is intended as a starting point towards this question.

3.5.3 Experiments on text domain

In order to understand if our findings apply beyond images, we evaluate the contrastive pipeline on text domain. We use the AG News classification dataset [7] [Zhang et al., 2015], where inputs are new articles and the 4 classes correspond to topics of the articles. Inspired by recent augmentations strategies like word/span deletion and word reordering [Wu et al., 2020b, Giorgi et al., 2021, Meng et al., 2021, Yan et al., 2021], we consider four simple augmentations for our study: (i) *Drop*: randomly drop 30% tokens but keep the order of remaining tokens, (ii) *Drop+Permute*: randomly drop 30% tokens and randomly permute the remaining tokens, (iii) *Split*: randomly return either the left half or right half of the text, and (iv) *Split+Full*: randomly return from the full text, its left half, or its right half.

We run contrastive learning with three models on this dataset. The first is a simple Bag-of-Word (`BoW`) model that learns word embeddings and returns the average word embedding of a text, second model is a Gated Recurrent Unit (`GRU`) Chung et al. [2014] and the last is a `Transformer` [Vaswani et al., 2017] model. Both `GRU` and `Transformer` are unidirectional and we pick the final word representation for contrastive learning and downstream linear classification evaluation. All models are trained from scratch to minimize the contrastive loss, without the auxiliary MLM objective as in some prior work [Wu et al., 2020b, Giorgi et al., 2021, Meng et al., 2021]. See Section 3.10.3 for details on the experimental setup and hyperparameters.

Figure 3.4 visualizes the training trajectories through the $(L_{\text{cont}}(f), 1-L_{\text{clf}}(f))$ space, i.e. the contrastive learning $\rightarrow$ downstream accuracy transfer plots. We observe that for all augmentations, `BoW` has the best downstream performance, despite having worse contrastive loss. For the drop augmentation (top left), the `BoW` and `GRU` plots might suggest that the augmentation is good; however the `Transformer` model leads to brittle transfer, i.e. it fails to solve the downstream task despite achieving lower contrastive loss. This kind of difference in transfer performance is unexplained by existing function class agnostic theoretical guarantees. Since the `BoW` representation is order invariant, we also test the augmentation that permutes tokens after dropping 30% of them (top right). This change does help the downstream accuracy of `Transformer`, however it does not completely bridge the gap. While the split augmentation (bottom left) works for `BoW`, both `GRU` and `Transformer` display brittle transfer. However a simple change of including the original text as an augmentation leads to both `GRU` and `Transformer`

[7] We use the PyTorch torchtext library: `https://pytorch.org/text/stable/index.html`

doing well downstream. This is particularly surprising, since including the identity augmentation only decreases the probability of overlap between augmentations, undesirable based on our current understanding of contrastive learning. In Section 3.10.3 we verify that this difference in performance is not just due to distribution shift (augmentations in contrastive learning v/s unaugmented inputs in downstream evaluation).

In Figure 3.10 we visualize two dimensional representations learned using contrastive learning, where it is evident that while the `Transformer` makes the representations invariant to augmentations, representations of augmentations from different classes look very similar to each in distribution and are thus not linear separable. This phenomenon aligns with our lower bound Lemmas 3.4.2 and 3.4.3, whose proofs reveal how such spurious representations can be constructed. The main takeaway is that for various augmentations, a weaker (less expressive) function class can succeed with weaker augmentations, while more expressive ones like `GRU` and `Transformer` might require stronger augmentations to transfer well to downstream tasks. This phenomenon is not well understood by current theory and deserves more exploration.

3.6 Conclusion

Contrastive learning has emerged as a unifying paradigm for building flexible learners that can adapt to many tasks. It is imperative to understand it better at a conceptual and mathematical level. The current work lays out simple experiments and theoretical examples which suggest gaps in our current understanding. Filling these gaps will require incorporating the inductive bias of the deep nets being used, which has primarily been studied in simplistic architectures (e.g., depth 2 or 3) so far. The hypercube example from Section 3.3 and the behavior of simple architectures like MLPs is already an open problem. Incorporating function class bias into transfer bounds is quite non-trivial and our results show how this can be done for linear representations. Extending these results to more complex function classes, and incorporating training procedures could potentially give us new insights. Our study in this work has been diagnostic in nature: identifying gaps in our understanding. Converting these insights into algorithmic approaches is a very promising direction. We also hope that visualizations of contrastive loss $\rightarrow$ downstream performance can aid selection of more robust augmentations.

3.7 Omitted Proofs

3.7.1 Proof of Corollary 3.4.11

The proof of (a) follows directly from Lemma 3.4.2, since all the conditions are satisfied and the augmentations are disjoint.

For the eigenvalues, we can compute the covariances by using $\tau \sim \mathcal{U}((0,1])$

$$\Sigma(\phi) = \operatorname{diag}\left(\mathbf{1}_k, \mathop{\mathbb{E}}_{\tau}[\tau^2]\mathbf{1}_{D-k}\right) = \operatorname{diag}(\mathbf{1}_k, 1/3\mathbf{1}_{D-k})$$
$$\Sigma(\phi_{\mathcal{A}}) = \operatorname{diag}\left(\mathbf{1}_k, (\mathop{\mathbb{E}}_{\tau}[\tau])^2\mathbf{1}_{D-k}\right) = \operatorname{diag}(\mathbf{1}_k, 1/4\mathbf{1}_{D-k})$$

Thus the matrix of interest is

$$I_D - \Sigma(\phi)^{-\frac{1}{2}}\Sigma(\phi_{\mathcal{A}})\Sigma(\phi)^{-\frac{1}{2}} = \operatorname{diag}\left(\mathbf{0}_k, 1/4\mathbf{1}_{D-k}\right)$$

giving us $\lambda_i = 0$ for $i \leq k$ and $\lambda_i = 1/4$ for $i > k$. Plugging into Theorem 3.4.10 for $d' = k$ finishes the proof.

3.8 Proof for linear representation upper bound

Firstly we set up some notation. For sets P and set Q, where P is finite, we denote Q^P to denote the set of all functions from $P \to Q$. We abuse notation and also denote Q^P to be a subset of $Q^{|P|}$, where an element $r \in Q^P$ is a vector of $|P|$ dimensions and coordinates are indexed by elements of P. For instance, when $Q = \{\pm 1\}$ and P is finite, $Q^P = \{\pm 1\}^P$ denotes all functions mapping elements in P to either 1 or -1. Furthermore, $r \in \{\pm 1\}^P$ denotes a vector in $\{\pm 1\}^{|P|}$ that looks like $(r(p))_{p \in P}$. Similarly we denote $Q^{P \times R}$ to denote a matrix in $Q^{|P| \times |R|}$. For a matrix $Q \in \mathbb{R}^{m \times n}$, $Q_{:d} \in \mathbb{R}$

We now prove function class dependent guarantees for the class of linear representations. As in Section 3.4.3, for a feature map $\phi : \mathcal{X} \to \mathbb{R}^D$, we define the linear representation class $\mathcal{F}_\phi = \left\{f(\cdot) = W^\top \phi(\cdot) \mid W \in \mathbb{R}^{D \times D}\right\}$. We wish to show downstream guarantees for contrastive learning that depend not only on the contrastive loss of a representation $f : \mathcal{X} \to \mathbb{R}^d$ but also uses the fact that it belongs to the class $\mathcal{F}_\phi$. In particular, we desire a bound that looks like $L_{\text{clf}}(f; \bar{y}^\star) \leq \mathcal{T}(\mathcal{A}, \bar{y}^\star, L_{\text{cont}}(f), \mathcal{F}_\phi)$, as described in Equation (3.6).

Table 3.2: Notations

Notation	Definition	Description
	Distributions	
$\bar{\mathcal{X}}, \mathcal{X}$		Set of inputs and augmentations
$\mathcal{A}$	$x \sim \mathcal{A}(\cdot \mid \bar{x})$	Augmentation distribution
$\mathcal{D}_{\bar{\mathcal{X}}}$	$\bar{x} \sim \mathcal{D}_{\bar{\mathcal{X}}}$	Marginal distribution on inputs $\bar{\mathcal{X}}$
$\mathcal{D}_{\mathcal{X}}$	$\mathop{\mathbb{E}}_{\bar{x}}[\mathcal{A}(\cdot \mid \bar{x})]$	Marginal distribution on augmentations $\mathcal{X}$
$\bar{D} \in \mathbb{R}^{\bar{\mathcal{X}} \times \bar{\mathcal{X}}}$	$\bar{D}[\bar{x}, \bar{x}] = \mathcal{D}_{\bar{\mathcal{X}}}(\bar{x})$	Matrix of marginal distributions on $\bar{\mathcal{X}}$
$D \in \mathbb{R}^{\mathcal{X} \times \mathcal{X}}$	$D[x, x] = \mathcal{D}_{\mathcal{X}}(x)$	Matrix of marginal distributions on $\mathcal{X}$
$\bar{A} \in \mathbb{R}^{\bar{\mathcal{X}} \times \mathcal{X}}$	$\bar{A}[\bar{x}, x] = \mathcal{D}_{\bar{\mathcal{X}}}(\bar{x})\mathcal{A}(x \mid \bar{x})$	Input augmentation distribution
$\bar{A}_\circ \in \mathbb{R}^{\bar{\mathcal{X}} \times \mathcal{X}}$	$\bar{D}^{-\frac{1}{2}} \bar{A} D^{-\frac{1}{2}}$	Normalized matrix version of $\bar{A}$
$A \in \mathbb{R}^{\mathcal{X} \times \mathcal{X}}$	$A[x, x] = \mathcal{D}_{\text{sim}}(x, x)$	Matrix of joint distribution of augmentations
$A_\circ \in \mathbb{R}^{\mathcal{X} \times \mathcal{X}}$	$D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = \bar{A}_\circ^\top \bar{A}_\circ$	Normalized matrix version of A
	Fixed features	
$\phi : \mathcal{X} \to \mathbb{R}^D$		Fixed feature map for augmentations
$\phi_{\mathcal{A}} : \bar{\mathcal{X}} \to \mathbb{R}^D$	$\mathop{\mathbb{E}}_{x \sim \mathcal{A}(x \mid \cdot)}[\phi(x)]$	Augmentation averaged feature
$\Sigma(\phi) \in \mathbb{R}^{D \times D}$	$\mathop{\mathbb{E}}_{x}\left[\phi(x)\phi(x)^\top\right]$	Covariance of feature map ϕ
$\Phi \in \mathbb{R}^{\mathcal{X} \times D}$	$\Phi[x] = \phi(x)$	Matrix version of feature map ϕ
$\Phi_\circ \in \mathbb{R}^{\mathcal{X} \times D}$	$D^{\frac{1}{2}} \Phi$	Normalized version of Φ
	Representation	
$f : \mathcal{X} \to \mathbb{R}^d$		Representation function
$f_\circ : \mathcal{X} \to \mathbb{R}^d$	$\sqrt{\mathcal{D}_{\mathcal{X}}(\cdot)} f(\cdot)$	Normalized version of f
$F \in \mathbb{R}^{\mathcal{X} \times d}$	$F[x, i] = f(x)_i$	Matrix version of f
$F_\circ \in \mathbb{R}^{\mathcal{X} \times d}$	$D^{\frac{1}{2}} F$	Normalized version of F
	Function classes	
$\mathcal{F} \subseteq \{f : \mathcal{X} \to \mathbb{R}^d\}$		Representation function class
$\mathcal{F}_\phi \subseteq \{f : \mathcal{X} \to \mathbb{R}^d\}$	$\left\{W^\top \phi(\cdot) \mid W \in \mathbb{R}^{D \times d}\right\}$	Linear representation class
$\mathcal{F}_\Phi \subseteq \mathbb{R}^{\mathcal{X} \times d}$	$\left\{\Phi W \mid W \in \mathbb{R}^{D \times d}\right\}$	Linear representation class (matrix version)

We employ the strategy from HaoChen et al. [2021] and show guarantees for the spectral contrastive loss, defined in Equation (3.2) as

$$L_{\text{spec}}(f) = -2 \mathop{\mathbb{E}}_{(x,x^+)\sim\mathcal{D}_{\text{sim}}} \left[f(x)^\top f(x^+)\right] + \mathop{\mathbb{E}}_{x,x^-\sim\mathcal{D}_{\text{neg}}^2} \left[\left(f(x)^\top f(x^-)\right)^2\right] \tag{3.10}$$

We first provide a sketch of their proof in our notation and highlight the main steps. Our result is similar in spirit to theirs, but deviates at crucial junctions due to incorporation of the function class.

1. Rewrite as matrix factorization. Lemma 3.2 from HaoChen et al. [2021] shows that this objective can be rewritten as matrix factorization. For any two augmentations $x, x' \in \mathcal{X}$, define $w_{x,x'} = \mathcal{D}_{\text{sim}}(x, x') = \mathop{\mathbb{E}}_{\bar{x}}[\mathcal{A}(x \mid \bar{x})\mathcal{A}(x \mid \bar{x})]$ to be the probability that x and x' appear as a similar pair, i.e. two augmentations of the same input. Let $w_x = \sum_{x'\in\mathcal{X}} w_{x,x'} = \mathcal{D}_{\text{neg}}(x)$ be the marginal probability. Then the objective can be rewritten as follows:

$$\begin{aligned}
L_{\text{spec}}(f) &= \mathop{\mathbb{E}}_{(x,x^+)\sim\mathcal{D}_{\text{sim}}} \left[f(x)^\top f(x^+)\right] + \mathop{\mathbb{E}}_{x,x^-\sim\mathcal{D}_{\text{neg}}^2} \left[\left(f(x)^\top f(x^-)\right)^2\right] \\
&= -2 \sum_{x,x^+\in\mathcal{X}} w_{x,x^+} f(x)^\top f(x^+) + \sum_{x,x^-\in\mathcal{X}} w_x w_{x^-} \left(f(x)^\top f(x^-)\right)^2 \\
&= \sum_{x,x'\in\mathcal{X}} \left(-2 w_{x,x'} f(x)^\top f(x') + w_x w_{x'} \left(f(x)^\top f(x')\right)^2\right) \\
&= C + \sum_{x,x'\in\mathcal{X}} \left(\frac{w_{x,x'}}{\sqrt{w_x w_{x'}}} - \left(\sqrt{w_x} f(x)\right)^\top \left(\sqrt{w_{x'}} f(x')\right)\right)^2
\end{aligned}$$

where C depends only on w and thus only on $\mathcal{A}$, but not f. Thus $L_{\text{spec}}(f)$ can be interpreted as a matrix factorization objective, with the matrix being $A_\circ \in \mathbb{R}^{\mathcal{X}\times\mathcal{X}}$ such that $A_\circ[x, x'] = \frac{w_{x,x'}}{\sqrt{w_x w_{x'}}}$ and scaled version of representation $u_x = \sqrt{w_x} f(x)$ is being used to factorize this. Note that $A_\circ$ only depends on w's which in turn only depend on the distributions $\mathcal{A}$, $\mathcal{D}_{\bar{\mathcal{X}}}$ and $\mathcal{D}_{\mathcal{X}}$. We stack the representation f into a matrix $F_\circ \in \mathbb{R}^{\mathcal{X}\times d}$, where the column corresponding to $x \in \mathcal{X}$ is $F_\circ[x] = \sqrt{w_x} f(x)$. Then the objective can be written as

$$L_{\text{spec}}(f) = C + \|A_\circ - F_\circ F_\circ^\top\|_F^2. \tag{3.11}$$

This helps characterize the optimal solution $f^\star$ of the contrastive objective, which corresponds to the matrix $F_\circ^\star$ learning the top d eigen-directions of the matrix $A_\circ$. Inspired by this analysis, we also show that the spectral loss with the function class $\mathcal{F}_\phi$ is a matrix factorization problem, but for a different matrix that

depends on both $A_\circ$ and ϕ.

2. ϵ-optimal solution f While the above characterization tells us something about the optimal representation $f^\star$, in general we might have a representation that has sub-optimality of $\epsilon = L_{\text{spec}}(f) - L_{\text{spec}}(f^\star)$. In this case, it can be argued that such a representation captures significant mass of the first d eigen-directions of $A_\circ$ as long as ϵ is small and the eigen-gap is large. More specifically, if $\gamma_1, \ldots, \gamma_{\mathcal{X}}$ denote the eigenvalues of $A_\circ$, then the suboptimal f will capture all except $\mathcal{O}\left(\frac{\epsilon}{(\gamma_{d+1}-\gamma_d)^2}\right) = \mathcal{O}\left(\frac{L_{\text{spec}}(f) - \inf_{f^\star} L_{\text{spec}}(f^\star)}{(\gamma_{d+1}-\gamma_d)^2}\right)$ mass of the first d eigen-directions of $A_\circ$. For our analysis, we will suffer a suboptimality only w.r.t. the function class $\mathcal{F}_\phi$, i.e. the ϵ will be $L_{\text{spec}}(f) - \inf_{f^\star \in \mathcal{F}_\phi} L_{\text{spec}}(f^\star)$ rather than $L_{\text{spec}}(f) - \inf_{f^\star} L_{\text{spec}}(f^\star)$

3. Connecting to downstream. It remains to show why approximately learning the top d directions of the augmentation matrix $A_\circ$ can help with a downstream task $\bar{y}^\star$. This step uses two assumptions, (1) there is sufficient overlap in augmentation distributions overall, and (2) augmentations are approximately label invariant, i.e. there is not much overlap in augmentations of inputs from different classes. These assumptions imply that the true label vector $y^\star \in \{\pm 1\}^{\mathcal{X}}$ has a high component on the first d directions of $A_\circ$. We use similar properties but with less stringent conditions on the amount of overlap between augmentations. In addition to this we need a crucial assumption that the function class $\mathcal{F}_\phi$ is expressive enough to solve the classification task on augmentations.

3.8.1 Matrix notation

Given the backdrop of the results from HaoChen et al. [2021], we now presentation the matrix notations for various functions that will be helpful to prove our main result. All definitions and notations are summarized in Table 3.2.

Distributions to matrices

Let $w_{\bar{x}} = \mathcal{D}_{\bar{\mathcal{X}}}(x)$ denote the marginal probabilities of input $\bar{x} \in \bar{\mathcal{X}}$ and $w_{\bar{x},x} = \mathcal{A}(x \mid \bar{x}) w_{\bar{x}}$ denote the joint probability of input and augmentation. The marginal for augmentations can then be defined as $w_x = \mathcal{D}_{\mathcal{X}}(x) = \sum_{\bar{x}} w_{\bar{x},x}$. To summarize

$$w_{\bar{x}} = \mathcal{D}_{\bar{\mathcal{X}}}(x) \tag{3.12}$$

$$w_{x|\bar{x}} = \mathcal{A}(x \mid \bar{x}) \tag{3.13}$$

$$w_{x,\bar{x}} = w_{\bar{x},x} = \mathcal{A}(x \mid \bar{x}) w_{\bar{x}} = w_{x|\bar{x}} w_{\bar{x}} \tag{3.14}$$

$$w_x = \mathcal{D}_{\mathcal{X}}(x) \tag{3.15}$$

Let $\bar{D} \in \mathbb{R}^{\bar{\mathcal{X}} \times \bar{\mathcal{X}}}$ denote a diagonal matrix of marginal probabilities, i.e. $\bar{D} = \text{diag}((w_{\bar{x}})_{\bar{x} \in \bar{\mathcal{X}}})$. Similarly $D \in \mathbb{R}^{\mathcal{X} \times \mathcal{X}}$ is the diagonal matrix of augmentation marginals. Thus these diagonal matrices satisfy

$$\bar{D}[\bar{x}, \bar{x}] = w_{\bar{x}}, \; D[x, x] = w_x \tag{3.16}$$

We express the augmentation distributions $\mathcal{A}(\cdot \mid \bar{x})_{\bar{x} \in \bar{\mathcal{X}}}$ as a matrix $\bar{A} \in \mathbb{R}^{\bar{\mathcal{X}} \times \mathcal{X}}$, where $\bar{A}[\bar{x}, x] = w_{\bar{x},x}$. A normalized version of $\bar{A}$ is denoted by $\bar{A}_\circ \in \mathbb{R}^{\bar{\mathcal{X}} \times \mathcal{X}}$ and defined as $\bar{A}_\circ[\bar{x}, x] = \frac{w_{\bar{x},x}}{\sqrt{w_{\bar{x}} w_x}}$. We summarize these definitions below, along with a matrix equation that follows easily from the definition

$$\bar{A}[\bar{x}, x] = w_{\bar{x},x}, \; \bar{A}_\circ[\bar{x}, x] = \frac{w_{\bar{x},x}}{\sqrt{w_{\bar{x}} w_x}}, \; \bar{A}_\circ = \bar{D}^{-\frac{1}{2}} \bar{A} D^{-\frac{1}{2}} \tag{3.17}$$

For the similarity distribution $\mathcal{D}_{\text{sim}}$ on pairs of augmentations, define the following

$$w_{x,x'} = \mathcal{D}_{\text{sim}}(x, x') = \mathop{\mathbb{E}}_{\bar{x}}[\mathcal{A}(x \mid \bar{x}) \mathcal{A}(x' \mid \bar{x})] = \sum_{\bar{x}} w_{\bar{x}} w_{x|\bar{x}} w_{x'|\bar{x}}. \tag{3.18}$$

$\mathcal{D}_{\text{sim}}$ is expressed as a matrix $A \in \mathbb{R}^{\mathcal{X} \times \mathcal{X}}$, where $A[x, x'] = w_{x,x'}$. The normalized version of A is defined as $A_\circ \in \mathbb{R}^{\mathcal{X} \times \mathcal{X}}$, where $A_\circ[x, x'] = \frac{w_{x,x'}}{\sqrt{w_x w_{x'}}}$. We summarize these definitions below, along with a matrix equation that follows easily from the definition

$$A[x, x'] = w_{x,x'}, \; A_\circ[x, x'] = \frac{w_{x,x'}}{\sqrt{w_x w_{x'}}}, \; A_\circ = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \tag{3.19}$$

The following lemma connects the $\bar{A}_\circ$ and $A_\circ$

Lemma 3.8.1. *For $\bar{A}_\circ$ and $A_\circ$ defined in Table 3.2, we have the following*

$$A_\circ = \bar{A}_\circ^\top \bar{A}_\circ \tag{3.20}$$

Proof. Firstly from Equation (3.17), we get that $\bar{A}_\circ^\top \bar{A}_\circ = D^{-\frac{1}{2}} \bar{A}^\top \bar{D}^{-1} \bar{A} D^{-\frac{1}{2}}$. Given that $A_\circ = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$

from Equation (3.19), it suffices to show that $A = \bar{A}^\top \bar{D}^{-1} \bar{A}$. The (x, x') entry of the RHS is as follows

$$\left(\bar{A}^\top \bar{D}^{-1} \bar{A}\right)[x, x'] = \sum_{\bar{x}} \frac{w_{\bar{x},x}\ w_{\bar{x},x'}}{w_{\bar{x}}} =^{(a)} \sum_{\bar{x}} \frac{w_{x|\bar{x}} w_{\bar{x}}\ w_{x'|\bar{x}} w_{\bar{x}}}{w_{\bar{x}}} = \sum_{\bar{x}} w_{\bar{x}} w_{x|\bar{x}} w_{x'|\bar{x}} =^{(b)} w_{x,x'} = A[x, x']$$

where (a) follows from Equation (3.14) and (b) follows from Equation (3.18). This completes the proof. □

Representations to matrices

The previous section described how to convert distributions to matrices. We now do the same for representation functions. For a feature map $\phi : \mathcal{X} \to \mathbb{R}^D$, we denote $\Phi \in \mathbb{R}^{\mathcal{X} \times D}$ to be the matrix of representations, with the rows being $\Phi[x] = \phi(x)$. The distributionally normalized version of the representation $\phi_\circ(x) = \sqrt{\mathcal{D}_{\mathcal{X}}(x)}\phi(x) = \sqrt{w_x}\phi(x)$ is denoted by $\Phi_\circ \in \mathbb{R}^{\mathcal{X} \times D}$ with row for $x \in \mathcal{X}$ being $\phi_\circ(x)$. We similarly define the matrices for representation $f : \mathcal{X} \to \mathbb{R}^d$ to be F, $F_\circ$ for the distributionally normalized version. It is easy to see the following relationship between F and $F_\circ$: $F_\circ = D^{\frac{1}{2}} F$. For the function class of linear representations $\mathcal{F}_\phi = \{W^\top \phi(\cdot) \mid W \in \mathbb{R}^{D \times d}\}$, the matrix version is defined as $\mathcal{F}_\Phi = \{\Phi W \mid W \in \mathbb{R}^{D \times d}\}$.

3.8.2 Connecting losses to matrix notations

We first define various downstream evaluation metrics for representation.

Definition 3.8.2. *We define the classification and regression error for any augmentation representation function $h : \mathcal{X} \to \mathbb{R}^d$. For any ground-truth labeling $\bar{y}^\star : \bar{\mathcal{X}} \to \{\pm 1\}$ on original inputs, we define the following*

$$L_{clf}(h; \bar{y}^\star) = \inf_{w \in \mathbb{R}^d} \mathop{\mathbb{E}}_{\bar{x} \sim \bar{\mathcal{X}}} \left[\mathbb{1}\left\{\operatorname{sign}\left(w^\top h_{\mathcal{A}}(\bar{x})\right) = \bar{y}^\star(\bar{x})\right\}\right] \tag{3.21}$$

$$L_{reg}(h; \bar{y}^\star) = \inf_{w \in \mathbb{R}^d} \mathop{\mathbb{E}}_{\bar{x} \sim \bar{\mathcal{X}}} \left[\left(w^\top h_{\mathcal{A}}(\bar{x}) - \bar{y}^\star(\bar{x})\right)^2\right] \tag{3.22}$$

where $h_{\mathcal{A}}(\bar{x}) = \mathop{\mathbb{E}}_{x \sim \mathcal{A}(\cdot|\bar{x})}[h(x)]$ is the augmentation averaged representation (see Table 3.2). For any labeling $g : \mathcal{X} \to \{\pm 1\}$ on augmentations, we define the following

$$L_{reg}(h; g) = \inf_{w \in \mathbb{R}^d} \mathop{\mathbb{E}}_{x \sim \mathcal{X}} \left[\left(w^\top h(\bar{x}) - g(x)\right)^2\right] \tag{3.23}$$

We now connect the downstream regression loss with matrix versions of feature map ϕ.

Lemma 3.8.3. *For an arbitrary predictor on augmentations $g \in \{\pm 1\}^{\mathcal{X}}$ and its normalized version $g_\circ = D^{\frac{1}{2}} g$,*

and an augmentation feature map $\phi : \mathcal{X} \to \mathbb{R}^d$ and its normalized matrix $\Phi_\circ$,

$$L_{reg}(\phi; g) = \|P^{\perp}_{\Phi_\circ} g_\circ\|^2 \tag{3.24}$$

Proof. Note that $\Phi_\circ = D^{\frac{1}{2}}\Phi$, where $\Phi \in \mathbb{R}^{\mathcal{X}\times d}$ is the matrix version of the augmentation feature map ϕ (refer Table 3.2). We prove the result by rewriting L_{reg} as follows

$$L_{\text{reg}}(\phi; g) = \inf_{w\in\mathbb{R}^d} \mathop{\mathbb{E}}_{x} \left(\phi(x)^\top w - g(x)\right)^2 = \inf_{w\in\mathbb{R}^d} \sum_{x\in\mathcal{X}} D(x) \left(\phi(x)^\top w - g(x)\right)^2 \tag{3.25}$$

$$= \inf_{w\in\mathbb{R}^d} \sum_{x} \left(\sqrt{D(x)}\phi(x)^\top w - \sqrt{D(x)}g(x)\right)^2 \tag{3.26}$$

$$= \inf_{w\in\mathbb{R}^d} \left\|D^{\frac{1}{2}}\Phi w - D^{\frac{1}{2}} g\right\|^2 = \inf_{w\in\mathbb{R}^d} \|\Phi_\circ w - g_\circ\|^2 \tag{3.27}$$

$$= \left\|P^{\perp}_{\Phi_\circ} g_\circ\right\|^2 \tag{3.28}$$

□

We now express the spectral contrastive loss and upper bound the downstream classification error using matrix versions of distributions and representations.

Lemma 3.8.4. *For any representation f and its corresponding normalized matrix $F_\circ \in \mathbb{R}^{\mathcal{X}\times d}$, the spectral contrastive loss (Equation* (3.2)*) and classification loss (Equation* (3.3)*) can be rewritten and upper bounded as*

$$L_{spec}(f) = L_{spec}(F_\circ) = \left\|A_\circ - F_\circ F_\circ^\top\right\|_F^2 - \|A_\circ\|_F^2 = \left\|\bar{A}_\circ^\top \bar{A}_\circ - F_\circ F_\circ^\top\right\|_F^2 - \left\|\bar{A}_\circ^\top \bar{A}_\circ\right\|_F^2 \tag{3.29}$$

$$L_{clf}(f; \bar{y}^\star) \le L_{reg}(f; \bar{y}^\star) = \inf_{w\in\mathbb{R}^d} \left\|\bar{A}_\circ F_\circ w - \bar{y}^\star_\circ\right\|_2^2 = \|P^{\perp}_{\bar{A}_\circ F_\circ} \bar{y}^\star_\circ\|_2^2 \tag{3.30}$$

Proof. We first prove the expression for $L_{\text{spec}}(f)$. Note that $A_\circ[x, x'] = w_{x,x'} = \mathcal{D}_{\text{sim}}(x, x')$ from Equation (3.18). Furthermore $F_\circ[x] = \sqrt{\mathcal{D}_{\mathcal{X}}(x)} f(x)$ from Table 3.2. On expanding out the contrastive loss, we get

$$L_{\text{spec}}(f) = \mathop{\mathbb{E}}_{(x,x^+)\sim\mathcal{D}_{\text{sim}}} \left[f(x)^\top f(x^+)\right] + \mathop{\mathbb{E}}_{x,x^-\sim\mathcal{D}^2_{\text{neg}}} \left[\left(f(x)^\top f(x^-)\right)^2\right]$$

$$= -2 \sum_{x,x^+\in\mathcal{X}} w_{x,x^+} f(x)^\top f(x^+) + \sum_{x,x^-\in\mathcal{X}} w_x w_{x^-} \left(f(x)^\top f(x^-)\right)^2$$

$$
\begin{aligned}
&= \sum_{x,x' \in \mathcal{X}} \left(-2w_{x,x'} f(x)^\top f(x') + w_x w_{x'} \left(f(x)^\top f(x') \right)^2 \right) \\
&= -\sum_{x,x' \in \mathcal{X}} \left(\frac{w_{x,x'}}{\sqrt{w_x w_{x'}}} \right)^2 + \sum_{x,x' \in \mathcal{X}} \left(\frac{w_{x,x'}}{\sqrt{w_x w_{x'}}} - \left(\sqrt{w_x} f(x)\right)^\top \left(\sqrt{w_{x'}} f(x')\right) \right)^2 \\
&= -\sum_{x,x'} A_\circ[x,x']^2 + \sum_{x,x'} \left(A_\circ[x,x'] - F_\circ[x]^\top F_\circ[x'] \right)^2 \\
&= -\|A_\circ\|_F^2 + \|A_\circ - F_\circ F_\circ^\top\|_F^2
\end{aligned}
$$

Now we prove the upper bound of $L_{\text{clf}}(f; \bar{y}^\star)$. Firstly note that for any input representation $h : \bar{\mathcal{X}} \to \{\pm 1\}$, we have that

$$
L_{\text{clf}}(h; \bar{y}^\star) = \inf_{w \in \mathbb{R}^d} \mathbb{E}_{\bar{x}} \left[\mathbb{1}\left\{ \bar{y}^\star(\bar{x}) \left(h(\bar{x})^\top w \right) < 0 \right\} \right] \leq^{(a)} \inf_{w \in \mathbb{R}^d} \mathbb{E}_{\bar{x}} \left[\left(\bar{y}^\star(\bar{x}) - h(\bar{x})^\top w \right)^2 \right] = L_{\text{reg}}(h; \bar{y}^\star)
$$

where (a) from the fact that whenever $\bar{y}^\star(\bar{x}) \left(h(\bar{x})^\top w \right) < 0$, $h(\bar{x})^\top w$ has different sign compared to $\bar{y}^\star \in \{\pm 1\}$, and so $(h(\bar{x})^\top w - \bar{y}^\star)^2 \geq \bar{y}^{\star 2} = 1$. Thus for an augmentation representation $f : \mathcal{X}$, we have

$$
\begin{aligned}
L_{\text{reg}}(f; \bar{y}^\star) = L_{\text{reg}}(f_{\mathcal{A}}; \bar{y}^\star) &= \inf_{w \in \mathbb{R}^d} \mathbb{E}_{\bar{x}} \left[\left(f_{\mathcal{A}}(\bar{x})^\top w - \bar{y}^\star(\bar{x}) \right)^2 \right] = \inf_{w \in \mathbb{R}^d} \sum_{\bar{x}} \left[\left(\sqrt{w_{\bar{x}}} f_{\mathcal{A}}(\bar{x})^\top w - \sqrt{w_{\bar{x}}} \bar{y}^\star(\bar{x}) \right)^2 \right] \\
&= \inf_{w \in \mathbb{R}^d} \sum_{\bar{x}} \left[\left(\sqrt{w_{\bar{x}}} f_{\mathcal{A}}(\bar{x})^\top w - \bar{y}_\circ^\star(\bar{x}) \right)^2 \right]
\end{aligned}
$$

We first observe the following about $f_{\mathcal{A}}$:

$$
\begin{aligned}
\sqrt{w_{\bar{x}}} f_{\mathcal{A}}(\bar{x}) = \sqrt{w_{\bar{x}}} \mathop{\mathbb{E}}_{x \sim \mathcal{A}(\cdot|\bar{x})} [f(x)] &= \sqrt{w_{\bar{x}}} \sum_{x \in \mathcal{X}} \mathcal{A}(x \mid \bar{x}) f(x) = \sum_{x \in \mathcal{X}} \sqrt{w_{\bar{x}}} \frac{w_{\bar{x},x}}{w_{\bar{x}}} f(x) = \sum_{x \in \mathcal{X}} \frac{w_{\bar{x},x}}{\sqrt{w_{\bar{x}} w_x}} \sqrt{w_x} f(x) \\
&= \sum_x A_\circ[\bar{x}, x] F_\circ[x] = (A_\circ F_\circ)[\bar{x}]
\end{aligned}
$$

Plugging this back into the previous calculation, we get

$$
L_{\text{reg}}(f; \bar{y}^\star) = \inf_{w \in \mathbb{R}^d} \sum_{\bar{x}} \left[\left((A_\circ F_\circ)[\bar{x}]^\top w - \bar{y}_\circ^\star[\bar{x}] \right)^2 \right] = \inf_{w \in \mathbb{R}^d} \|A_\circ F_\circ w - \bar{y}_\circ^\star\|_F^2
$$

The final step follows from the standard expression for error of linear regression, which is the norm of the component of $\bar{y}_\circ^\star$ on the null space of $A_\circ F_\circ$, i.e. $\|P^\perp_{A_\circ F_\circ}\|_F^2$. □

We now show a more specialized form of matrix factorization objective that results from the representation

belonging to a particular linear function class.

Lemma 3.8.5. *For any representation $f \in \mathcal{F}_\phi$ and its normalized matrix $F_\circ \in \mathbb{R}^{\mathcal{X}\times d}$, the spectral contrastive loss (Equation (3.2)) can be rewritten as*

$$L_{spec}(f) = L_{spec}(F_\circ) = \left\|P_{\Phi_\circ} A_\circ P_{\Phi_\circ} - F_\circ F_\circ^\top\right\|_F^2 + C = \left\|P_{\Phi_\circ} \bar{A}_\circ^\top \bar{A}_\circ P_{\Phi_\circ} - F_\circ F_\circ^\top\right\|_F^2 + C \tag{3.31}$$

where C is a constant independent of f but dependent on features ϕ. Here $\Phi_\circ$ is the normalized matrix for the features ϕ and $P_{\Phi_\circ} \in \mathbb{R}^{\mathcal{X}\times\mathcal{X}}$ is the column projection matrix of $\Phi_\circ$.

Proof. From Lemma 3.8.4, we know that $L_{\text{spec}}(f)$ can be written as a matrix factorization objective as

$$L_{\text{spec}}(f) = \left\|A_\circ - F_\circ F_\circ^\top\right\|_F^2 - \|A_\circ\|_F^2$$

Since f is from the class $\mathcal{F}_\phi$, the matrix form F belongs to the class $\mathcal{F}_\Phi = \left\{\Phi W \mid W \in \mathbb{R}^{D\times d}\right\}$ (refer to Table 3.2). Thus $F_\circ = D^{\frac{1}{2}}F$ can be written as $F_\circ = D^{\frac{1}{2}}F = D^{\frac{1}{2}}\Phi W = \Phi_\circ W$ for some $W \in \mathbb{R}^{D\times d}$. We can conclude that $P_{\Phi_\circ} F_\circ = F_\circ$ and $P_{\Phi_\circ}^\perp F_\circ = 0$ and further simplify the contrastive loss as

$$\begin{aligned}
L_{\text{spec}}(f) &=^{(a)} \left\|P_{\Phi_\circ} A_\circ P_{\Phi_\circ} + P_{\Phi_\circ} A_\circ P_{\Phi_\circ}^\perp + P_{\Phi_\circ}^\perp A_\circ P_{\Phi_\circ} + P_{\Phi_\circ}^\perp A_\circ P_{\Phi_\circ}^\perp - F_\circ F_\circ^\top\right\|_F^2 - \|A_\circ\|_F^2 \\
&=^{(b)} \left\|P_{\Phi_\circ} A_\circ P_{\Phi_\circ} - F_\circ F_\circ^\top\right\|_F^2 + \left\|P_{\Phi_\circ} A_\circ P_{\Phi_\circ}^\perp + P_{\Phi_\circ}^\perp A_\circ P_{\Phi_\circ}\right\|_F^2 + \left\|P_{\Phi_\circ}^\perp A_\circ P_{\Phi_\circ}^\perp\right\|_F^2 - \|A_\circ\|_F^2 \\
&=^{(c)} \left\|P_{\Phi_\circ} A_\circ P_{\Phi_\circ} - F_\circ F_\circ^\top\right\|_F^2 + C
\end{aligned}$$

where (a) follows by decomposing $A_\circ = (P_{\Phi_\circ} + P_{\Phi_\circ}^\perp)A_\circ(P_{\Phi_\circ} + P_{\Phi_\circ}^\perp)$, (b) follows because cross terms cancel through $P_{\Phi_\circ} P_{\Phi_\circ}^\perp$ multiplications, and (c) because all other terms are independent of $F_\circ$ (and so f). This completes the proof. □

We now restate the definition of Inconsistency from Section 3.4.3 and then relate it to some matrix form.

Definition 3.8.6 (Inconsistency)**.** *We define inconsistency of a labeling function $g \in \{\pm 1\}^{\mathcal{X}}$ on augmentations w.r.t. some ground truth labeling $\bar{y}^\star \in \{\pm 1\}^{\bar{\mathcal{X}}}$ on original inputs, as followed:*

$$\Delta_{\mathcal{A}}(g, \bar{y}^\star) = \mathop{\mathbb{E}}_{\bar{x}}\left[\mathop{\mathbb{E}}_{x\sim\mathcal{A}(\cdot|\bar{x})}\left[\mathbb{1}\{g(x) \neq \bar{y}^\star(\bar{x})\}\right]\right] \tag{3.32}$$

Lemma 3.8.7. *For the normalized matrix $\bar{A}_\circ \in \mathbb{R}^{\bar{\mathcal{X}} \times \mathcal{X}}$ corresponding to augmentation distribution $\mathcal{A}$ (refer Table 3.2), ground-truth labeling $\bar{y}^\star \in \{\pm 1\}^{\bar{\mathcal{X}}}$ on original inputs and its normalized version $\bar{y}^\star_\circ = D^{\frac{1}{2}} \bar{y}^\star$, and an arbitrary predictor $g \in \{\pm 1\}^{\mathcal{X}}$ on augmentations and its normalized version $g_\circ = D^{\frac{1}{2}} g$, we have*

$$\bar{y}^{\star\top}_\circ \bar{A}_\circ g_\circ = 1 - 2\Delta_{\mathcal{A}}(g, \bar{y}^\star) \tag{3.33}$$

Proof. Since $\bar{y}^\star_\circ = \bar{D}^{\frac{1}{2}} \bar{y}^\star$, $\bar{A}_\circ = \bar{D}^{-\frac{1}{2}} \bar{A} D^{-\frac{1}{2}}$ and $g_\circ = D^{\frac{1}{2}} g$, the left hand size is equivalent to $\bar{y}^{\star\top} \bar{A} g$. Expanding this further we get

$$\bar{y}^{\star\top} \bar{A} g = \sum_{\bar{x}, x} \bar{A}[\bar{x}, x] \bar{y}^\star(\bar{x}) g(x) =^{(a)} \sum_{\bar{x}, x} w_{\bar{x},x} \bar{y}^\star(\bar{x}) g(x) =^{(b)} \sum_{\bar{x}, x} w_{\bar{x},x} \left(1 - 2\mathbb{1}\{\bar{y}^\star(\bar{x}) \neq g(x)\}\right) \tag{3.34}$$

$$=^{(c)} \sum_{\bar{x} \in \bar{\mathcal{X}}} w_{\bar{x}} \sum_{x \in \mathcal{X}} w_{x|\bar{x}} \left(1 - 2\mathbb{1}\{\bar{y}^\star(\bar{x}) \neq g(x)\}\right) \tag{3.35}$$

$$= 1 - 2 \mathop{\mathbb{E}}_{\bar{x}} \mathop{\mathbb{E}}_{x \sim \mathcal{A}(\cdot|\bar{x})} [\mathbb{1}\{\bar{y}^\star(\bar{x}) \neq g(x)\}] = 1 - 2\Delta_{\mathcal{A}}(g, \bar{y}^\star) \tag{3.36}$$

where (a) follows from Equation (3.19), (b) follows from Equation (3.14), and (c) follows from $\bar{y}^\star(\bar{x}), g(x) \in \{\pm 1\}$. □

3.8.3 Proof of main result

We first state the key lemmas that will used to prove the main result.

The following lemma says that if there is a predictor on augmentations that is consistent with $\bar{y}^\star$ and also expressible enough by fixed features ϕ, then most of $\bar{y}^\star$ is retained by multiplication by $P_{\Phi_\circ} \bar{A}_\circ$.

Lemma 3.8.8. *For the normalized matrix $\bar{A}_\circ \in \mathbb{R}^{\bar{\mathcal{X}} \times \mathcal{X}}$ corresponding to augmentation distribution $\mathcal{A}$, an augmentation feature map ϕ and corresponding normalized matrix $\Phi_\circ$ (refer Table 3.2), ground-truth labeling $\bar{y}^\star \in \{\pm 1\}^{\bar{\mathcal{X}}}$ on original inputs and its normalized version $\bar{y}^\star_\circ = D^{\frac{1}{2}} \bar{y}^\star$, and an arbitrary predictor $g \in \{\pm 1\}^{\mathcal{X}}$ on augmentations and its normalized version $g_\circ = D^{\frac{1}{2}} g$, we have*

$$\left\| P_{\Phi_\circ} \bar{A}^\top_\circ \bar{y}^\star_\circ \right\| \geq 1 - 2\Delta_{\mathcal{A}}(g, \bar{y}^\star) - \sqrt{L_{reg}(\phi; g)} \tag{3.37}$$

Proof. We will use Lemma 3.8.7 to prove this result. Note that $\|\bar{y}^\star_\circ\| = \|g_\circ\| = 1$. First we lower bound

$\|P_{\Phi_\circ}\bar{A}_\circ^\top \bar{y}_\circ^\star\|$ by computing $\bar{y}_\circ^{\star\top}\bar{A}_\circ P_{\Phi_\circ} g_\circ$

$$\begin{aligned}
\|P_{\Phi_\circ}\bar{A}_\circ^\top \bar{y}_\circ^\star\| &\geq^{(a)} \frac{\bar{y}_\circ^{\star\top}\bar{A}_\circ P_{\Phi_\circ} g_\circ}{\|g_\circ\|} = \bar{y}_\circ^{\star\top}\bar{A}_\circ P_{\Phi_\circ} g_\circ \\
&= \bar{y}_\circ^{\star\top}\bar{A}_\circ g_\circ - \bar{y}_\circ^{\star\top}\bar{A}_\circ P_{\Phi_\circ}^\perp g_\circ \\
&=^{(b)} 1 - 2\Delta_{\mathcal{A}}(g, \bar{y}^\star) - \bar{y}_\circ^{\star\top}\bar{A}_\circ P_{\Phi_\circ}^\perp g_\circ \\
&\geq^{(c)} 1 - 2\Delta_{\mathcal{A}}(g, \bar{y}^\star) - \|\bar{y}_\circ^\star\|\|\bar{A}_\circ\|_2\|P_{\Phi_\circ}^\perp g_\circ\| \\
&\geq^{(d)} 1 - 2\Delta_{\mathcal{A}}(g, \bar{y}^\star) - \|P_{\Phi_\circ}^\perp g_\circ\| \\
&\geq^{(e)} 1 - 2\Delta_{\mathcal{A}}(g, \bar{y}^\star) - \sqrt{L_{\mathrm{reg}}(\phi; g)}
\end{aligned}$$

where (a) and (c) follow from Cauchy-Schwarz inequality, (b) follows from Lemma 3.8.7, (d) follows from the fact that $\|\bar{A}_\circ\|_2 = 1$ and (e) follows from Lemma 3.8.3 □

The next lemma quantifies how much of the top singular directions of $P_{\Phi_\circ}\bar{A}_\circ$ are be captured by an ϵ-optimal representation f (or its matrix version $F_\circ$). This is related to Lemma D.10 from HaoChen et al. [2021], however it differs in the fact that we are decomposing $P_{\Phi_\circ}\bar{A}_\circ$ instead of $\bar{A}_\circ$, and we have a better dependence on d on the right hand side. Furthermore, the sub-optimality term is w.r.t. the best representation in the class $\mathcal{F}_\phi$ rather than the unconstrained optimizer of L_{spec}.

Lemma 3.8.9. *Let $f \in \mathcal{F}_\phi \in \mathcal{F}_\phi$ be an augmentation representation function. Suppose $F_\circ \in \mathbb{R}^{\mathcal{X}\times d}$ is the normalized representation matrix corresponding to f, $\bar{A}_\circ$ is the normalized matrix corresponding to augmentation distribution $\mathcal{A}$ and $\Phi_\circ$ is normalized version of Φ (refer Table 3.2). Let $P_{\Phi_\circ}\bar{A}_\circ^\top = USV^\top$ be the singular value decomposition, with $\sqrt{\gamma_1}, \ldots, \sqrt{\gamma_D}$ being the singular values in decreasing order. Then for $d' \leq d$,*

$$\left\|P_{F_\circ}^\perp U_{:d'}\right\|_F^2 \leq \frac{L_{spec}(f) - \inf_{f^\star \in \mathcal{F}_\phi} L_{spec}(f^\star)}{\gamma_{d'}^2 - \gamma_{d+1}^2} \leq \frac{L_{spec}(f) - \inf_{f^\star \in \mathcal{F}_\phi} L_{spec}(f^\star)}{\left(\gamma_{d'} - \gamma_{d+1}\right)^2} \tag{3.38}$$

where $U_{:d'} \in \mathbb{R}^{\mathcal{X}\times d'}$ corresponds to the first d' columns (and thus singular vectors) of $P_{\Phi_\circ}\bar{A}_\circ$.

Proof. We first note, using Lemma 3.8.5, that the contrastive loss can be written as the following matrix

factorization objective

$$L_{\text{spec}}(f) = \|P_{\Phi_\circ} A_\circ P_{\Phi_\circ} - F_\circ F_\circ^\top\|_F^2 = \|P_{\Phi_\circ} \bar{A}_\circ^\top \bar{A}_\circ P_{\Phi_\circ} - F_\circ F_\circ^\top\|_F^2 = \|US^2V^\top - F_\circ F_\circ^\top\|_F^2$$

It is is easy to see that $\gamma_i \leq 1$ for every i, since $\max_i \gamma_i = \|P_{\Phi_\circ} \bar{A}_\circ\|_2^2 \leq \|\bar{A}_\circ\|_2^2 \leq 1$. Thus we can invoke Lemma D.10 from HaoChen et al. [2021], but for matrix $P_{\Phi_\circ} A_\circ P_{\Phi_\circ}$ instead of $A_\circ$, to argue that

$$\|P_{F_\circ}^\perp u_i\|_F^2 \leq \frac{\epsilon}{(\gamma_i^2 - \gamma_{d+1}^2)} \leq \frac{\epsilon}{(\gamma_i - \gamma_{d+1})^2}$$

where ϵ is the suboptimality $L_{\text{spec}}(f) - \inf_{f^\star} L_{\text{spec}}(f^\star) = L_{\text{spec}}(f) - \inf_{f^\star \in \mathcal{F}_\phi} L_{\text{spec}}(f^\star)$, since the optimal decomposition for $P_{\Phi_\circ} A_\circ P_{\Phi_\circ}$ must lie in the span of $\Phi_\circ$, and thus $f^* \in \mathcal{F}_\phi$. Adding these for $i \in [d']$ we get

$$\|P_{F_\circ}^\perp U_{:d'}\|_F^2 \leq \sum_{i=1}^{d'} \frac{\epsilon}{(\gamma_i - \gamma_{d+1})^2} \leq \frac{\epsilon d'}{(\gamma_{d'} - \gamma_{d+1})^2}$$

This completes the proof. □

We now show conditions under which the top d' directions of the matrix being factorized captures significant mass of the ground-truth labels.

Lemma 3.8.10. *Let $\bar{A}_\circ$ be the normalized matrix corresponding to augmentation distribution $\mathcal{A}$ and $\Phi_\circ$ be the normalized version of Φ (refer Table 3.2). Let $P_{\Phi_\circ} \bar{A}_\circ^\top = USV^\top$ be the singular value decomposition, with $\sqrt{\gamma_1}, \ldots, \sqrt{\gamma_D}$ being the singular values in decreasing order. Then we have,*

$$\left\|V_{d':}^\top \bar{y}_\circ^\star\right\|^2 \leq \frac{1 - \left\|P_{\Phi_\circ} \bar{A}_\circ^\top \bar{y}_\circ^\star\right\|^2}{1 - \gamma_{d'+1}} \tag{3.39}$$

where $V_{d':} \in \mathbb{R}^{\mathcal{X} \times (|\mathcal{X}| - d')}$ corresponds to the last $|\mathcal{X}| - d'$ columns (and thus singular vectors) of $P_{\Phi_\circ} \bar{A}_\circ$.

Proof. We expand out the term $1 - \left\|P_{\Phi_\circ} \bar{A}_\circ^\top \bar{y}_\circ^\star\right\|^2$ as follows

$$1 - \left\|P_{\Phi_\circ} \bar{A}_\circ^\top \bar{y}_\circ^\star\right\|^2 = 1 - \left\|USV^\top \bar{y}_\circ^\star\right\|^2 = 1 - \left\|SV^\top \bar{y}_\circ^\star\right\|^2 = 1 - \sum_{i=1}^{D} \gamma_i (v_i^\top \bar{y}_\circ^\star)^2$$

$$=^{(a)} \|\bar{y}_\circ^\star\|_2^2 - \sum_{i=1}^{D} \gamma_i (v_i^\top \bar{y}_\circ^\star)^2$$

$$\geq^{(b)} \sum_{i=1}^{D} (v_i^\top \bar{y}_\circ^\star)^2 - \sum_{i=1}^{D} \gamma_i (v_i^\top \bar{y}_\circ^\star)^2 = \sum_{i=1}^{D} (1-\gamma_i)(v_i^\top \bar{y}_\circ^\star)^2$$
$$\geq^{(c)} (1-\gamma_{d'+1}) \sum_{i=d'+1}^{D} (v_i^\top \bar{y}_\circ^\star)^2 = (1-\gamma_{d'+1}) \|V_{d':}\bar{y}_\circ^\star\|^2$$

where (a) follows because $\|\bar{y}_\circ^\star\|^2 = \sum_{\bar{x}} \mathcal{D}_{\bar{\mathcal{X}}}(\bar{x})\bar{y}^\star(\bar{x}) = 1$, (b) follows because $\{v_i\}_{i=1}^D$ form a partial orthonormal basis and (c) is true because $\gamma_i \leq 1$ for every i and since γ_i's are in decreasing order. Rearranging terms completes the proof $\square$

The following lemma connects the eigenvalues of augmentation averaged features that shows up in the final bound, to the eigenvalues of the matrix being decomposed in the spectral contrastive loss.

Lemma 3.8.11. *Let $\Sigma(\cdot)$ be the covariance operator for features and $\phi_{\mathcal{A}}$ denote augmentation averaged representation obtained from ϕ (see Table 3.2). Let $\lambda_1, \cdots, \lambda_D$ be the eigenvalues of $I_D - \Sigma(\phi)^{-\frac{1}{2}}\Sigma(\phi_{\mathcal{A}})\Sigma(\phi)^{-\frac{1}{2}}$ in increasing order. Let $\bar{A}_\circ$ be the normalized matrix corresponding to augmentation distribution $\mathcal{A}$ and $\Phi_\circ$ be the normalized version of ϕ (refer Table 3.2). Let $P_{\Phi_\circ}\bar{A}_\circ^\top = USV^\top$ be the singular value decomposition, with $\sqrt{\gamma_1}, \ldots, \sqrt{\gamma_D}$ being the singular values in decreasing order. Then we have,*

$$\lambda_i = 1 - \gamma_i, \forall\ i \in [D] \tag{3.40}$$

Proof. Let $w_x, w_{\bar{x}}, w_{x,\bar{x}}, w_{x|\bar{x}}$ be as defined in Equations (3.12) to (3.15). Using $\Phi_\circ = D^{\frac{1}{2}}\Phi$ and that D is diagonal with $D[x,x] = w_x$, we first simplify $\Sigma(\phi)$ as follows

$$\Sigma(\phi) = \mathop{\mathbb{E}}_{x\sim\mathcal{D}_{\mathcal{X}}} \left[\phi(x)\phi(x)^\top\right] = \sum_{x\in\mathcal{X}} \left[w_x(x)\phi(x)\phi(x)^\top\right] = \Phi^\top D\Phi = \Phi_\circ^\top\Phi_\circ$$

Next we find the matrix version of $\phi_{\mathcal{A}}$ using $\bar{A}[\bar{x}, x] = w_{\bar{x},x}$ and the following sequence of equalities.

$$\phi_{\mathcal{A}}(\bar{x}) = \mathop{\mathbb{E}}_{x\sim\mathcal{A}(\cdot|\bar{x})}[\phi(x)] = \sum_{x\in\mathcal{X}} w_{x|\bar{x}}\phi(x) = \frac{1}{w_{\bar{x}}}\sum_{x\in\mathcal{X}} w_{x,\bar{x}}\phi(x) = \frac{1}{w_{\bar{x}}}\sum_x \bar{A}[\bar{x},x]\Phi[x]$$

Thus the matrix form of $\phi_{\mathcal{A}}$ is $\Phi_{\mathcal{A}} = \bar{D}^{-1}\bar{A}\Phi$. Similar to the argument for $\Sigma(\phi)$, we can then write $\Sigma(\phi_{\mathcal{A}})$ as follows

$$\Sigma(\phi_{\mathcal{A}}) = \Phi_{\mathcal{A}}^\top\bar{D}\Phi_{\mathcal{A}} = (\bar{D}^{-1}\bar{A}\Phi)^\top\bar{D}(\bar{D}^{-1}\bar{A}\Phi) = \Phi^\top\bar{A}^\top\bar{D}^{-1}\bar{A}\Phi$$

$$= \Phi^\top D^{\frac{1}{2}} \left(D^{-\frac{1}{2}} \bar{A}^\top \bar{D}^{-\frac{1}{2}}\right) \left(\bar{D}^{-\frac{1}{2}} \bar{A} D^{-\frac{1}{2}}\right) D^{\frac{1}{2}} \Phi$$
$$=^{(a)} \Phi_\circ^\top \bar{A}_\circ^\top \bar{A}_\circ \Phi_\circ$$

where (a) follows from Equation (3.17). Let $MNR^\top = \Phi_\circ$ be the SVD, so $P_{\Phi_\circ} = MM^\top$ and $\Phi_\circ^\top \Phi_\circ = R^\top N^2 R$. Using this, we simplify the matrix $\Sigma(\phi)^{-\frac{1}{2}} \Sigma(\phi_{\mathcal{A}}) \Sigma(\phi)^{-\frac{1}{2}}$.

$$\begin{aligned}
\Sigma(\phi)^{-\frac{1}{2}} \Sigma(\phi_{\mathcal{A}}) \Sigma(\phi)^{-\frac{1}{2}} &= \left(\Phi_\circ^\top \Phi_\circ\right)^{-\frac{1}{2}} \Phi_\circ^\top \bar{A}_\circ^\top \bar{A}_\circ \Phi_\circ \left(\Phi_\circ^\top \Phi_\circ\right) \\
&= (RN^2R^\top)^{-\frac{1}{2}} \left(RNM^\top\right) \bar{A}_\circ^\top \bar{A}_\circ \left(MNR^\top\right) (RN^2R^\top)^{-\frac{1}{2}} \\
&= (RN^{-1}R^\top) \left(RNM^\top\right) \bar{A}_\circ^\top \bar{A}_\circ \left(MNR^\top\right) (RN^{-1}R^\top) \\
&= RM^\top \bar{A}_\circ^\top \bar{A}_\circ MR^\top
\end{aligned}$$

Since $\{\lambda_i\}_{i=1}^D$ are the eigenvalues of $I_D - \Sigma(\phi)^{-\frac{1}{2}} \Sigma(\phi_{\mathcal{A}}) \Sigma(\phi)^{-\frac{1}{2}}$, $\{1-\lambda_i\}_{i=1}^D$ are the eigenvalues of $\Sigma(\phi)^{-\frac{1}{2}} \Sigma(\phi_{\mathcal{A}}) \Sigma(\phi)^{-\frac{1}{2}} = RM^\top \bar{A}_\circ^\top \bar{A}_\circ MR^\top$. Thus $\{\sqrt{1-\lambda_i}\}_{i=1}^D$ are the singular values of $RM^\top \bar{A}_\circ^\top$ and thus $M^\top \bar{A}_\circ^\top$ and thus $MM^\top \bar{A}_\circ^\top = P_{\Phi_\circ} \bar{A}_\circ^\top$. The previous statements are true because multiplication by an orthogonal matrix does not change the singular values. Thus $\sqrt{\gamma_i} = \sqrt{1-\lambda_i}$, finishing the proof. □

Lemma 3.8.12. *Let $\Sigma(\cdot)$ be the covariance operator for features and $\phi_{\mathcal{A}}$ denote augmentation averaged representation obtained from ϕ (see Table 3.2). Let $L_\circ = I - A_\circ$ be the Laplacian of the augmentation graph. If the features ϕ are full rank, then the eigenvalues of $I_D - \Sigma(\phi)^{-\frac{1}{2}} \Sigma(\phi_{\mathcal{A}}) \Sigma(\phi)^{-\frac{1}{2}}$ are the same as the eigenvalues of $L_\circ$.*

Proof. Note from Lemma 3.8.1 that the normalized adjacency matrix can be rewritten as $A_\circ = \bar{A}_\circ^\top \bar{A}_\circ$. Also from Lemma 3.8.11, we can imply that the eigenvalues of $I - P_{\Phi_\circ} \bar{A}_\circ^\top \bar{A}_\circ P_{\Phi_\circ}$ are the same as the eigenvalues of $I_D - \Sigma(\phi)^{-\frac{1}{2}} \Sigma(\phi_{\mathcal{A}}) \Sigma(\phi)^{-\frac{1}{2}}$. Since ϕ is full rank, so is the matrix $\Phi_\circ$, thus $P_{\Phi_\circ} = I$. So $I - P_{\Phi_\circ} \bar{A}_\circ^\top \bar{A}_\circ P_{\Phi_\circ} = I - \bar{A}_\circ^\top \bar{A}_\circ = I - A_\circ = L_\circ$; this completes the proof. □

We are now ready to present our main result.

Theorem 3.4.10. *Let $\Sigma(\cdot)$ be the covariance operator for features and $\phi_{\mathcal{A}}$ denote augmentation averaged representation obtained from ϕ (see Table 3.2). Let $\lambda_1, \cdots, \lambda_d$ be the eigenvalues of $I_d - \Sigma(\phi)^{-\frac{1}{2}} \Sigma(\phi_{\mathcal{A}}) \Sigma(\phi)^{-\frac{1}{2}}$*

in increasing order, then for $d \leq D$, any representation $f \in \mathcal{F}_\phi$, will satisfy

$$L_{clf}(f; \bar{y}^\star) \leq \min_{1 \leq d' \leq d} \left\{ \frac{\min_{g \in \{\pm 1\}^{\mathcal{X}}} 4 \left(2\Delta_{\mathcal{A}}(g, \bar{y}^\star) + \sqrt{L_{reg}(\phi; g)} \right)}{\lambda_{d'+1}} + \frac{2d' \left(L_{spec}(f) - \inf_{f^\star \in \mathcal{F}_\phi} L_{spec}(f^\star) \right)}{(1 - \lambda_{d'})(\lambda_{d+1} - \lambda_{d'})^2} \right\} \quad (3.41)$$

where $\Delta_{\mathcal{A}}$ is defined in Definition 3.4.9 and L_{reg} in Definition 3.8.2.

Proof. We first sketch an outline of the proof and how different lemmas will be used to prove the final result. We will use matrix versions of distributions and functions from Table 3.2 throughout the proof. The following are the main steps:

1. (Matrix factorization) The spectral contrastive loss is shown to be equivalent to a matrix factorization objective as in HaoChen et al. [2021]. For representations in the class $\mathcal{F}_\phi$, Lemma 3.8.5 shows that the problem of contrastive learning is reduced to matrix factorization of a projected adjacency matrix $P_{\Phi_\circ} A_\circ P_{\Phi_\circ}$ through the objective $L_{\text{spec}}(F) = \left\| P_{\Phi_\circ} \bar{A}_\circ^\top \bar{A}_\circ P_{\Phi_\circ} - F_\circ F_\circ^\top \right\|_F^2 + C$, where $A_\circ$ is the normalized matrix corresponding to augmentation distribution $\mathcal{A}$ and $F_\circ$ is the normalized matrix for representation function f (refer Table 3.2). Thus the spectral contrastive loss L_{spec} is attempting to find a rank d approximation for $P_{\Phi_\circ} \bar{A}_\circ^\top$.

2. (ϵ-optimal solutions) If $P_{\Phi_\circ} \bar{A}_\circ^\top = USV^\top$ is the singular value decomposition, then any ϵ-optimal representation f (and corresponding $F_\circ$), i.e. $L_{\text{spec}}(f) - \inf_{f^\star \in \mathcal{F}_\phi} L_{\text{spec}}(f^\star) \leq \epsilon$, can be shown (Lemma 3.8.9) to capture most of the signal for the top d' directions of $P_{\Phi_\circ} \bar{A}_\circ^\top$, i.e. $\left\| P_{F_\circ}^\perp U_{:d'} \right\|_F = \mathcal{O}(\epsilon)$ is small.

3. (Connecting to downstream) Top d' directions of $P_{\Phi_\circ} \bar{A}_\circ^\top$ can be shown to capture a lot of the mass of $\bar{y}^\star$ if the features ϕ and augmentation distribution $\mathcal{A}$ are "nice" enough, as quantified by Lemma 3.8.10 that upper bounds $\|V_{d':} \, \bar{y}_\circ^\star\|^2$, in conjunction with Lemma 3.8.8 which quantifies these nice properties of $\mathcal{A}$ and features ϕ.

4. (Wrapping up) Both of the above steps will have upper bounds that depend on the singular values of $P_{\Phi_\circ} \bar{A}_\circ$. Relating the singular values of $P_{\Phi_\circ} \bar{A}_\circ$ with the eigenvalues of $\left(I_d - \Sigma(\phi)^{-\frac{1}{2}} \Sigma(\phi_{\mathcal{A}}) \Sigma(\phi)^{-\frac{1}{2}} \right)$ through Lemma 3.8.11 completes the proof.

Step 1: We can rewrite the contrastive loss in matrix forms using Lemma 3.8.4.

$$L_{\text{spec}}(f) = L_{\text{spec}}(F_\circ) = \left\|A_\circ - F_\circ F_\circ^\top\right\|_F^2 - \|A_\circ\|_F^2 = \left\|\bar{A}_\circ^\top \bar{A}_\circ - F_\circ F_\circ^\top\right\|_F^2 - \left\|\bar{A}_\circ^\top \bar{A}_\circ\right\|_F^2 \tag{3.42}$$

For representation $F \in \mathcal{F}_\phi$, we can write it as $F = \Phi W$, thus giving $F_\circ = D^{\frac{1}{2}} F = D^{\frac{1}{2}} \Phi W = \Phi_\circ W$. Note that $P_{\Phi_\circ} = \Phi_\circ \Phi_\circ^\dagger$ is the projection matrix of column space of $\Phi_\circ$; then we have $F_\circ = P_{\Phi_\circ} F_\circ$. From Lemma 3.8.5, we know that $L_{\text{spec}}(F) = \left\|P_{\Phi_\circ} \bar{A}_\circ^\top \bar{A}_\circ P_{\Phi_\circ} - F_\circ F_\circ^\top\right\|_F^2 + C$.

Thus from this we see that the contrastive learning is aiming to learn a good rank d decomposition of the matrix $P_{\Phi_\circ} A_\circ P_{\Phi_\circ} = P_{\Phi_\circ} \bar{A}_\circ^\top \bar{A}_\circ P_{\Phi_\circ}$. This is a similar to the formulation in HaoChen et al. [2021] where the matrix $A_\circ$ is being factorized instead. The classical result on low-rank approximation of matrices tells us that the minimizer $F_\circ$ will span the top d singular of $P_{\Phi_\circ} \bar{A}_\circ$.

Step 2: Let $P_{\Phi_\circ} \bar{A}_\circ = USV^\top$ be the singular value decomposition, with $S = \text{diag}(\sqrt{\gamma_1}, \ldots, \sqrt{\gamma_d})$ being the singular values in decreasing order. Then we know that the optimal solution $F_\circ^\star$ be will $U_{:d} S_{:d} R$ for any orthogonal matrix R. Note that $F_\circ^\star \in \mathcal{F}_{\Phi_\circ}$ since the matrix $P_{\Phi_\circ} \bar{A}_\circ$ being decomposed is in the span of $\Phi_\circ$. This argument can be extended to ϵ-optimal representation f (or matrix F) by invoking Lemma 3.8.9, which gives us that

$$\left\|P_{F_\circ}^\perp U_{:d'}\right\|_F^2 \leq \frac{\epsilon d'}{\gamma_{d'}^2 - \gamma_{d+1}^2} \leq \frac{\epsilon d'}{(\gamma_{d'} - \gamma_{d+1})^2} \tag{3.43}$$

This tells us that being close to optimality ensures that the representation captures most of the top d' singular directions of U and thus $P_{\Phi_\circ} \bar{A}_\circ$, whenever $d' \leq d$. Note that the gap $\gamma_{d'} - \gamma_{d+1}$ in singular values determines how the suboptimality affects the magnitude of "signal" captured.

Step 3: Given that ϵ-optimal solutions can capture the top directions of $P_{\Phi_\circ} \bar{A}_\circ$ (or U), we now focus our attention on what this means for downstream performance. We invoke Lemma 3.8.4 again to upper bound the downstream classification error L_{clf} (refer Definition 3.8.2) as

$$L_{\text{clf}}(f; \bar{y}^\star) \leq L_{\text{reg}}(f; \bar{y}^\star) \leq L_{\text{reg}}(F_\circ) = \inf_{w \in \mathbb{R}^d} \left\|\bar{A}_\circ F_\circ w - \bar{y}_\circ^\star\right\|_2^2 \tag{3.44}$$

$$\leq^{(a)} \inf_{w \in \mathbb{R}^d} \left\|\bar{A}_\circ P_{\Phi_\circ} F_\circ w - \bar{y}_\circ^\star\right\|_2^2 \leq^{(b)} \inf_{w \in \mathbb{R}^d} \left\|\bar{A}_\circ P_{\Phi_\circ} P_{F_\circ} w - \bar{y}_\circ^\star\right\|_2^2 \tag{3.45}$$

$$\leq^{(c)} \inf_{w \in \mathbb{R}^d} \left\|VSU^\top P_{F_\circ} w - \bar{y}_\circ^\star\right\|_2^2 \tag{3.46}$$

where (a) follows from the fact that $F_\circ \in \mathcal{F}_{\Phi_\circ}$ and thus $P_{\Phi_\circ} F_\circ = F_\circ$, (b) is true since for any $w \in \mathbb{R}^d$, there exists $w' \in \mathbb{R}^d$ such that $F_\circ w = P_{F_\circ} w'$, and (c) uses the singular value decomposition of $P_{\Phi_\circ} \bar{A}_\circ$. Thus the downstream error is upper bounded by a quantity that depends on how much of $\bar{y}^\star_\circ$ is not captured by the columns of $\bar{A}_\circ P_{\Phi_\circ} P_{F_\circ} = VSU^\top P_{F_\circ}$. We show this quantity is small, by arguing that the top d' directions of V captures enough component of $\bar{y}^\star_\circ$ Lemma 3.8.8, and that an ϵ-optimal representation will capture a large enough portion of the top d' directions. Note that for any matrix $B \in \mathbb{R}^{n \times n}$, $B_{:m} \in \mathbb{R}^{m \times n}$ denotes the first m columns of B and $B_{m:} \in \mathbb{R}^{n-m \times n}$ denotes that last m columns of B. The calculation is as follows

$$
\begin{aligned}
L_{\text{clf}}(f; \bar{y}^\star) &\leq \inf_{w \in \mathbb{R}^d} \left\| VSU^\top P_{F_\circ} w - \bar{y}^\star_\circ \right\|_2^2 \\
&= \inf_{w \in \mathbb{R}^d} \left\| VSU^\top P_{F_\circ} w - V_{:d'} V_{:d'}^\top \bar{y}^\star_\circ + V_{d':} V_{d':}^\top \bar{y}^\star_\circ \right\|_2^2 \\
&\leq^{(a)} 2 \left(\inf_{w \in \mathbb{R}^d} \left\| VSU^\top P_{F_\circ} w - V_{:d'} V_{:d'}^\top \bar{y}^\star_\circ \right\|^2 + \left\| V_{d':} V_{d':}^\top \bar{y}^\star_\circ \right\|_2^2 \right) \\
&= 2 \inf_{w \in \mathbb{R}^d} \left\| V_{:d'} S_{:d'} U_{:d'}^\top P_{F_\circ} w - V_{:d'} V_{:d'}^\top \bar{y}^\star_\circ \right\|^2 + 2 \left\| V_{d':} V_{d':}^\top \bar{y}^\star_\circ \right\|_2^2 \\
&= 2 \inf_{w \in \mathbb{R}^d} \left\| S_{:d'} U_{:d'}^\top P_{F_\circ} w - V_{:d'}^\top \bar{y}^\star_\circ \right\|^2 + 2 \left\| V_{d':}^\top \bar{y}^\star_\circ \right\|_2^2 \\
&\leq^{(b)} 2 \left\| S_{:d'} U_{:d'}^\top P_{F_\circ} U_{:d'} S_{:d'}^{-1} V_{:d'}^\top \bar{y}^\star_\circ - V_{:d'}^\top \bar{y}^\star_\circ \right\|^2 + 2 \left\| V_{d':}^\top \bar{y}^\star_\circ \right\|_2^2 \\
&= 2 \left\| S_{:d'} U_{:d'}^\top P^\perp_{F_\circ} U_{:d'} S_{:d'}^{-1} V_{:d'}^\top \bar{y}^\star_\circ \right\|^2 + 2 \| V_{d':}^\top \bar{y}^\star_\circ \|_2^2 \\
&\leq^{(c)} 2 \| S_{:d'} \|_2^2 \left\| P^\perp_{F_\circ} U_{:d'} \right\|_2^2 \left\| S_{:d'}^{-1} \right\|_2^2 \| \bar{y}^\star_\circ \|^2 + 2 \left\| V_{d':}^\top \bar{y}^\star_\circ \right\|_2^2 \\
&\leq^{(d)} \frac{2 \left\| P^\perp_{F_\circ} U_{:d'} \right\|_F^2}{\gamma_{d'}} + 2 \left\| V_{d':}^\top \bar{y}^\star_\circ \right\|_2^2
\end{aligned}
$$

where (a) follows from the inequality $\|a+b\|^2 \leq 2(\|a\|^2 + \|b\|^2)$, (b) follows by a picking a specific value $w = U_{:d'} S_{:d'}^{-1} V_{:d'} \bar{y}^\star_\circ$, (c) follows from multiple applications of Cauchy-Schwarz inequality and that $\|V_{:d'}^\top \bar{y}^\star_\circ\| \leq \|\bar{y}^\star_\circ\|$ and (d) follows from $\|S_{:d'}\|_2 \leq 1$ and $\|S_{:d'}^{-1}\|_2 \leq \gamma_{d'}^{-1}$.

The first term is upper bounded in step 2 already by the sub-optimality of f, while the second term is upper bounded using Lemma 3.8.10. Plugging these in, we get

$$
L_{\text{clf}}(f; \bar{y}^\star) \leq \frac{2\epsilon d'}{\gamma_{d'}(\gamma_{d'} - \gamma_{d+1})^2} + \frac{2(1 - \left\| P_{\Phi_\circ} \bar{A}_\circ^\top \bar{y}^\star_\circ \right\|^2)}{1 - \gamma_{d'+1}} \leq \frac{2\epsilon d'}{\gamma_{d'}(\gamma_{d'} - \gamma_{d+1})^2} + \frac{4(1 - \left\| P_{\Phi_\circ} \bar{A}_\circ^\top \bar{y}^\star_\circ \right\|)}{1 - \gamma_{d'+1}}
$$

where for the last inequality we use that $1 - x^2 = (1-x)(1+x) \leq 2(1-x)$ for $x \in [0,1]$. This is further

simplified using Lemma 3.8.8 to

$$L_{\text{clf}}(f;\bar{y}^\star) \leq \frac{2\epsilon d'}{\gamma_{d'}(\gamma_{d'}-\gamma_{d+1})^2} + \frac{4\left(2\Delta_{\mathcal{A}}(g,\bar{y}^\star) + \sqrt{L_{\text{reg}}(\phi;g)}\right)}{1-\gamma_{d'+1}}$$

Step 4: Finally the singular values γ_i are linked the eigenvalues λ_i in the theorem statement through Lemma 3.8.11. Specifically, we have $\gamma_i = 1 - \lambda_i$, giving us the final result

$$L_{\text{clf}}(f;\bar{y}^\star) \leq \frac{4\left(2\Delta_{\mathcal{A}}(g,\bar{y}^\star) + \sqrt{L_{\text{reg}}(\phi;g)}\right)}{\lambda_{d'+1}} + \frac{2\epsilon d'}{(1-\lambda_{d'})(\lambda_{d+1}-\lambda_{d'})^2}$$

where ϵ is the suboptimality $L_{\text{spec}}(f) - \inf_{f^\star \in \mathcal{F}_\phi} L_{\text{spec}}(f^\star)$. The above inequality holds for every $g \in \{\pm 1\}^{\mathcal{X}}$ and for every $d' \in [d]$. Taking a min over both completes the proof. This completes the proof.

□

3.8.4 Discussion of upper bound

We dissect our result from Theorem 3.4.10, and compare it to the result from HaoChen et al. [2021], presented in Theorem 3.4.5. For the representation $f \in \mathcal{F}_\phi$, downstream performance is good if

- $L_{spec}(f) - \inf_{f^\star \in \mathcal{F}_\phi} L_{spec}(f^\star)$ *is small*: The contrastive loss of f is close to the optimal loss in $\mathcal{F}_\phi$, even if best in class is far from the absolute minimizer. The equivalent term in Theorem 3.4.5 was the global sub-optimality of f, i.e. $L_{\text{spec}}(f) - \inf_{f^\star} L_{\text{spec}}(f^\star)$.

- $2\Delta_{\mathcal{A}}(g,\bar{y}^\star) + \sqrt{L_{reg}(\phi;g)}$ *is small*: This happens if there exists a predictor $g \in \{\pm 1\}^{\mathcal{X}}$ on augmentations that is expressible by the features ϕ and is sufficiently consistent with the ground-truth labels $\bar{y}^\star$ on inputs. Note that if augmentation distributions overlap across classes, then $\Delta_{\mathcal{A}}(g,\bar{y}^\star)$ cannot be made small. In fact, $\Delta_{\mathcal{A}}(g,\bar{y}^\star)$ is of the same order as α from Theorem 3.4.5. The extra condition we need here is that $\sqrt{L_{\text{reg}}(\phi;g)}$ is small, i.e. despite ϕ not being full rank, it can roughly express a function that is consistent with ground-truth labels.

- *Eigenvalues* $\lambda_{d'}$ *and eigen-gaps* $\lambda_{d+1} - \lambda_{d'}$ *are not too small*: This is very similar to Theorem 3.4.5, except there the eigenvalues were of the normalized Laplacian (that only depended on distributions), while here the eigenvalues also depend on ϕ and thus the function class. Intuitively these values are large if the

augmentation graph is dense *in the view of the features* ϕ.

3.9 Proofs for lower bounds for (approximately) disjoint augmentations

Here we prove that the global minimizer of the contrastive objective can achieve trivial downstream performance when the augmentation distributions do not overlap.

Theorem 3.9.1. *Let* $N \in \mathbb{N}$ *be given and let* $d \in \mathbb{N}$ *satisfy* $3 \leq d \leq cN/\log_2(N)$ *for a universal constant* $c > 0$. *Let* $\bar{\mathcal{X}}$ *be a set of* $|\bar{\mathcal{X}}| = N$ *instances,* $\bar{y}^\star \in \{\pm 1\}^N$ *be any labeling function with* $\sum_i y_i^\star = 0$, *and let* $\mathcal{D}$ *be the uniform distribution over* $\bar{\mathcal{X}}$. *Suppose that the augmentation distribution* $\mathcal{A}(\cdot \mid \bar{x})$ *is such that* $\forall \bar{x}, \bar{x}' \in \bar{\mathcal{X}} : \mathrm{supp}(\mathcal{A}(\cdot \mid \bar{x})) \cap \mathrm{supp}(\mathcal{A}(\cdot \mid \bar{x}')) = \emptyset$. *Additionally assume either*

- ***Unnormalized case:*** *representations are unconstrained; or*
- ***Normalized case:*** *representations are constrained (to any set) and there is a fixed source of randomness* $W \in \Delta(\mathcal{W})$ *and mapping* $T : \bar{\mathcal{X}} \times \mathcal{W} \to \mathcal{X}$ *that is invertible in* w *for any* $\bar{x}$ *such that* $x \sim \mathcal{A}(\cdot \mid \bar{x}) \equiv w \sim W, x = T(\bar{x}, w)$.

Then for any representation $f^\star : \mathcal{X} \to \mathbb{R}^d$ *there exists a representation* $\hat{f} : \mathcal{X} \to \mathbb{R}^d$ *such that:*

$$L_{cont}(\hat{f}) \leq L_{cont}(f^\star), \quad \text{and} \quad L_{clf}(\hat{f}) = \min_{w \in \mathbb{R}^d} \frac{1}{N} \sum_{i=1}^{n} \mathbf{1}\{\mathrm{sign}(w^\top \hat{f}(\bar{x}_i)) \neq y_i^\star\} \geq \frac{1}{2} - O(\sqrt{d \log(N)/N})$$

Proof. Let us start by considering the unnormalized case. Let $f^\star : \mathcal{X} \to \mathbb{R}^d$ be any representation function. The proof consists of three steps:

1. Show that every instance $\bar{x} \in \bar{\mathcal{X}}$ has an embedding $v_{\bar{x}}$, such that if we embed $\bar{x}$ and all of its augmentations to $v_{\bar{x}}$ then we obtain a new embedding function $\hat{f}$ for which $L_{\mathrm{cont}}(\hat{f})$ is no worse than $L_{\mathrm{cont}}(f^\star)$.
2. Let $\mathcal{V} := \{v_{\bar{x}} : \bar{x} \in \bar{\mathcal{X}}\}$. Show that for any bijection $\pi : \mathcal{V} \to \mathcal{V}$, we have $L_{\mathrm{cont}}(\pi \circ \hat{f}) = L_{\mathrm{cont}}(\hat{f})$. In other words, if we apply a permutation to the embeddings of $\hat{f}$ we do not change the contrastive loss.
3. Show that there exists some permutation π such that $\pi \circ \hat{f}$ has very high downstream error rate.

Part 1. Let us first show that embeds all augmentations of an instance $\bar{x}$ identically only lowers the contrastive loss. By convexity, we have that

$$\begin{aligned} L_{\text{cont}}(f) &:= \mathbb{E}_{\bar{x}_1,\bar{x}_2\sim\mathcal{D}}\mathbb{E}_{(x,x^+)\sim\mathcal{A}(\cdot|\bar{x}_1),x^-\sim\mathcal{A}(\cdot|\bar{x}_2)}\left[-\log\left(\frac{\exp(f(x)^\top f(x^+))}{\exp(f(x)^\top f(x^+))+\exp(f(x)^\top f(x^-))}\right)\right] \\ &\geq \mathbb{E}_{\bar{x}_1,\bar{x}_2\sim\mathcal{D}}\left[-\log\left(\frac{\exp(g(\bar{x}_1)^\top g(\bar{x}_1))}{\exp(g(\bar{x}_1)^\top g(\bar{x}_1))+\exp(g(\bar{x}_1)^\top g(\bar{x}_2))}\right)\right] \end{aligned}$$

where $g(\bar{x}) = \mathbb{E}_{x\sim\mathcal{A}(\cdot|\bar{x})}[f(x)]$ is the mean embedding for f. Note that this inequality is strict if f does not embed all augmentations of an instance identically. If $f^\star$ did not embed augmentations identically, we could replace $f^\star$ with the mean embedding $\hat{f}$ and reduce the contrastive loss. Thus there exists a set $\mathcal{V} := \{v_{\bar{x}} : \bar{x} \in \bar{\mathcal{X}}\}$ such that for each $\bar{x}$, we can assume that $\hat{f}$ embeds $\bar{x}$ and all of its augmentations as $v_{\bar{x}}$.

The same argument also hold for the spectral contrastive loss defined in Equation (3.2), since it is also convex in the inner products.

Part 2. Let us rename the embedding vectors $\{v_{\bar{x}}\}_{\bar{x}\in\bar{\mathcal{X}}}$ to $\mathcal{V} := \{v_i\}_{i=1}^N$. Then we can rewrite the objective as

$$L_{\text{cont}}(\hat{f}) = \frac{1}{N^2}\sum_{i,j} -\log\left(\frac{\exp(v_i^\top v_i)}{\exp(v_i^\top v_i)+\exp(v_i^\top v_j)}\right)$$

Any bijection from $\mathcal{V}$ to $\mathcal{V}$ can be equivalently viewed as a permutation $\pi : [N] \to [N]$. We claim that the above objective is invariant to permuting the indices. This is easy to see, since all pairs (i, j) appear with equal weighting in the above expression. Thus we see that $L_{\text{cont}}(\pi \circ \hat{f}) = L_{\text{cont}}(\hat{f})$.

Part 3. In the last step of the proof, we use a combinatorial argument to show that there exists some permutation π with high error rate. First, note that the embedding function $\pi \circ \hat{f}$ embeds $\bar{x}_i$ and all of its augmentations to $v_{\pi(i)}$. Thus, the downstream loss when using linear function w is

$$\frac{1}{N}\sum_{i=1}^n \mathbf{1}\{\text{sign}(w^\top v_{\pi(i)}) \neq y_i^\star\} = \frac{1}{N}\sum_{i=1}^n \mathbf{1}\{\text{sign}(w^\top v_i) \neq y^\star_{\pi^{-1}(i)}\}$$

So instead of permuting the embeddings $\{v_i\}$, we can equivalently permute the labels $\{y_i^\star\}$. Define:

$$\mathcal{Y} := \{(y^\star_{\pi(i))})_{i=1}^N : \pi \text{ is a permutation } \}$$

$$\mathcal{W} := \left\{(\text{sign}(w^\top v_i))_{i=1}^N : w \in \mathbb{R}^d\right\}$$
$$\mathcal{Z}_\tau := \{x \in \{\pm 1\}^N : \exists b \in \mathcal{W} \text{ s.t. } \frac{1}{N}\sum_{i=1}^N \mathbf{1}\{x_i \neq b_i\} \leq \tau\}.$$

Here $\mathcal{Y}$ are the possible labellings we can generate by permuting the indices (which as we discussed is equivalent to permuting the embedding vectors). $\mathcal{W}$ is the labels we can generate via a linear function of the embeddings. Finally $\mathcal{Z}_\tau$ is the set of labellings that are τ close to the ones that our embeddings can generate. The statement of the theorem is equivalent to $\mathcal{Y} \setminus \mathcal{Z}_\tau \neq \emptyset$ for some large τ, which means that there is some permutation of the labels that is far from all linear functions of our embeddings.

We prove this via a combinatorial argument. First, since $\sum_i y_i^\star = 0$ (meaning that the classes are balanced), we have $|\mathcal{Y}| = \binom{N}{N/2} \geq 2^{N/2}$. On the other hand, by Sauer's lemma,

$$|\mathcal{W}| \leq \sum_{i=0}^d \binom{N}{i}, \quad \text{hence} \quad |\mathcal{Z}_\tau| \leq \sum_{i=0}^d \binom{N}{i} \cdot \sum_{i=0}^{N\tau} \binom{N}{i} \tag{3.47}$$

Let $H(p) := p\log_2(1/p) + (1-p)\log_2(1/(1-p))$ be the binary entropy function, for $p \in [0,1]$. Standard bounds on the volume of Hamming cubes Cover [1999] gives that

$$|\mathcal{Z}_\tau| \leq 2^{H(d/N)\cdot N} 2^{H(\tau)\cdot N}.$$

We also have

$$\binom{N}{N/2} \geq 2^{H(1/2)\cdot N} \cdot \frac{2}{eN} \geq 2^{N - \log_2(eN/2)}$$

Therefore, a sufficient condition is

$$H(d/N) + H(\tau) <= 1 - \frac{\log_2(eN/2)}{N}$$

To proceed, we upper bound the entropy functional on the left hand side using the taylor expansion. For the $H(d/N)$ term we use a first order expansions around $p = 1/N$, which, by concavity, yields an upper bound.

$$H(d/N) \leq H(1/N) + \left.\frac{\partial H(x)}{\partial x}\right|_{x=1/N} (d/N - 1/N)$$

$$= 1/N \log_2(N) + (1 - 1/N) \log_2(N/(N-1)) - \log_2\left(\frac{1/N}{1-1/N}\right) \cdot (d/N - 1/N)$$

$$= \log_2(N/(N-1)) + d/N \cdot \log_2(N-1)$$

$$\leq 2/N + d\log_2(N)/N.$$

The last inequality holds for $N \geq 2$. For $H(\tau)$ we have the upper bound

$$H(\tau) = H(1/2) - \frac{4}{\ln(2)} \cdot \frac{1}{2}(1/2 - \tau)^2 + \frac{H^{(3)}(\xi)}{6}(\tau - 1/2)^3 \leq 1 - \frac{2}{\ln(2)}(1/2 - \tau)^2,$$

Here the first equality is Taylor's remainder theorem where $\xi \in [\tau, 1/2]$ and the second holds because the third derivative is non-negative on the interval $[0, 1/2]$ and we will take $\tau \leq 1/2$. Putting these together, a sufficient condition is

$$\frac{2}{N} + \frac{d\log_2(N)}{N} + 1 - \frac{2}{\ln(2)}(1/2 - \tau)^2 \leq 1 - \frac{\log_2(eN/2)}{N}$$

$$\Leftarrow \tau < \frac{1}{2} - \sqrt{\frac{\ln(2)}{2}\left(\frac{d\log_2(N)}{N} + \frac{2}{N} + \frac{\log_2(eN/2)}{N}\right)}$$

So the error rate is $1/2 - O(\sqrt{d\log(N)/N})$.

For the normalized case, the proof is structurally very similar, except that we cannot rely on the argument in part 1 to show that $f^\star$ embeds $\bar{x}$ and all of its augmentations to the same vector $v_{\bar{x}}$. However, we only use the mean vector $v_{\bar{x}} := \mathbb{E}_{x \sim \mathcal{A}(\cdot|\bar{x})}[f^\star(x)]$ in subsequent steps of the proof and we will see that we can remap embeddings $f^\star(x)$ so that we (a) preserve the NCE loss of $f^\star$ and (b) permute all of the mean vectors $v_{\bar{x}}$.

Let us number the original inputs $\bar{x}_1, \ldots, \bar{x}_N$ and let $\pi : [N] \to [N]$ be any permutation. Let $\mathcal{W}$ be the choices for the random seed and for input $\bar{x}_i$ let $x_{i,w} = T(\bar{x}_i, w)$ be the augmentation obtained when using seed w on input $\bar{x}$. The invertibility of $\mathcal{A}(\bar{x}_i, \cdot)$ implies that $x_{i,w} \neq x_{i,w'}$. This means that we can define a new predictor f_π as

$$f_\pi : x_{i,w} \mapsto f^\star(x_{\pi(i),w}).$$

Since the examples are sampled uniformly at random and since the random seed is independent of the example, we can show that $L_{\text{cont}}(f_\pi) = L_{\text{cont}}(f^\star)$ using a similar argument to the one we used to show permutation invariance in the unnormalized case. At the same time, we have changed the mean embeddings so that $\bar{x}$ is

now embedded as $v_{\bar{x}_{\pi(i)}}$. So now we can continue with part 3 to obtain the result.

□

3.9.1 Approximately disjoint augmentations

Definition 3.9.2. *For an augmentation distribution $\mathcal{A}$, we define Bayes-error($\mathcal{A}$) as the Bayes error of augmentation classification as the minimum error achievable in the input identification task, i.e. predicting the input that could have generated an augmentation. Formally we define it as follows:*

$$\textit{Bayes-error}(\mathcal{A}) = \inf_{g:\mathcal{X}\to\bar{\mathcal{X}}} \mathop{\mathbb{E}}_{\bar{x}} \left[\mathop{\mathbb{E}}_{x\sim\mathcal{A}(\cdot|\bar{x})} \left[\mathbb{1}\left\{ g(x) \neq \bar{x} \right\} \right] \right] \tag{3.48}$$

Lemma 3.9.3. *For an augmentation distribution $\mathcal{A}$, the Bayes error from Definition 3.9.2 has the following expression*

$$\textit{Bayes-error}(\mathcal{A}) = 1 - \mathop{\mathbb{E}}_{x\sim\mathcal{D}} \left[\|\mathcal{A}(\cdot \mid x)\|_\infty \right] \tag{3.49}$$

where $\mathcal{A}(\cdot \mid x)$ is the posterior distribution over original inputs given an augmentation x.

Proof. In the above definition of Bayes error, we pick the optimal predictor g to be $g(x) = \arg\max_{\bar{x}} \mathcal{A}(\bar{x} \mid x)$, which will give us the expression for Bayes error. □

Lemma 3.9.4. *Consider the augmentation distribution $\mathcal{A}$ and its normalized adjacency matrix $A_\circ$, and let $\lambda_1, \ldots, \lambda_{|\mathcal{X}|}$ be the eigenvalues of the normalized Laplacian $I_{|\mathcal{X}|} - A_\circ$ in increasing order. The eigen-gap $\lambda_{d+1} - \lambda_d$ can be upper bounded as follows:*

$$\lambda_{d+1} - \lambda_d \leq \lambda_{d+1} \leq \frac{2\bar{\rho}\ \textit{Bayes-error}(\mathcal{A})}{1 - d/|\bar{\mathcal{X}}|} \tag{3.50}$$

where $\bar{\rho} = \frac{\bar{\mathcal{D}}_{\max}}{\bar{\mathcal{D}}_{\min}}$ is the ratio of max and min probabilities over inputs.

Proof. Let γ_d be the d^{th} largest eigenvalue of the normalized adjacency matrix $A_\circ = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \in \mathbb{R}^{\mathcal{X}\times\mathcal{X}}$, where $A[x, x']$ is the joint probability of augmentations x and x' appearing as two augmentations of the same input. Then we know that the i^{th} smallest eigenvalue of $I_{|\mathcal{X}|} - A_\circ$ is $\lambda_i = 1 - \gamma_i$. Furthermore we note that

$A_\circ \in \mathbb{R}^{\mathcal{X}\times\mathcal{X}}$ has rank at most $|\bar{\mathcal{X}}|$, since from Table 3.2 we know that $A_\circ = \bar{A}_\circ^\top \bar{A}_\circ$, where $\bar{A}_\circ \in \mathbb{R}^{\bar{\mathcal{X}}\times\mathcal{X}}$ is the normalized of the input-augmentation distribution (refer Table 3.2) that has entries $\bar{A}_\circ[\bar{x}, x] = \frac{\mathcal{D}_{\text{sim}}(x,\bar{x})}{\sqrt{\mathcal{D}_{\mathcal{X}}(x)}\sqrt{\mathcal{D}_{\bar{\mathcal{X}}}(\bar{x})}}$. Thus we can conclude that $\gamma_i = 0$ for $|\bar{\mathcal{X}}| < i \leq |\mathcal{X}|$. First we prove the statement $A_\circ = \bar{A}_\circ^\top \bar{A}_\circ$ below

$$
\begin{aligned}
(\bar{A}_\circ^\top \bar{A}_\circ)[x, x'] &= \sum_{\bar{x}} \bar{A}_\circ[\bar{x}, x]\bar{A}_\circ[\bar{x}, x'] = \sum_{\bar{x}} \frac{\mathcal{A}(x,\bar{x})}{\sqrt{\mathcal{D}_{\mathcal{X}}(x)}\sqrt{\mathcal{D}_{\bar{\mathcal{X}}}(\bar{x})}} \frac{\mathcal{A}(x',\bar{x})}{\sqrt{\mathcal{D}_{\mathcal{X}}(x')}\sqrt{\mathcal{D}_{\bar{\mathcal{X}}}(\bar{x})}} \\
&= \frac{1}{\sqrt{\mathcal{D}_{\mathcal{X}}(x)\mathcal{D}_{\mathcal{X}}(x')}} \sum_{\bar{x}} \frac{\mathcal{A}(x,\bar{x})\mathcal{A}(x',\bar{x})}{\mathcal{D}_{\bar{\mathcal{X}}}(\bar{x})} = \frac{1}{\sqrt{\mathcal{D}_{\mathcal{X}}(x)\mathcal{D}_{\mathcal{X}}(x')}} \sum_{\bar{x}} \mathcal{D}_{\bar{\mathcal{X}}}(\bar{x})\mathcal{A}(x \mid \bar{x})\mathcal{A}(x' \mid \bar{x}) \\
&= \frac{A[x, x']}{\sqrt{\mathcal{D}_{\mathcal{X}}(x)\mathcal{D}_{\mathcal{X}}(x')}} = A_\circ[x, x']
\end{aligned}
$$

We now connect Bayes-error($\mathcal{A}$) to the normalized augmentation matrix $A_\circ$ by using Lemma 3.9.3.

$$
\begin{aligned}
(1 - \text{Bayes-error}(\mathcal{A}))^2 &=^{(a)} \left(\mathop{\mathbb{E}}_{x\sim\mathcal{D}} [\|\mathcal{A}(\cdot \mid x)\|_\infty] \right)^2 \leq^{(b)} \left(\mathop{\mathbb{E}}_{x\sim\mathcal{D}} [\|\mathcal{A}(\cdot \mid x)\|_2] \right)^2 \\
&\leq^{(c)} \mathop{\mathbb{E}}_{x\sim\mathcal{D}} \left[\|\mathcal{A}(\cdot \mid x)\|_2^2 \right] = \sum_{x\in\mathcal{X}} \mathcal{D}_{\mathcal{X}}(x) \sum_{\bar{x}\in\bar{\mathcal{X}}} \mathcal{A}(\bar{x} \mid x)^2 \\
&= \sum_{\bar{x}\in\bar{\mathcal{X}}} \mathcal{D}_{\bar{\mathcal{X}}}(\bar{x}) \sum_{x\in\mathcal{X}} \mathcal{A}(x \mid \bar{x})\mathcal{A}(\bar{x} \mid x) \\
&= \sum_{\bar{x}} \mathcal{D}_{\bar{\mathcal{X}}}(\bar{x}) \sum_{x} \frac{\mathcal{A}(x,\bar{x})}{\mathcal{D}_{\bar{\mathcal{X}}}(\bar{x})} \frac{\mathcal{A}(x,\bar{x})}{\mathcal{D}_{\mathcal{X}}(x)} \\
&= \sum_{\bar{x}} \mathcal{D}_{\bar{\mathcal{X}}}(\bar{x}) \sum_{x} \frac{\mathcal{A}(x,\bar{x})}{\sqrt{\mathcal{D}_{\bar{\mathcal{X}}}(\bar{x})\mathcal{D}_{\mathcal{X}}(x)}} \frac{\mathcal{A}(x,\bar{x})}{\sqrt{\mathcal{D}_{\bar{\mathcal{X}}}(\bar{x})\mathcal{D}_{\mathcal{X}}(x)}} \\
&= \sum_{\bar{x}} \mathcal{D}_{\bar{\mathcal{X}}}(\bar{x}) \sum_{x} \bar{A}_\circ[\bar{x}, x]\bar{A}_\circ[\bar{x}, x] = \sum_{\bar{x}} \mathcal{D}_{\bar{\mathcal{X}}}(\bar{x})(\bar{A}_\circ\bar{A}_\circ^\top)[\bar{x}, \bar{x}] \\
&= \text{tr}\left(\bar{D}\bar{A}_\circ\bar{A}_\circ^\top\right)
\end{aligned}
$$

where (a) follows from Lemma 3.9.3, (b) follows from $\|\cdot\|_\infty \leq \|\cdot\|_2$, (c) follows from Jensen's inequality since $h(x) = x^2$ is convex. This upper bound can be used to lower bound the Bayes error as follows:

$$
\begin{aligned}
2\ \text{Bayes-error}(\mathcal{A}) &\geq 1 - (1 - \text{Bayes-error}(\mathcal{A}))^2 \\
&\geq^{(a)} 1 - \text{tr}\left(\bar{D}\bar{A}_\circ\bar{A}_\circ^\top\right) =^{(b)} \text{tr}\left(\bar{D}\right) - \text{tr}\left(\bar{D}\bar{A}_\circ\bar{A}_\circ^\top\right) =^{(c)} \text{tr}\left(\bar{D}(I_{|\bar{\mathcal{X}}|} - \bar{A}_\circ\bar{A}_\circ^\top)\right) \\
&\geq^{(d)} \|\bar{D}^{-1}\|_2^{-1}\ \text{tr}\left(I_{|\bar{\mathcal{X}}|} - \bar{A}_\circ\bar{A}_\circ^\top\right) = \bar{\mathcal{D}}_{\min}\ \text{tr}\left(I_{|\bar{\mathcal{X}}|} - \bar{A}_\circ\bar{A}_\circ^\top\right) \qquad (3.51)
\end{aligned}
$$

where $\bar{\mathcal{D}}_{\min} = \min_{\bar{x}\in\bar{\mathcal{X}}} \mathcal{D}_{\bar{\mathcal{X}}}(x)$. In the above sequence, (a) follows the preceeding calculation, (b) follows from $\text{tr}(\bar{D}) = \sum_{\bar{x}} \bar{D}(\bar{x}) = 1$ and (c) follows from linearity of the trace operator. The penultimate step (d) follows

from the fact that $\text{tr}(XY) \leq \|X\|_2 \text{ tr}(Y)$ for symmetric psd matrices X, Y; a proof for this can be found in Lemma 18 from Jin et al. [2017]. We now connect this quantity to the eigenvalues of $A_\circ$ as follows:

$$\begin{aligned}
\text{tr}\left(I_{|\bar{\mathcal{X}}|} - \bar{A}_\circ \bar{A}_\circ^\top\right) &= |\bar{\mathcal{X}}| - \text{tr}\left(\bar{A}_\circ \bar{A}_\circ^\top\right) =^{(a)} |\bar{\mathcal{X}}| - \text{tr}\left(\bar{A}_\circ^\top \bar{A}_\circ\right) = |\bar{\mathcal{X}}| - \text{tr}\left(A_\circ\right) \\
&=^{(b)} |\bar{\mathcal{X}}| - \sum_{i=1}^{|\mathcal{X}|} \gamma_i =^{(c)} |\bar{\mathcal{X}}| - \sum_{i=1}^{|\bar{\mathcal{X}}|} \gamma_i = |\bar{\mathcal{X}}| - \sum_{i=1}^{|\bar{\mathcal{X}}|} (1 - \lambda_i) = \sum_{i=1}^{|\bar{\mathcal{X}}|} \lambda_i \\
&\geq \sum_{i=d+1}^{|\bar{\mathcal{X}}|} \lambda_i \geq (|\bar{\mathcal{X}}| - d)\lambda_{d+1}
\end{aligned} \tag{3.52}$$

where (a) follows from $\text{tr}(PQ) = \text{tr}(QP)$, (b) is true because γ_i's are the eigenvalues of $A_\circ$ and because trace of a symmetric matrix is the sum of its eigenvalues, (c) follows because $A_\circ$ is rank $|\bar{\mathcal{X}}|$ and so $\gamma_i = 0$ for $i > |\bar{\mathcal{X}}|$. Combining Equations (3.51) and (3.52), we get Bayes-error$(\mathcal{A}) \geq 1/2 \bar{\mathcal{D}}_{\min}(|\bar{\mathcal{X}}| - d)\lambda_{d+1}$. Note that

$$\bar{\mathcal{D}}_{\min} = \frac{\bar{\mathcal{D}}_{\min}}{\bar{\mathcal{D}}_{\max}} \bar{\mathcal{D}}_{\max} \geq \frac{\bar{\mathcal{D}}_{\min}}{\bar{\mathcal{D}}_{\max}} |\bar{\mathcal{X}}|^{-1} = \bar{\rho}^{-1} |\bar{\mathcal{X}}|^{-1} \tag{3.53}$$

Plugging this into the bound gives Bayes-error$(\mathcal{A}) \geq \frac{1}{2\bar{\rho}} \left(1 - \frac{d}{|\bar{\mathcal{X}}|}\right) \lambda_{d+1}$, giving us

$$\lambda_{d+1} \leq \frac{2\bar{\rho} \text{ Bayes-error}(\mathcal{A})}{1 - d/|\bar{\mathcal{X}}|}$$

□

3.10 Experiment details

In this section, we provide additional notes, tables, and figures on the experiments.

3.10.1 Synthetic experiments: hypercube example

Figure Figure 3.5 shows the results from Section 3.3 in greater detail. This section completes the details omitted in the main paper.

Data and augmentations. As outlined in Section 3.3, the data are drawn uniformly from the hypercube in dimension $D = 50$. The downstream labels are determined by a randomly drawn linear classifier w, whose first $k = 10$ coefficients are drawn from $\mathcal{N}(0, 1)$; the rest are 0. The training set (under which L_{cont} is minimized) is of size 50000; the downstream accuracies under a linear classifier are evaluated on a holdout

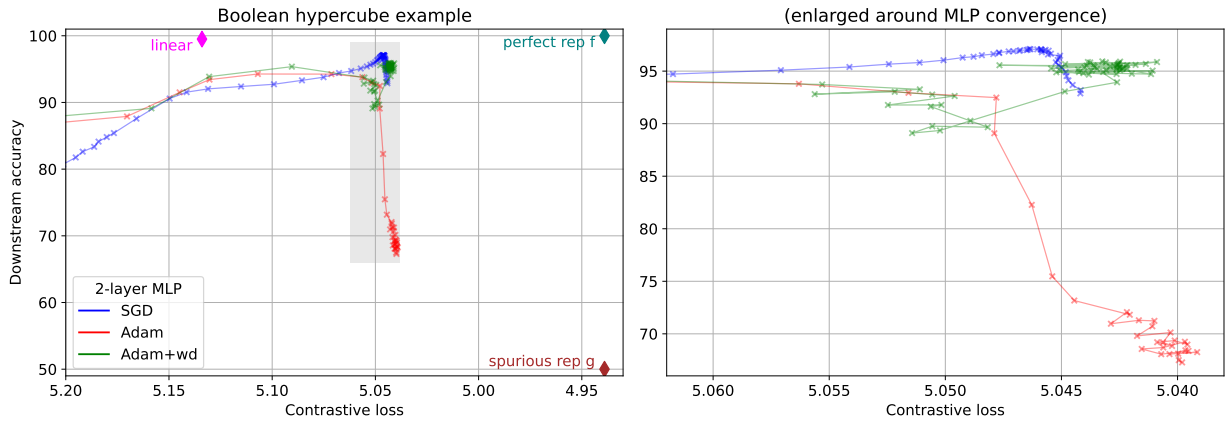


Figure 3.5: Full plots for the synthetic experiments, with all contrastive loss minimizers shown from various function classes (left) and enlarged plot near convergence of trajectories of solutions found by training 2-layed MLPs with various configurations of first-order optimizers (right).

validation set of 12500. The augmentations are selected by i.i.d. random scaling factors $\tau \sim \text{Unif}([0, 1])$ and scaling down the last 40 coordinates.

Training and evaluation. The two-layer MLP models used a hidden layer width of $2D = 100$, and an output (i.e. representation) dimension of 20. Adam was run with a learning rate of 10^{-3}, and default parameters $\beta_1 = 0.9, \beta_2 = 0.99$. The weight decay parameter for Adam was 0.004, selected from $\{0.001, 0.002, \ldots, 0.007\}$ based on best transfer performance. SGD was run with learning rate 0.01. Quantitative results are shown in Table Table 3.1; means and 95% confidence intervals are computed from 10 random seeds. 500 epochs of pre-training were run, with batch size 512.

3.10.2 CIFAR-10 + SimCLR experiments

For all `ResNet` experiments, we use the `ResNet-18` architecture from PyTorch, with the standard modification for CIFAR-10 of replacing the first 7×7 convolution layer with a 3×3 convolution and removing the maxpool layer. We use the `ViT` implementation from `https://github.com/lucidrains/vit-pytorch` with patch size: 4, hidden dimension: 256, depth: 6 and number of heads: 8. For `MLP-Mixer` we use the implementation from `https://github.com/lucidrains/mlp-mixer-pytorch` with patch size: 4, hidden dimension: 256 and number of heads: 8. In each model, the representation for contrastive learning is computed by adding an extra MLP (projection layer) on top of the base model, as proposed in Chen et al. [2020a]. The projection layer has 1 hidden layer with 2048 dimensions, followed by a batch norm layer and ReLU non-linearity, and output dimensionality of 1024.

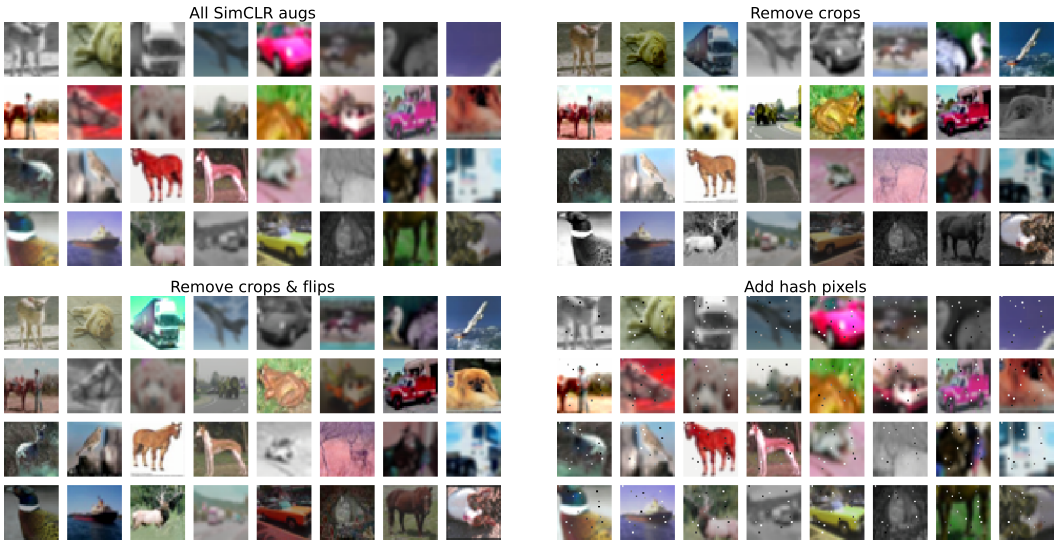


Figure 3.6: Examples of augmented images from CIFAR-10 used in the SimCLR experiments. **TL:** Full pipeline of augmentations from SimCLR [Chen et al., 2020a]. **TR:** Remove random cropping. **BL:** Remove random cropping and horizontal flip. **BR:** Add "hash pixels" to each image, which uniquely identify the particular example.

Augmentations. The following augmentations are used, inspired by [Chen et al., 2020a]:

```
transforms.Compose([

    RandomResizedCrop(32, scale=(0.3, 1.0)),
    RandomHorizontalFlip(p=0.5),
    transforms.RandomApply([transforms.ColorJitter(0.4, 0.4, 0.4, 0.1)], p=0.8),
    RandomGrayscale(p=0.2),
    GaussianBlur(kernel_size=3)

])
```

For experiments in Figure 3.3 we use the full pipeline of augmentations (top left) and sequentially remove random cropping (top right) and horizontal flipping (bottom left). Examples of augmented CIFAR-10 images are shown in Figure Figure 3.6

Contrastive training. We train the model for 1000 epochs, by performing a pass over this training dataset and minimize the SimCLR contrastive learning loss. We normalize the representations f to unit norm when computing the SimCLR loss, as is common in many works. Formally, given a batch $\{(x_i, x_i')\}_{i=1}^{B}$ of pairs of

Table 3.3: Hyperparameter values for experiments on CIFAR-10 trained using `ResNet-18`.

Hyperparameters	Values	
Contrastive training	Max epoch	1000
	Learning rate	0.001
	Optimizer	Adam + weight decay (0.0005)
	Batch size	512
	Representation dimension	1024
Downstream training	Epochs	1000
	Learning rate (start)	0.01
	Optimizer	Adam + weight decay (0.000005)
	Scheduler	ExponentialLR (gamma: $10^{0.004}$)
	Batch size	1000

augmentations we perform a single update of Adam to minimize the following loss:

$$L(f) = -\frac{1}{2B}\sum_{i=1}^{n}\frac{w(x_i, x_i')}{\sum_{j=1}^{n} w(x_i, x_j') + \sum_{j=1, j\neq i}^{n} w(x_i, x_j)} - \frac{1}{2B}\sum_{i=1}^{n}\frac{w(x_i, x_i')}{\sum_{j=1}^{n} w(x_i', x_j) + \sum_{j=1, j\neq i}^{n} w(x_i', x_j')}, \tag{3.54}$$

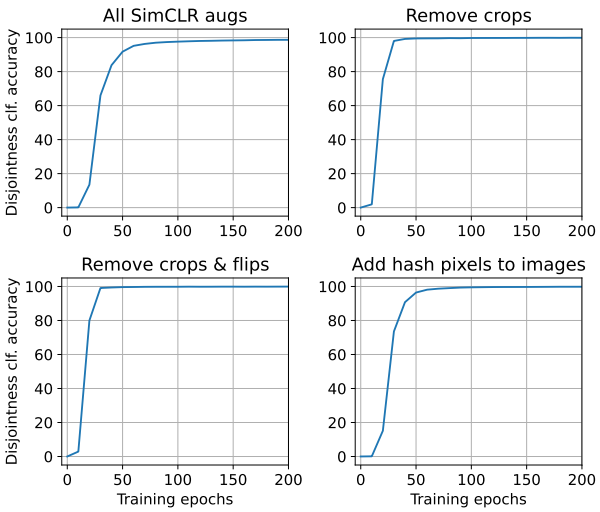
where $w(x, x') = \exp\left(\frac{f(x)^\top f(x')}{\tau \|f(x)\|_2 \|f(x')\|_2}\right)$ and we pick the temperature parameter as $\tau = 0.5$. Training hyperparameters are presented in Table 3.3.

Downstream evaluation. The downstream evaluation is linear classification accuracy of the learned representation f to predict the class for an image. The linear classifier is trained for 1000 epochs using Adam; hyperparameter details are presented in Table 3.3.

For the plots in Figure 3.3 we evaluate every the contrastive loss and downstream accuracy every 5 epochs of training and stop when the average test contrastive loss (window size of 5) is minimized.

Hash experiment.

To enforce the disjoint augmentation regime, we select a set of 16 pixels, and modify an augmentation by replacing those 16 pixels (8 bits each) with an 128-bit MD5 hash of the image that generated the augmentation. This way the original image hash (and thus its identity) can be recovered from any of its augmentations. The result of training on this small variation of the standard pipeline is presented in Figure 3.3 (bottom right). Some examples of this augmentation is shown in Figure 3.6 (bottom right).



Aug. distribution	Classifier acc. (%)
CIFAR-10 + SimCLR	99.623 ± 0.061
Remove crops	99.981 ± 0.024
Remove crops & flips	99.974 ± 0.038
Add hash pixels	99.935 ± 0.033

Figure 3.7: Demonstration of augmentation disjointness for CIFAR-10 with SimCLR augmentations. As described in Section 3.5.2 we train classifiers to distinguish between 5000 same-class examples, for each class. These classifiers reach $\approx 100\%$ accuracy (averaged over 10 classes) in the 5000-way classification task, in all 4 settings from Figure Figure 3.3 (standard deviations shown over 10 random epoch picked close to end of training). This is evidence that these distributions are close to the disjoint regime, despite contrastive learning leading to good downstream accuracy.

Label-orthogonal training.

We largely follow the same procedure as standard training, but modify the representation $f(x)$ for an augmentation before passing it to the contrastive loss. In particular for an augmentation and label pair (x, y), compute the representation $f(x)$ as usual. Before passing it into the contrastive loss, convert apply the transformation $f'(x) = f(x) - \mu_y$, where μ_y is the mean representation for augmentations from class y. μ_y is computed at every step, using augmentations from a memory bank of 10240 pairs of (x,y) collected over training. Then $f'(x)$ is passed into the SimCLR loss in Equation (3.54) instead of $f(x)$ and everything else remains the same. The subtraction of the mean μ_y from the representation make the representation orthogonal to the labels, thus declining its ability to linearly classify images. The result of training with this procedure is presented in Figure 3.3 (top left). Note that the implicit assumption in the calculation of the contrastive loss is that different classes do not share any augmentations, i.e. the labels are almost invariant to standard augmentations.

Table 3.4: Hyperparameter values for experiments on AG News. Unless specified, the same hyperparameter value is used for both contrastive learning and the downstream classification task.

Hyperparameters	Values
Max epoch	100
Learning rate for contrastive learning	0.01 for `BoW`, 0.001 otherwise
Learning rate for downstream linear classification	0.01
Patience	10
Batch size	128
Representation dimension	768
Gradient clipping norm	2.5

3.10.3 Experiments on text domain

Experimental Setup. We evaluate on the AG News classification dataset Zhang et al. [2015]. This dataset contains 4 classes ("World", "Sports", "Business", "Sci/Tech") and each class contains news articles from that topic. We use the tokenizer from torchtext library. If a token sequence is of length more than 60, we then trim it to its first 60 tokens, leading to a vocabulary size of 11970.

We perform contrastive learning similar to SimCLR. We train the model in epochs and in each epoch we sample pairs of augmentation for 50,000 randomly chosen pieces of text in the training dataset. We then perform a single pass over this dataset and minimize the SimCLR contrastive learning loss from Equation (3.54), with temperature $\tau = 1$. The downstream evaluation task is to simply predict the class given the text.

At the start of contrastive learning, we create a held-out validation set of pairs of augmentation sampled for 10,000 randomly chosen examples from the original validation set. At the end of each epoch of contrastive learning, we evaluate the model on this held-out validation set by computing the SimCLR loss. We also train a linear classifier on top of fixed model representations, to evaluate the model on the downstream classification task. During the downstream training, we evaluate the model at the end of epoch on the validation set and report the linear classifier with the best validation loss. We stop training if the best validation loss does not improve for κ consecutive epochs where κ is the patience hyperparameter, or if we hit a maximum number of epochs. Hyperparameter values are listed in Table Table 3.4.

Model Details. We evaluate three models on the AG News task. All models encode a given text to a d-dimensional representation. The first model is a bag of word (`BoW`) that trains a word embedding matrix and simply returns the average word embedding of tokens in the text. The second model is Gated Recurrent Unit (`GRU`), which is a recurrent neural network Chung et al. [2014]. The `GRU` is uni-directional, uses a 300

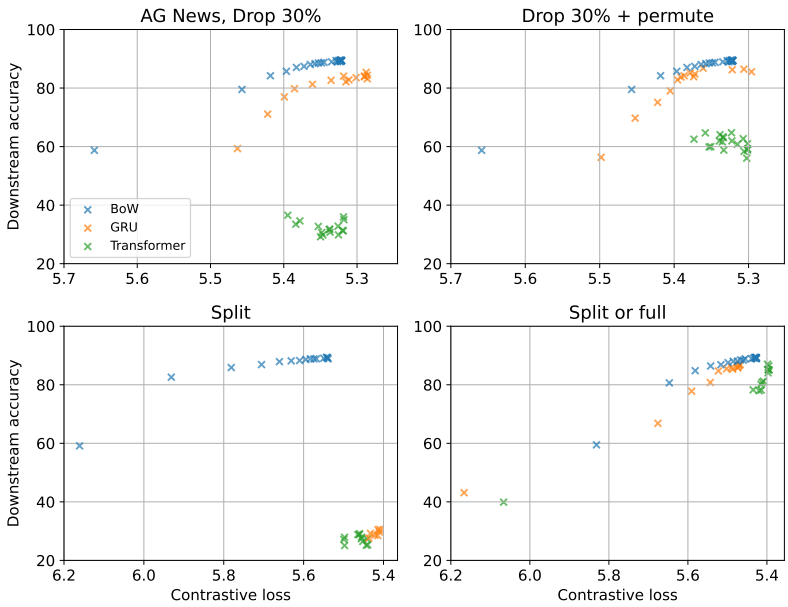


Figure 3.8: Contrastive loss → accuracy transfer plots for AG News with bag-of-words (`BoW`), `GRU` and `Transformer` architectures with representation dimensionality $d = 128$. These plots use the average representation of augmentations $f_{\mathcal{A}}$ for downstream evaluation rather than the representation f directly. Augmentations in each case are as follows: **TL:** Drop random 30% of tokens. **TR:** Drop random 30% of tokens and randomly permute the rest. **BL:** Either the first half or second half of the input. **BR:** Either the first half, second half or the full input. The plots here are almost identical to the plots from Figure 3.4, suggesting that the distribution shift from augmentations to unaugmented inputs from contrastive learning to downstream evaluation does not play a big role.

dimensional input word embedding, dropout of 0.3, hidden dimension of 768, has 4 layers, and linearly maps the hidden state representation of the final token from the layer to d-dimensions. The final model is a `Transformer` [Vaswani et al., 2017], which is the base model for many state-of-the-art neural networks in NLP. The `Transformer` is uni-directional, hidden dimension of 128, has 4 layers and 4 attention heads, and linearly maps the hidden state representation of the final token from the layer to d-dimensions.

Robust evaluation.

Standard practice is to train a representation f on augmentations x, and use the same function to compute representations for unaugmented inputs $\bar{x}$. This is the strategy we employ for the plots in Figure 3.4. However, as discussed in Section 3.2, this causes an obvious distribution shift, since the representations have been trained to output something meaningful for unaugmented inputs. This could be a potential reason for the brittle transfer performance of `GRU` and `Transformer`. However we verify that this distribution shift is not the reason, by instead evaluating downstream performance using the augmentation-averaged representation

$f_{\mathcal{A}}$, as defined in Equation (3.4). These robust evaluation transfer plots are presented in Figure 3.8, which look almost identical to those in Figure 3.4.

Visualizing 2-dimensional representations.

We train contrastive learning models with output dimensionality $d = 2$ and visualize the contrastive loss $\rightarrow$ accuracy in Figure 3.9. Firstly we note that the trends are not exactly the same as in Figure 3.4 that plot the same for $d = 128$. Most interestingly, for the split augmentation, `GRU` does not perform well on downstream accuracy for $d = 128$, but it does almost as well as `BoW` at $d = 2$. This kind of non-monotonic behavior w.r.t. representation dimensionality d is also unexplained by existing theory.

Next we visualize the learned representations for augmentations from different classes (normalized to unit norm) in Figure 3.10, for the drop augmentation. We sample 100 inputs per class and 4 augmentations per input, and encode them with the trained `BoW`, gru and `Transformer` models. For clear visualization, we plot the 4 augmentations per image with the same color, **with each of them plotted at different radii** (1.0, 1.133, 1.267, 1.4). We observe that the `BoW` representations look roughly linearly separable since different classes tend to roughly occupy different quadrants of the circle, corroborating its good downstream performance from Figure 3.9. It does so by roughly bringing augmentations of the same input (points with the same color) closer to each, although the representations not perfectly augmentation invariant. The `GRU` representations in every class, on the other hand, are spread out and brings augmentations very closer to each other than `BoW` representations, reminiscent of the uniformity and alignment properties from Wang and Isola [2020]. However these representations are not linearly separable. The `Transformer` representations are intriguing since they are not uniformly spread out, but almost perfectly augmentation invariant. Furthermore the representation distributions for different classes are identical to each other, justifying its bad downstream performance from Figure 3.9. This phenomenon aligns with our lower bound Lemmas 3.4.2 and 3.4.3, whose proofs reveal how such spurious representations can be constructed.

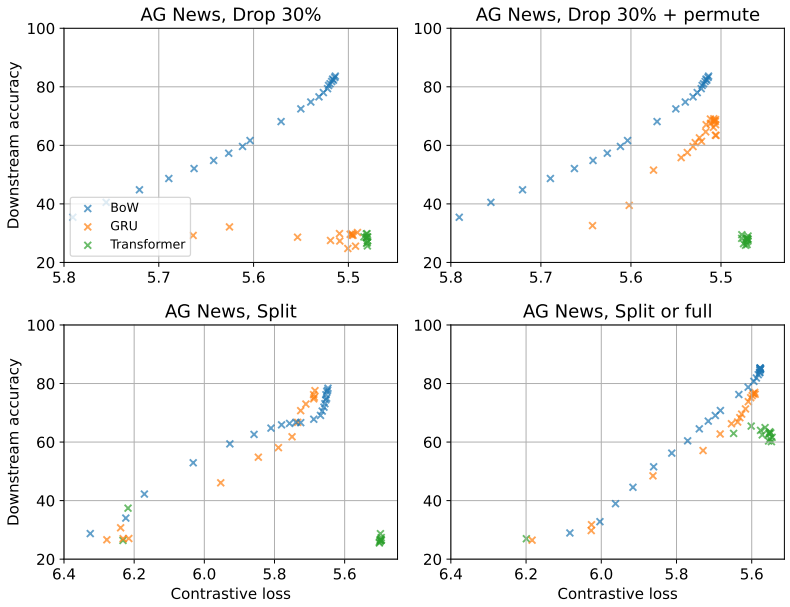


Figure 3.9: Contrastive loss → accuracy transfer plots for AG News with bag-of-words (`BoW`), `GRU` and `Transformer` architectures with representation dimensionality $d = 2$. Augmentations in each case are as follows: **TL:** Drop random 30% of tokens. **TR:** Drop random 30% of tokens and randomly permute the rest. **BL:** Either the first half or second half of the input. **BR:** Either the first half, second half or the full input. In all cases `BoW` representation does quite well downstream ($\sim 80\%$), but either `Transformer` or both `GRU` and `Transformer` demonstrate brittleness of transfer for different augmentations.

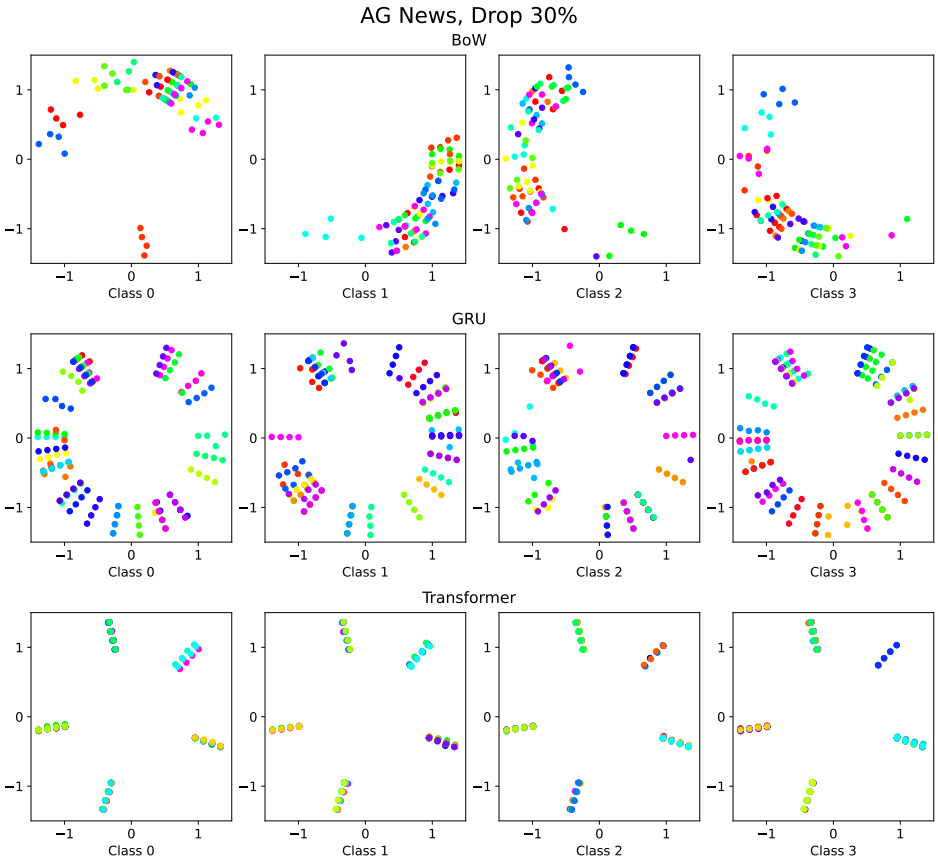


Figure 3.10: We plot representations of augmentations from different classes, for `BoW`, `GRU` and `Transformer` respectively. The 30% drop augmentation is used for these plots. While all representations are supposed to be normalized to unit norm, for clear visualization, we plot the 4 augmentations per image with the same color, **with each of them plotted at different radii** (1.0, 1.133, 1.267, 1.4). We observe that `GRU` and `Transformer` are quite augmentation invariant, but are not linearly separable. See Section 3.10.3 for more discussion about this.

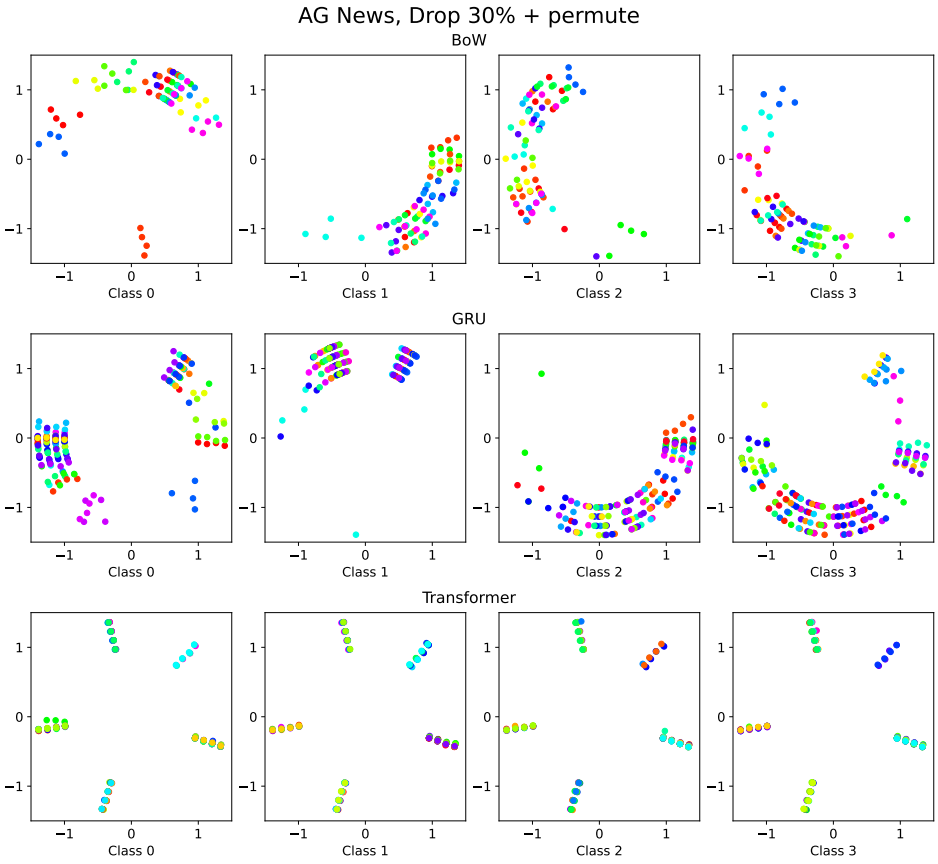


Figure 3.11: We plot representations of augmentations from different classes, for `BoW`, `GRU` and `Transformer` respectively. The 30% drop + permute augmentation is used for these plots. While all representations are supposed to be normalized to unit norm, for clear visualization, we plot the 4 augmentations per image with the same color, **with each of them plotted at different radii** (1.0, 1.133, 1.267, 1.4). We observe that `BoW` representations are roughly linearly classifiable, `GRU` representations are somewhat classifiable while `Transformer` are quite augmentation invariant, but not linearly separable.

Part II

Self-Prediction Methods

Chapter 4

Predicting What You Already Know Helps: Provable Self-Supervised Learning

In this chapter we study self-prediction based SSL methods, where the idea is to pre-train a model to predict part of an input from the rest of it. Self-prediction methods include ideas like predicting a missing image patch, recovering the color channels of an image from context, or predicting missing words in text. It is a priori puzzling as to why predicting this *known* information helps in learning representations effective for downstream tasks. We posit a mechanism exploiting the statistical connections between certain *reconstruction-based* pretext tasks that guarantee to learn a good representation. Formally, we quantify how the approximate independence between the components of the pretext task (conditional on the label and latent variables) allows us to learn representations that can solve the downstream task by just training a linear layer on top of the learned representation. We prove the linear layer yields small approximation error even for complex ground truth function class and will drastically reduce labeled sample complexity. Next, we show a simple modification of our method leads to nonlinear CCA, analogous to the popular SimSiam algorithm, and show similar guarantees for nonlinear CCA. This chapter is based on previously published work [Lee et al., 2021].

4.1 Introduction

Self-supervised learning creates pseudo labels solely based on input features, and solves auxiliary prediction tasks (or pretext tasks) in a supervised manner. However, the underlying principles of self-supervised learning are mysterious since it is a-priori unclear why predicting what we already know should help. We thus raise the following question:

What conceptual connection between pretext and downstream tasks ensures good representations? What is a good way to quantify this?

As a thought experiment, consider a simple downstream task of classifying desert, forest, and sea images. A meaningful pretext task is to predict the background color of images (known as image colorization [Zhang et al., 2016]). Denote X_1, X_2, Y to be the input image, color channel, and the downstream label respectively. Given knowledge of the label Y, one can possibly predict the background X_2 without knowing much about X_1. In other words, X_2 is approximately independent of X_1 conditional on the label Y. Consider another task of inpainting [Pathak et al., 2016] the front of a building (X_2) from the rest (X_1). While knowing the label "building" (Y) is not sufficient for successful inpainting, adding additional latent variables Z such as architectural style, location, window positions, etc. will ensure that variation in X_2 given Y, Z is small. We can mathematically interpret this as X_1 being approximate conditionally independent of X_2 given Y, Z.

The main insight that we exploit in this work is that with approximate conditional independence (as in the above examples), a method that predicts X_2 from X_1 will inadvertently implicitly encode and learn to predict Y (and Z) from X_1 as an intermediate step, and then predict X_2 from Y[1]. Building upon this insight, we make the following contributions.

Contributions. The goal of this work, as in statistical learning theory, is to investigate the *statistical connections* between the random variables of input features (in this work (X_1, X_2)) and downstream labels Y, and show how specific connections can guarantee a successful learning procedure. For self-supervised learning (SSL), success is measured using the following 2 notions, 1) expressivity, i.e. does the learned representation from SSL have the ability to express the ground truth prediction function for labels Y, and 2) sample complexity, i.e. can it do so with way fewer labeled samples than what would be required without SSL.

In this work, we establish theoretical analysis for self-supervised learning fulfilling these goals.

[1]This is formally demonstrated in the proof sketch of Lemma 4.3.2.

- We provide generalization guarantees for a class of self-supervised algorithms under a statistical assumption of *approximate conditional independence (ACI)*. Specifically, we show
  - *small representation error:* the learned representation can almost linearly separate downstream targets
  - *small estimation error:* learning a predictor for downstream tasks require very few number of samples.
- Our analysis focused on *reconstruction-based* SSL methods (Zhang et al. [2016], Pathak et al. [2016], Devlin et al. [2019], Grill et al. [2020]) is presented in sections Section 4.3 and Section 4.4. In Section 4.5, we instantiate the bound from the analysis in the topic modeling framework, a standard generative model for text [Papadimitriou et al., 2000, Hofmann, 1999], where X_1 and X_2 are chosen to be two halves of a text document. Although data can be sampled from a potentially infinite mixtures of k underlying topics, an appropriate ACI assumption can be shown that leads to a downstream sample complexity of $\mathcal{O}(k)$.
- We also build the connection and extend the analysis to a variant of the SimSiam [Chen and He, 2021] method, a non-linear canonical correlation analysis (CCA) method for self-supervised learning in Section 4.6. Further connecting this to alternating conditional expectation (ACE) algorithm [Breiman and Friedman, 1985], we show how this problem is related to decomposing the conditional distribution $X_2 \mid X_1$.
- We quantify our notion of ACI by a certain partial covariance matrix (Definition 4.4.2) and our risk bound scales linear with it. We verify this and other aspects of our main generalization bound (Theorem 4.4.4) using simulation experiments in Section 4.7. We also find that pretext task experimentally helps when CI is approximately enforced in text domain. We further demonstrate on a real-world image dataset that a pretext task-based linear model performs at least as well as many baselines.

4.1.1 Related work

Self-supervised learning (SSL) methods in practice: There has been a flurry of self-supervised methods lately. One class of methods reconstruct images from corrupted or incomplete versions of it, like denoising auto-encoders [Vincent et al., 2008], image inpainting [Pathak et al., 2016], and split-brain autoencoder [Zhang et al., 2017b]. Pretext tasks are also created using visual common sense, including predicting rotation angle [Gidaris et al., 2018], relative patch position [Doersch et al., 2015], recovering color channels [Zhang et al., 2016], solving jigsaw puzzle games [Noroozi and Favaro, 2016], and discriminating images created from distortion [Dosovitskiy et al., 2014]. We refer to the above procedures as reconstruction-based SSL. Another

popular paradigm is contrastive learning [Chen et al., 2020a,b]. The idea is to learn representations that bring similar data points closer while pushing randomly selected points further away [Wang and Gupta, 2015, Logeswaran and Lee, 2018, Arora et al., 2019] or to maximize a contrastive-based mutual information lower bound between different views [Hjelm et al., 2019, Oord et al., 2018, Tian et al., 2020a]. A popular approach for text domain is based on language modeling where models like BERT and GPT create auxiliary tasks for next word predictions [Devlin et al., 2019, Radford et al., 2018]. The natural ordering or topology of data is also exploited in video-based [Wei et al., 2018, Misra et al., 2016, Fernando et al., 2017], graph-based [Yang et al., 2021, Hu et al., 2020] or map-based [Zhang et al., 2019] SSL. For instance, the pretext task is to determine the correct temporal order for video frames as in [Misra et al., 2016].

Theory for SSL: While we theoretically study reconstruction-based SSL, prior work has different flavors of theoretical results for different kinds of SSL methods. Most relevant are the guarantees for representation learning using SSL methods on downstream tasks that just learn a linear classifier on top of the learned representations. Arora et al. [2019] shows guarantees for representations from a contrastive learning objective: $L_1^{cont}(\psi) = \mathbb{E}_{(X_1,X_2),X_2'}[\log(1 + e^{-\psi(X_1)^\top \psi(X_2)+\psi(X_1)^\top \psi(X_2')})]$. Under a class conditional independence assumption, i.e. $X_1 \perp X_2 \mid Y$, they show that representation ψ that does well on contrastive objective, i.e. $L_1^{cont}(\psi) \leq \epsilon$, will have $\mathcal{O}(\epsilon)$ linear classification loss on the average binary task involving pairs of classes (y_1, y_2). However, their analysis cannot handle the general case of approximate conditional independence. Recently, Tosh et al. [2021b] show that contrastive learning representations can *linearly* recover continuous functions of the underlying topic posterior under a topic modeling assumption for text. While their assumption bears similarity to ours, the assumption of independent sampling of words is strong and does not generalizable to other domains like images. Most relevant is a concurrent work [Tosh et al., 2021a] that shows guarantees for a contrastive learning objective that looks like $L_2^{cont}(\psi, \eta) = \mathbb{E}_{(X_1,X_2),X_2'}\left[\log(1 + e^{-\psi(X_1)^\top \eta(X_2)}) + \log(1 + e^{\psi(X_1)^\top \eta(X_2')})\right]$, with a multi-view redundancy assumptions that is very similar to our ACI assumption. We take a closer look at their assumption in Section 4.14.2. All the above objectives are different from the simple reconstruction-based objective we consider: $L(\psi) = \mathbb{E}_{(X_1,X_2)}\left[\|X_2 - \psi(X_1)\|^2\right]$. Saunshi et al. [2021] show guarantees for representations learned using language modeling on sentence classification tasks. Some more recent work [Tsai et al., 2020, Mitrovic et al., 2021, Tian et al., 2020c, Wang and Isola, 2020] provide theoretical understanding on SSL respectively based on causality, mutual information, gradient-descent dynamics, and alignment/uniformity of representations, without explicit risk bounds for downstream tasks. There is a mutual information maximization view of con-

trastive learning, but Tschannen et al. [2020] points out issues with it. Previous attempts to explain negative sampling [Mikolov et al., 2013b] based methods use the theory of noise contrastive estimation [Gutmann and Hyvärinen, 2010, Ma and Collins, 2018] to show asymptotic guarantees, without explicit connections to downstream tasks. CI is also used in sufficient dimension reduction [Fukumizu et al., 2009, 2004], while CI and redundancy assumptions on multiple views [Kakade and Foster, 2007, Ando and Zhang, 2007] are used to analyze a canonical-correlation based dimension reduction algorithm and also for self-supervised learning algorithms like co-training [Blum and Mitchell, 1998]. Finally, Alain and Bengio [2014], Vincent [2011] provide a theoretical analysis for denoising auto-encoder.

4.1.2 Overview of results:

Section 4.2 introduces notation, setup, and the self-supervised learning procedure considered in this work. In Section 4.3, we analyze downstream sample complexity under exact CI and unlimited labeled data to highlight the key ideas. Section 4.4 presents our main result with relaxed conditions: under ACI with latent variables, and assuming finite samples in both pretext and downstream tasks, for various function classes, and both regression and classification tasks. Section 4.5 demonstrates our results with an example in the setting of topic modeling. In Section 4.6 we extend our results to self-supervised tasks that enforce two views of data to have similar representations, or namely SimSiam Chen and He [2021]. Experiments verifying our theoretical findings are in Section 4.7. Proofs of most results are in the Appendix.

4.2 Preliminary

4.2.1 Notation

We use lower case symbols (x) to denote scalar quantities, bold lower case symbols ($\boldsymbol{x}$) for vector values, capital letters (X) for random variables, and capital and bold letters $\boldsymbol{X}$ for matrices. P_X denotes the probability law of random variable X, and the space of square-integrable functions with probability P is denoted by $L^2(P)$. We use standard $\mathcal{O}$ notation to hide universal factors and $\tilde{\mathcal{O}}$ to hide log factors. $\|\cdot\|$ stands for ℓ_2-norm for vectors or Frobenius norm for matrices.

Linear conditional expectation. $\mathbb{E}^L[Y|X]$ denotes the prediction of Y with linear regression:

$$\mathbb{E}^L[Y|X=\boldsymbol{x}] := \boldsymbol{W}^*\boldsymbol{x} + \boldsymbol{b}^*, \quad \text{where } \boldsymbol{W}^*, \boldsymbol{b}^* := \arg\min_{\boldsymbol{W},\boldsymbol{b}} \mathbb{E}[\|Y - \boldsymbol{W}X - \boldsymbol{b}\|^2].$$

In other words, $\mathbb{E}^L[Y|X]$ denotes the best linear predictor of Y given X. We also note that $\mathbb{E}[Y|X] \equiv \arg\min_f \mathbb{E}[\|Y - f(X)\|^2]$ is the best predictor of Y given X.

(Partial) covariance matrix. For random variables X, Y, we denote $\mathbf{\Sigma}_{XY}$ to be covariance matrix of X and Y. For simplicity in most cases, we assume $\mathbb{E}[X] = 0$ and $\mathbb{E}[Y] = 0$; thus we do not distinguish $\mathbb{E}[XY]$ and $\mathbf{\Sigma}_{XY}$. The partial covariance matrix between X and Y given Z is:

$$\mathbf{\Sigma}_{XY|Z} := \text{cov}\{X - \mathbb{E}^L[X|Z], Y - \mathbb{E}^L[Y|Z]\} \equiv \mathbf{\Sigma}_{XY} - \mathbf{\Sigma}_{XZ}\mathbf{\Sigma}_{ZZ}^{-1}\mathbf{\Sigma}_{ZY}, \tag{4.1}$$

which captures the correlation between X and Y setting aside the effect of Z.

Sub-gaussian random vectors. $X \in \mathbb{R}^d$ is ρ^2-sub-gaussian if for every fixed unit vector $\boldsymbol{v} \in \mathbb{R}^d$, the variable $\boldsymbol{v}^\top X$ is ρ^2-sub-gaussian, i.e., $\mathbb{E}[e^{s \cdot \boldsymbol{v}^\top (X - \mathbb{E}[X])}] \leq e^{s^2\rho^2/2}$ $(\forall s \in \mathbb{R})$.

4.2.2 Setup and methodology

We denote by X_1 the input variable, X_2 the target random variable for the pretext task, and Y the label for the downstream task, with $X_1 \in \mathcal{X}_1 \subset \mathbb{R}^{d_1}, X_2 \in \mathcal{X}_2 \subset \mathbb{R}^{d_2}$ and $Y \in \mathcal{Y} \subset \mathbb{R}^k$. If $\mathcal{Y}$ is finite with $|\mathcal{Y}| = k$, we assume $\mathcal{Y} \subset \mathbb{R}^k$ is the one-hot encoding of the labels. $P_{X_1X_2Y}$ denotes the joint distribution over $\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{Y}$. P_{X_1Y}, P_{X_1} denote the corresponding marginal distributions. Our proposed self-supervised learning aims to fulfill the following two steps:

Step 1 (pretext task): Learn a representation $\psi(\boldsymbol{x}_1)$ close to $\psi^* \coloneqq \arg\min_{g \in \mathcal{H}} \mathbb{E}\|X_2 - g(X_1)\|^2$, where $\mathcal{H}$ can vary for different settings that we will specify and discuss later.

Step 2 (downstream task): Perform linear regression on Y with $\psi(X_1)$, i.e. $f(\boldsymbol{x}_1) := (\boldsymbol{W}^*)^\top \psi(\boldsymbol{x}_1)$, where $\boldsymbol{W}^* \leftarrow \arg\min_{\boldsymbol{W}} \mathbb{E}_{X_1,Y}[\|Y - \boldsymbol{W}^\top \psi(X_1)\|^2]$. Namely we learn $f(\cdot) = \mathbb{E}^L[Y|\psi(\cdot)]$.

We study this simplified version in the main text, where in practice, the SSL procedure may utilize an encoder-decoder structure, while the downstream task uses both X_1 and X_2 to predict Y. We incorporate these extensions in Section 4.11.3 and Section 4.14.3.

With finite samples, performance of a learned representation ψ on the downstream task depends on the following quantities that capture expressivity and sample complexity respectively:

Approximation error indicates whether Y is *linearly separable* by the learned representation ψ, thus

measuring expressivity. We measure this by comparing $\boldsymbol{W}\psi(X_1)$ to the optimal predictor $f^* := \mathbb{E}[Y|X_1 = \boldsymbol{x}_1]$. Denote $e_{\text{apx}}(\psi) = \min_{\boldsymbol{W}} \mathbb{E}[\|f^*(X_1) - \boldsymbol{W}\psi(X_1)\|^2]$. This gives a measure of how well ψ can linearly predict Y when given infinite samples for the task.

Estimation error measure sample complexity of ψ on the downstream task and assume access to n_2 i.i.d. samples $(\boldsymbol{x}_1^{(1)}, \boldsymbol{y}^{(1)}), \cdots, (\boldsymbol{x}_1^{(n_2)}, \boldsymbol{y}^{(n_2)})$ drawn from $P_{X_1 Y}$. We express the n_2 samples collectively as $\boldsymbol{X}_1^{\text{down}} \in \mathbb{R}^{n_2 \times d_1}$, $\boldsymbol{Y} \in \mathbb{R}^{n_2 \times k}$ and overload notation to say $\psi(\boldsymbol{X}_1^{\text{down}}) = \left[\psi(\boldsymbol{x}_1^{(1)})|\psi(\boldsymbol{x}_1^{(2)})\cdots|\psi(\boldsymbol{x}_1^{(n_2)})\right]^\top \in \mathbb{R}^{n_2 \times d_2}$. We perform linear regression on the learned representation ψ and measure excess risk, that incorporates both approximation and estimation errors.

$$\hat{\boldsymbol{W}} \leftarrow \arg\min_{\boldsymbol{W}} \frac{1}{2n_2}\|\boldsymbol{Y} - \psi(\boldsymbol{X}_1)\boldsymbol{W}\|_F^2; \quad \text{ER}_\psi(\hat{\boldsymbol{W}}) \coloneqq \frac{1}{2}\mathbb{E}\|f^*(X_1) - \hat{\boldsymbol{W}}^\top \psi(X_1)\|_2^2.$$

4.3 Guaranteed recovery with conditional independence

In this section, we focus on the case where the input X_1 and pretext target X_2 are conditionally independent (CI) given the downstream label Y. While this is a strong assumption that is rarely satisfied in practice, it helps us understand the role of CI with clean results and builds up to our main results with ACI with latent variables in Section 4.4. As a warm-up, we show how CI helps when (X_1, X_2, Y) are jointly Gaussian to give us a flavor for the results to follow in Section 4.10. We then analyze it for general random variables under two settings: (a) when the function class used for ψ is universal, (b) when ψ is restricted to be a linear function of given features. For now we assume access to a large amount of unlabeled data so as to learn the optimal ψ^* perfectly and this will be relaxed later in Section 4.4. The general recipe for the results is as follows:

1. Find a closed-form expression for the optimal solution ψ^* for the pretext task.
2. Use conditional independence to show that optimal f^* is linear in ψ^*, i.e., $e_{\text{apx}}(\psi^*)$ is small.
3. Exploit the low rank structure of ψ^* to show small estimation error on downstream tasks.

Data assumption. Suppose $Y = f^*(X_1) + N$, where $f^* = \mathbb{E}[Y|X_1]$ and $\mathbb{E}[N] = 0$. We assume N is σ^2-subgaussian. For simplicity, we assume non-degeneracy: $\boldsymbol{\Sigma}_{X_i X_i}$, $\boldsymbol{\Sigma}_{YY}$ are full rank.

Assumption 4.3.1. *Let $X_1 \in \mathbb{R}^{d_1}, X_2 \in \mathbb{R}^{d_2}$ be random variables from some unknown distribution. Let label $Y \in \mathcal{Y}$ be a discrete random variable with $k = |\mathcal{Y}| < d_2$. We assume conditional independence: $X_1 \perp X_2 | Y$.*

Here Y can be interpreted as the multi-class labels where k is the number of classes. For regression problems, one can think about Y as the discretized values of continuous labels. We do not specify the dimension for Y since Y could be arbitrarily encoded but the results only depend on k and the variance of Y (conditional on the input X_1).

4.3.1 Universal function class.

Suppose we learn the optimal ψ^* among all measurable functions The optimal function ψ^* in this case is naturally given by conditional expectation: $\psi^*(\boldsymbol{x}_1) = \mathbb{E}[X_2|X_1 = \boldsymbol{x}_1]$. We show that CI implies that ψ^* is good for downstream tasks, which is not apriori clear.

Lemma 4.3.2 (Approximation error)**.** *If random variables X_1, X_2, Y satisfy Assumption 4.3.1, and $\boldsymbol{A} \in \mathbb{R}^{\mathcal{Y}\times d_2}$ with $\boldsymbol{A}_{y,:} := \mathbb{E}[X_2|Y = \boldsymbol{y}]$ has rank $k = |\mathcal{Y}|$. Then $f^* \equiv \boldsymbol{W}^*\psi^*$, i.e., $e_{apx}(\psi^*) = 0$.*

This tells us that although f^* could be nonlinear in $\boldsymbol{x}_1$, it is guaranteed to be linear in $\psi^*(\boldsymbol{x}_1)$.

Proof Sketch of Lemma 4.3.2. Lemma is proved by law of total expectation:

$$\begin{aligned}\psi^*(\cdot) := \mathbb{E}[X_2|X_1] =& \mathbb{E}[\mathbb{E}[X_2|X_1, Y]|X_1] = \mathbb{E}[\mathbb{E}[X_2|Y]|X_1] && \text{(uses CI)}\\ &= \sum_y P(Y = y|X_1)\mathbb{E}[X_2|Y = y] =: f(X_1)^\top \boldsymbol{A},\end{aligned}$$

where $f(x_1)_y = P(Y = y|X_1 = x_1)$, and $\boldsymbol{A} \in \mathbb{R}^{\mathcal{Y}\times d_2}$ satisfies $\boldsymbol{A}_{y,:} = \mathbb{E}[X_2|Y = y]$. One could see that through predicting X_2, due to the CI assumption, ψ^* has implicitly encoded the information of $Y|X_1$. Finally due to the fact that matrix $\boldsymbol{A}$ is full rank, we get that f^* is linear in ψ^* as well. □

We see that besides CI, another important property is $\mathbb{E}[X_2|Y]$ being rank k. This means X_2 is correlated with every instance of Y, and thus captures information of every prediction class. This is naturally a necessary assumption for X_2 to be a reasonable pretext task for predicting Y. Note that this assumption does not trivialize the problem and that even though ψ is designed to predict X_2, it can still be a better representation than X_2 for downstream tasks. Note that Y does not have to be linear in X_2 but is proven to be linear in ψ, since ψ learns to ignore some information in X_2 that is irrelevant to Y. We provide this simple example for better understanding:

Example 4.3.3. *Let $Y \in \{-1, 1\}$ be binary labels, and X_1, X_2 be $2-$mixture Gaussian random variables*

with $X_1 \sim \mathcal{N}(Y\boldsymbol{\mu}_1, \mathbf{I}), X_2 \sim \mathcal{N}(Y\boldsymbol{\mu}_2, \mathbf{I})$. *In this example,* $X_1 \perp X_2 | Y$. *Although* $\mathbb{E}[Y|X_2]$ *and* $\mathbb{E}[Y|X_1]$ *are not linear,* $\mathbb{E}[Y|\psi]$ *is linear:* $\psi(\boldsymbol{x}_1) = P(Y=1|X_1=\boldsymbol{x}_1)\boldsymbol{\mu}_2 - P(Y=-1|X_1=\boldsymbol{x}_1)\boldsymbol{\mu}_2$ *and* $f^*(\boldsymbol{x}_1) = P(Y=1|X_1=\boldsymbol{x}_1) - P(Y=-1|X_1=\boldsymbol{x}_1) \equiv \boldsymbol{\mu}_2^T\psi(\boldsymbol{x}_1)/\|\boldsymbol{\mu}_2\|^2$.

Given that ψ^* is good for downstream, we now care about the sample complexity. We will need to assume that the representation has some nice concentration properties. We make an assumption about the whitened data $\psi^*(X_1)$ to ignore scaling factors.

Assumption 4.3.4. *We assume the whitened feature variable* $U := \boldsymbol{\Sigma}_\psi^{-1/2}\psi(X_1)$ *is a* ρ^2*-subgaussian random variable, where* $\boldsymbol{\Sigma}_\psi = \mathbb{E}[\psi(X_1)\psi(X_1)^\top]$.

We note that all bounded random variables satisfy sub-gaussian property.

Theorem 4.3.5 (General conditional independence)**.** *Fix a failure probability* $\delta \in (0,1)$*, under the same assumption as Lemma 4.3.2 and Assumption 4.3.4 for* ψ^**, if additionally* $n_2 \gg \rho^4(k + \log(1/\delta))$*, then the excess risk of the learned predictor* $\boldsymbol{x}_1 \to \hat{\boldsymbol{W}}^\top\psi^*(\boldsymbol{x}_1)$ *on the downstream task satsifies*

$$\mathrm{ER}_{\psi^*}[\hat{\boldsymbol{W}}] \leq \tilde{\mathcal{O}}\left(\tfrac{k}{n_2}\sigma^2\right)^2$$

Remark 4.3.6. *This analysis assumes we could perfectly learn* $\psi^* = \mathbb{E}[X_2|X_1]$ *disregarding the number of samples in the SSL phase (unlabeled data is cheap to obtain). Here by sample complexity we refer to the labeled data* (X_1, Y)*. We defer the effect of imprecise representation* ψ *to Section 4.4.*

4.3.2 Function class induced by feature maps.

Given feature map $\phi_1 : \mathcal{X}_1 \to \mathbb{R}^{D_1}$, we consider the function class $\mathcal{H}_1 = \{\psi : \mathcal{X}_1 \to \mathbb{R}^{d_2} | \exists \boldsymbol{B} \in \mathbb{R}^{d_2 \times D_1}, \psi(\boldsymbol{x}_1) = \boldsymbol{B}\phi_1(\boldsymbol{x}_1)\}$.

Claim 4.3.7 (Closed form solution)**.** *The optimal function in* $\mathcal{H}$ *is* $\psi^*(\boldsymbol{x}_1) = \boldsymbol{\Sigma}_{X_2\phi_1}\boldsymbol{\Sigma}_{\phi_1\phi_1}^{-1}\phi_1(\boldsymbol{x}_1)$*, where* $\boldsymbol{\Sigma}_{X_2\phi_1} := \boldsymbol{\Sigma}_{X_2\phi_1(X_1)}$ *and* $\boldsymbol{\Sigma}_{\phi_1\phi_1} := \boldsymbol{\Sigma}_{\phi_1(X_1)\phi_1(X_1)}$.

We again show the benefit of CI, but only comparing the performance of ψ^* to the original features ϕ_1. Since ψ^* is linear in ϕ_1, it cannot have smaller approximation error than ϕ_1. However CI will ensure that ψ^* has the same approximation error as ϕ_1 and enjoys better sample complexity.

Lemma 4.3.8 (Approximation error)**.** *If Assumption 4.3.1 is satisfied, and if the matrix* $\boldsymbol{A} \in \mathbb{R}^{\mathcal{Y} \times d_2}$ *with*

[2] We will use $\tilde{O}$ to hide log factor $\log(k/\delta)$ or $\log(d_2/\delta)$.

$\boldsymbol{A}_{y,:} := \mathbb{E}[X_2|Y = \boldsymbol{y}]$ *is of rank* $k = |\mathcal{Y}|$. *Then* $e_{apx}(\psi^*) = e_{apx}(\phi_1)$.

We additionally need an assumption on the residual $a(\boldsymbol{x}_1) := \mathbb{E}[Y|X_1 = \boldsymbol{x}_1] - \mathbb{E}^L[Y|\phi_1(\boldsymbol{x}_1)]$.

Assumption 4.3.9. *(Bounded approx. error; Condition 3 in Hsu et al. [2012])) We have almost surely*

$$\|\boldsymbol{\Sigma}_{\phi_1\phi_1}^{-1/2}\phi_1(X_1)a(X_1)^\top\|_F \leq b_0\sqrt{k}$$

Theorem 4.3.10. *(CI with approximation error) Fix a failure probability* $\delta \in (0,1)$*, under the same assumption as Lemma 4.3.8, Assumption 4.3.4 for* ψ^* *and Assumption 4.3.9, if* $n_2 \gg \rho^4(k + \log(1/\delta))$*, then the excess risk of the learned predictor* $\boldsymbol{x}_1 \to \hat{\boldsymbol{W}}^\top\psi^*(\boldsymbol{x}_1)$ *on the downstream task satisfies:*

$$\mathrm{ER}_{\psi^*}[\hat{\boldsymbol{W}}] \leq e_{apx}(\phi_1) + \tilde{\mathcal{O}}\left(\tfrac{k}{n_2}\sigma^2\right).$$

Thus with SSL, the requirement of labels is reduced from complexity for D_1 to $\mathcal{O}(k)$.

4.4 Beyond conditional independence

In the previous section, we focused on the case where we have exact CI. A weaker but more realistic assumption is that Y captures some portion of the dependence between X_1 and X_2 but not all. We quantify this notion of approximate ACI through a quantity ϵ^2_{CI} (Definition 4.4.2), and show excess risk bounds for the representation learned from SSL[3]. In particular, the excess risk will have the form $\tilde{\mathcal{O}}\left(\frac{d_2}{n_2} + \epsilon^2_{\mathrm{CI}} + \epsilon^2_{\mathrm{pre}}\right)$, which suggests that only $n_2 = \mathcal{O}(d_2)$ labeled samples will be required to get small error on downstream task, as long as approximate CI is satisfied (ϵ^2_{CI} is small) and the pretext task is solved well enough ($\epsilon^2_{\mathrm{pre}}$ is small). This is in contrast to not doing SSL, where many more labeled samples will be required to learn a solve the downstream task that learns a complicated representation function from scratch. We now describe the SSL method on finite samples, followed by the definition of ACI which we use to discuss the main excess risk bound and its consequences.

SSL with finite samples and general function space: Let $\boldsymbol{X}_1^{\mathrm{pre}} = [\boldsymbol{x}_1^{(1,\mathrm{pre})}, \cdots, \boldsymbol{x}_1^{(n_1,\mathrm{pre})}]^\top \in \mathbb{R}^{n_1\times d_1}$ and $\boldsymbol{X}_2 = [\boldsymbol{x}_2^{(1)}, \cdots, \boldsymbol{x}_2^{(n_1)}]^\top \in \mathbb{R}^{n_1\times d_2}$ be n_1 training samples for pretext task, where $(\boldsymbol{x}_1^{(i,\mathrm{pre})}, \boldsymbol{x}_2^{(i)})$ is sampled from $P_{X_1X_2}$. The n_2 labeled samples for the downstream task are defined as $\boldsymbol{X}_1^{\mathrm{down}} \in \mathbb{R}^{n_2\times d_1}$, $\boldsymbol{Y} \in \mathbb{R}^{n_2\times d_3}$[4].

[3]Results for jointly-Gaussian variables is in Section 4.12.1; ACI is quantified by the partial covariance matrix.

[4]$d_3 = k$ and $Y \equiv \phi_y(Y)$ (one-hot encoding) refers multi-class classification task, $d_3 = 1$ refers to regression.

Given a representation function space $\mathcal{H} : \mathcal{X}_1 \to \mathbb{R}^{d_2}$, we learn $\tilde{\psi}$ from $\mathcal{H}$ using the n_1 unlabeled samples and then use the n_2 labeled samples to learn a linear classifier on the learned representation $\tilde{\psi}(\boldsymbol{X}_1^{\text{down}})$ to fit $\boldsymbol{Y}$. This process is summarized below.

$$1)\ \ \tilde{\psi} := \arg\min_{\psi \in \mathcal{H}} \frac{1}{n_1} \|\boldsymbol{X}_2 - \psi(\boldsymbol{X}_1^{\text{pre}})\|_F^2,\ 2)\ \ \hat{\boldsymbol{W}} \leftarrow \arg\min_{\boldsymbol{W}} \frac{1}{2n_2} \|\boldsymbol{Y} - \tilde{\psi}(\boldsymbol{X}_1^{\text{down}})\boldsymbol{W}\|_F^2. \tag{4.2}$$

In our main results, we consider two types of function spaces: $\mathcal{H} \in \{\mathcal{H}_1, \mathcal{H}_u\}$. Recall that $\mathcal{H}_1 = \{\psi(\cdot) = \boldsymbol{B}\phi_1(\cdot); \boldsymbol{B} \in \mathbb{R}^{d_2 \times D_1}\}$ is a class of *linear representations* induced by feature map $\phi_1 : \mathcal{X}_1 \to \mathbb{R}^{D_1}$. We use $\mathcal{H}_u$ to denote a function space with universal approximation power (e.g. deep networks) that ensures $\psi^* = \mathbb{E}[X_2|X_1] \in \mathcal{H}_u$. We define the optimal predictor in each case as $f^*_{\mathcal{H}}(X_1) = \mathbb{E}^L[Y|\phi_1(X_1)]$ when $\mathcal{H} = \mathcal{H}_1$, $f^*_{\mathcal{H}} = f^*$ for $\mathcal{H} = \mathcal{H}_u$, we define excess risk as

$$\text{ER}_{\tilde{\psi}}(\hat{\boldsymbol{W}}) := \mathbb{E}_{X_1}\left[\|f^*_{\mathcal{H}}(X_1) - \hat{\boldsymbol{W}}^\top \tilde{\psi}(X_1)\|_2^2\right].$$

Approximate conditional independence: Our new assumption will generalize Assumption 4.3.1 in two ways, 1) we allow for additional latent variables Z that together with Y could potentially make X_1 and X_2 independent, and 2) we allow this conditional independence to be approximate. Note that allowing for extra latent variable can trivially make X_1 and X_2 to be conditionally independent by picking a large enough Z (e.g. $Z = (X_1, X_2)$). However the following assumption, that needs the pretext target X_2 to correlate with all instances of variable $\bar{Y} = [Y, Z]$ (analogous to Lemma 4.3.2), will impose this restriction on how large Z can be.

Assumption 4.4.1 (Correlation between X_2 and Y, Z)**.** *Suppose there exists latent variable $Z \in \mathcal{Z}, |\mathcal{Z}| = m$ that ensures $\boldsymbol{\Sigma}_{\phi_{\bar{y}} X_2}$ is full column rank and $\|\boldsymbol{\Sigma}_{Y\phi_{\bar{y}}} \boldsymbol{\Sigma}^\dagger_{X_2\phi_{\bar{y}}}\|_2 = 1/\beta$, where $A^\dagger$ is pseudo-inverse, and $\phi_{\bar{y}}$ is the one-hot embedding for $\bar{Y} = [Y, Z]$.*

Just as in Section 4.3, this assumption will not assume away the problem (Example Example 4.3.3 can be suitably extended). The additional term $1/\beta$ here captures both the "scale" of X_2 and also the strength of correlation between X_2 and $[Y, Z]$ that was discussed after Lemma 4.3.2. For $\boldsymbol{\Sigma}_{\phi_{\bar{y}} X_2}$ to be full column rank, it is essential that $d_2 \geq km$, and this already gives an upper bound on the size of Z. Given this restriction on Z (and thus $\bar{Y}$), we define the notion of approximate conditional independence.

Definition 4.4.2 (Approximate conditional independence with function space $\mathcal{H}$)**.** *For $\bar{Y} = [Y, Z]$,*

1. For $\mathcal{H} = \mathcal{H}_1$, define $\epsilon_{CI} := \|\boldsymbol{\Sigma}_{\phi_1\phi_1}^{-1/2}\boldsymbol{\Sigma}_{\phi_1 X_2|\phi_{\bar{y}}}\|_F$.

2. For $\mathcal{H} = \mathcal{H}_u$, define $\epsilon_{CI}^2 := \mathbb{E}_{X_1}[\|\mathbb{E}[X_2|X_1] - \mathbb{E}_{\bar{Y}}[\mathbb{E}[X_2|\bar{Y}]|X_1]\|^2]$.

Firstly we note that this is indeed an extension of exact CI, since exact CI in both cases will imply that $\epsilon_{\text{CI}} = 0$. We present a unified analysis in the appendix that shows the ϵ_{CI} for the second case is same as the first case, with covariance operators instead of matrices (A direct derivation is in Claim Claim 4.12.9). We also present more relaxed and general form of the above assumptions in Section 4.14.1. With this assumption, we are ready to present our main bound.

Bound on excess risk: Recall that we assume that the residual term $N := Y - \mathbb{E}[Y|X_1]$ is mean zero and σ^2-subgaussian. Before showing our main result, analogous to Assumption 4.3.9, for the class $\mathcal{H}_1$ with non-universal features ϕ_1, we will need an assumption[5] on the residual $a := f^* - f^*_{\mathcal{H}_1} = \mathbb{E}[Y|X_1] - \mathbb{E}^L[Y|\phi_1(X_1)]$:

Assumption 4.4.3. *(Bounded approximation error on pretext phase [Hsu et al., 2012]) There exists a universal constant b_0, such that $\|\boldsymbol{\Sigma}_{\phi_1\phi_1}^{-1/2}\phi_1(X_1)a(X_1)^\top\|_F \le b_0\sqrt{d_2}$ almost surely.*

Theorem 4.4.4. *For a fixed $\delta \in (0,1)$, under Assumptions 4.3.4 and 4.4.1 for $\tilde{\psi}$ and ψ^* and Assumption 4.4.3 for non-universal feature maps, if $n_1, n_2 \gg \rho^4(d_2 + \log 1/\delta)$, and we learn the pretext tasks such that: $\mathbb{E}\|\tilde{\psi}(X_1) - \psi^*(X_1)\|_F^2 \le \epsilon_{pre}^2$. Then the generalization error for downstream task w.p. $1-\delta$ is:*

$$\text{ER}_{\tilde{\psi}}(\hat{\boldsymbol{W}}) \le \tilde{\mathcal{O}}\left(\underbrace{\sigma^2\frac{d_2}{n_2}}_{\text{estimation error}} + \underbrace{\frac{\epsilon_{CI}^2}{\beta^2} + \frac{\epsilon_{pre}^2}{\beta^2}}_{\text{approximation error}}\right) \tag{4.3}$$

We defer the proof to the appendix. The proof technique is similar to that of Section 4.3. The difference is that now $\tilde{\psi}(\boldsymbol{X}^{(\text{down})}) \in \mathbb{R}^{n_2 \times d_2}$ will be an approximately low rank matrix, where the low rank part is the high-signal features that implicitly comes from Y, Z that can linearly learn downstream task. The remaining part comes from ϵ_{CI} and ϵ_{pre} and causes the approximation error. Again by selecting the top km (dimension of $\phi_{\bar{y}}$) features we could further improve the bound:

Remark 4.4.5. *By applying PCA on $\tilde{\psi}(\boldsymbol{X}_1^{down})$ and keeping the top km principal components only, we can*

[5]This rules out the failure if one chooses a very simple function class to learn $\mathbb{E}[X_2|X_1]$. In practice we usually use neural networks (with universal approximation power) and this bound should be very small.

improve the bound in Theorem 4.4.4 to $\mathrm{ER}_{\tilde{\psi}}(\hat{\boldsymbol{W}}) \leq \tilde{\mathcal{O}}\left(\sigma^2 \frac{km}{n_2} + \frac{\epsilon_{CI}^2}{\beta^2} + \frac{\epsilon_{pre}^2}{\beta^2}\right)$.

We take a closer look at the different sources of errors in Remark 4.4.5: 1) The first term is estimation error on learning with finite samples n_2 with noise level σ^2 in $Y - f^*(X_1)$; 2) ϵ_{CI} measures the approximate CI; and 3) ϵ_{pre} is the error from not learning the pretext task exactly. The first term is optimal ignoring log factors as we do linear regression on mk-dimensional features. The second and third term together form approximation error. They are non-reducible due to the fact that f^* is not exactly linear in ψ and we use it as a fixed representation. Fine-tuning the representations might be necessary to get rid of these terms when we have sufficient downstream labeled data. We leave this exploring this as future work. Compared to traditional supervised learning, learning $f^*_{\mathcal{H}}$ requires sample complexity scaling with the (Rademacher/Gaussian) complexity of $\mathcal{H}$ (see e.g. Bartlett and Mendelson [2002], Shalev-Shwartz and Ben-David [2014]), which is very large for complicated models such as deep networks. Thus SSL can significantly reduce the labeled sample complexity down from this complexity measure of $\mathcal{H}$ to $\tilde{\mathcal{O}}(km)$, demonstrating the power of predicting what you already know using unlabeled data. In Section 4.15, we consider a similar result for classification.

4.5 Example: topic modeling

In this section, we will demonstrate how our framework can be instantiated for mixed-membership models including topic models, not just clustering. Topic modeling for text has a rich literature [Papadimitriou et al., 2000, Hofmann, 1999, Blei et al., 2003, Arora et al., 2012, 2013] and is used for analyzing and designing algorithms for information retrieval, dimensionality reduction and data analysis for large text corpora. We describe the basic setup below, followed by how our results for reconstruction-based SSL can be instantiated to learn such models.

For a set S, let Δ_S denote the set of all distributions on S. In the topic modeling framework, generation of a text document with a vocabulary set $[V] = \{1, \ldots, V\}$ is governed by certain latent topics from the set $[k]$, where k is the total number of topics. Each topic $i \in [k]$ is associated with a distribution over the vocabulary $[V]$ that is denoted by vector $A_i \in \Delta_{[V]}$; stack these vectors into the columns of a matrix $A \in \mathbb{R}^{V \times k}$. A document $X = (x_1, \ldots, x_n) \in [V]^N$ of length N is then sampled from a mixture of the k topics $\mu \in \Delta_{[k]}$. The generative process is described below:

1. Sample a topic mixture $\mu \sim \tau$, where τ is some underlying distribution over Δ_k, i.e. $\tau \in \Delta_{\Delta_{[k]}}$

2. For each $i \in [N]$, sample a topic $t_i \sim \mu$ and sample a word $x_i \sim A_{t_i}$ from the topic

For the reconstruction SSL task, we evenly split the document as $X = (\bar{X}_1, \bar{X}_2)$, where $\bar{X}_1$ and $\bar{X}_2$ denote the first and second halves of the document; note that $\bar{X}_1, \bar{X}_2 \in [V]^{N/2}$. We let X_1 and X_2 be the multiset of words in the two halves by using the normalized bag-of-words representation, i.e. $X_i = \frac{2}{N}\text{bag-of-words}(\bar{X}_i) \in \mathbb{R}^V,\ i \in \{1,2\}$[6]. The SSL task is to learn a representation $\psi \in \{\psi(\cdot) = \boldsymbol{B}\phi_1(\cdot); \boldsymbol{B} \in \mathbb{R}^{V \times V}\}$ that minimizes $\|\psi(X_1) - X_2\|^2$.

The downstream task is chosen to be a linear function of the topic posterior distribution μ for a given document X, i.e. $Y = w^\top \mathbb{E}[\mu|X] + N$, where N is 0 mean and σ^2-subgaussian. The error of a predictor $f : [V]^N \to \mathbb{R}$ is measured as $\mathbb{E}_{X,Y}\left[(f(X) - Y)^2\right]$, the optimal predictor being $f^*(X) = \mathbb{E}[Y \mid X]$.

A crucial property of topic model described above is that words in the document are sampled independently given the topic mixture μ, thus giving us the property: $X_1 \perp X_2 \mid \mu$. Although the cardinality of $\mu \in \Delta_{[k]}$ (that implicitly shows up in Theorem 4.4.4) is infinite, we can still show the benefit of SSL using our theoretical framework. We will show appropriate bounds for ϵ_{CI} and β, that show up in Theorem 4.4.4, using the topic model generative process.

Corollary 4.5.1. *Given a topic model characterized by (A, τ), suppose $\Gamma = \mathbb{E}_{\mu\sim\tau}\left[\mu\mu^\top\right]$ is the topic covariance matrix and let $\kappa = \frac{\lambda_{\max}(\Gamma)}{\lambda_{\min}(\Gamma)} < \infty$ be its condition number. Let ϵ_{CI} be the definition (2) from Definition 4.4.2 and β as defined in Assumption 4.4.1, then there exists a latent variable $\bar{Y} \in \bar{\mathcal{Y}}$ such that the following hold*

1. *$\bar{Y}$ takes k distinct values, i.e. $|\bar{\mathcal{Y}}| = k$*
2. *X_1 and X_1 are uncorrelated given $\bar{Y}$, which implies $\epsilon_{CI} = 0$.*
3. *$\mathbb{E}[Y|X_1]$ is a linear function of $\mathbb{E}[\bar{Y}|X_1]$*
4. *$\beta^{-1} \le \kappa\|w\|_2/\lambda_{\min}(A)$*

The proof is presented in Section 4.12.6. Thus the upper bound from Theorem 4.4.4 will look like $\tilde{\mathcal{O}}\left(\sigma^2\frac{k}{n_2} + \epsilon_{\text{pre}}^2\frac{\kappa\|w\|_2}{\lambda_{\min}(A)}\right)$, thus requiring only $\mathcal{O}(k)$ samples for the downstream task.

[6]We only need X_2 to be the bag-of-word representation, X_1 can be an ordered sentence.

4.6 Conditional distribution decomposition: SimSiam, CCA, ACE

In this section we establish the connection between SimSiam Chen and He [2021] and non-linear CCA between X_1 and X_2 and the alternating conditional expectation (ACE) algorithm. We show how our previous analysis can be extended to this setting and how the problem relates to decomposing the conditional distribution of $X_2 \mid X_1$.

4.6.1 Theoretical guarantees for non-linear CCA

In the previous sections, we used ψ to predict X_2 given X_1. As discussed in Remark Remark 4.11.1, we could have predicted $\eta(X_2)$ from X_1 for any function η, with all bounds depending on the function η. An alternative is to avoid choosing a specific η, but instead simultaneously learn an η that can be easily predicted from X_1. We further show how our problem setup and analysis can capture the popular method of SimSiam, an SSL method that does not use negative samples.

We first formulate the aforementioned problem and show that it corresponds to performing non-linear canonical component analysis (CCA) [Hardoon et al., 2004] on the joint distribution of (X_1, X_2). We let $L^2(X)$ denotes the Hilbert space of square integrable function with respect to the measure P_X, the marginal distribution of X. For instance, in our context of SSL, for a function $g : \mathbb{R}^{d_2} \to \mathbb{R}$, we denote $\|g\|^2_{L^2(X_2)} = \int g^2(x_2) dP_{X_2}(x_2)$ and thus $L^2(X_2) = \{g : \mathbb{R}^{d_2} \to \mathbb{R} \mid \|g\|^2_{L^2(X_2)} < \infty.\}$.

For zero-mean representation functions $\psi : \psi_i \in L^2(X_1), \eta : \eta_i \in L^2(X_2), i \in [k]$, we consider the generalized alternating conditional expectation (ACE) algorithm (Makur et al. [2015], Breiman and Friedman [1985], Buja [1990]) that optimizes the following:

$$\min_{\psi,\eta} L_{\text{ACE}}(\psi, \eta) := \mathbb{E}_{X_1,X_2}\left[\|\psi(X_1) - \eta(X_2)\|^2\right], \text{ s.t. } \boldsymbol{\Sigma}_{\psi,\psi} = \boldsymbol{\Sigma}_{\eta,\eta} = \boldsymbol{I}_k \tag{4.4}$$

Here $\boldsymbol{\Sigma}_{\psi,\psi} \in \mathbb{R}^{k\times k}$ and $(\boldsymbol{\Sigma}_{\psi,\psi})_{i,j} = \mathbb{E}_{X_1}[\psi_i(X_1)\psi_j(X_1)]$ and similarly for $\eta : \mathcal{X}_2 \to \mathbb{R}^k$. As we will show in Proposition 4.6.7, the above objective is equivalent to the following non-linear CCA:

$$\max_{\psi,\eta} L_{\text{CCA}}(\psi, \eta) := \mathbb{E}_{X_1,X_2}\left[\psi(X_1)^\top \eta(X_2)\right], \text{ s.t. } \boldsymbol{\Sigma}_{\psi,\psi} = \boldsymbol{\Sigma}_{\eta,\eta} = \boldsymbol{I}_k.$$

Connection to SimSiam: In the setting for the SimSiam Chen and He [2021] method, X_1 and X_2 are two randomly augmented images. The non-linear CCA problem is almost identical to SimSiam, except that

we use normalization of representation instead of stop-gradient to prevent representation collapse. CCA maximizes the inner product of the representations for each positive pairs (X_1, X_2) generated from their joint distribution. At the same time, the normalization constraint ensures that the representation doesn't collapse to trivial function, so we do not need negative samples. We now demonstrate how our previous analysis can easily apply to non-linear CCA.

Theorem 4.6.1 (General theorem for non-linear CCA)**.** *Let $\psi : \mathcal{X}_1 \to \mathbb{R}^k, \eta : \mathcal{X}_2 \to \mathbb{R}^k$ be the solution of Eqn. Equation* (4.4)*. Denote scalars $\sigma_i := \mathbb{E}_{X_1X_2}[\psi_i(X_1)\eta_i(X_2)]$. Then the approximation error of ψ satisfies:*

$$\begin{aligned} e_{apx}(\psi) &:= \min_{\boldsymbol{W}\in\mathbb{R}^{k\times k}} \mathbb{E}[\|f^*(X_1) - \boldsymbol{W}^\top\psi(X_1)\|^2] \\ &\le \sum_{y=1}^{k} \min_{g_y\in L^2(X_2)} 2(\|(\mathcal{T}_k - \mathcal{L})\circ g_y\|^2_{L^2(X_1)} + \|\mathcal{L}\circ g_y - f_y^*\|^2_{L^2(X_1)}). \end{aligned}$$

Here f^* is the optimal function to predict the one-hot encoder of Y with X_2, i.e., $f_y^*(x_1) = \mathbb{E}[1(Y = y)|X_1 = x_1] = P(Y = y|X_1 = x_1)$. Here $(\mathcal{T}_k \circ g_y)(x_1) := \sum_{i=1}^k \sigma_i \mathbb{E}[\eta_i(X_2)g_y(X_2)]\psi_i(x_1)$, and $(\mathcal{L}\circ g_y)(x_1) := \mathbb{E}_Y[\mathbb{E}_{X_2}[g_y(X_2)|Y]|X_1 = x_1]$.

The proof of this theorem and its corollaries below can be found in Section 4.13. With this theorem, we can apply different choices of g_y to derive the generalization bound. If we choose g_y such that $\mathbb{E}[g_y(X_2)|Y = y] = 1(Y = y)$, we get the following generalization bound:

Corollary 4.6.2 (Generalization bound with non-linear CCA.)**.** *In the same setting of Theorem 4.6.1, and suppose the learned ψ satisfies Assumption 4.3.4, then we have:*

$$ER_\psi(\hat{\boldsymbol{W}}) \le \tilde{O}\left(\frac{k\tilde{\epsilon}_{CI}^2}{\tilde{\lambda}^2} + \sigma^2\frac{k}{n_2}\right).$$

Here $\tilde{\epsilon}_{CI}^2 := \max_{\|g\|_{L^2(X_2)}=1} \mathbb{E}_{X_1}(\mathbb{E}[g(X_2)|X_1] - \mathbb{E}[\mathbb{E}[g(X_2)|Y]|X_1])^2$ is the measure of approximate conditional independence, and $\tilde{\lambda}$ is the $(k-1)$-th maximal correlation between X_2 and Y[7]*.*

Assumption 4.6.3 (α-Bayes error)**.** *We assume Y is almost deterministic when predicting from either X_1 or X_2. Specifically, there exists a classifier g_1^* such that $P_{X_1,Y}(g_1^*(x) \neq y) \le \alpha$; there exists g_2^* such that $P_{X_2,Y}(g_2^*(x) \neq y) \le \alpha$.*

[7]The definition and more discussion of maximal correlation between two random variable are deferred in Definition 4.6.8 and the next subsection.

If we choose $g_y(x_2) = 1(g_2^*(x_2) = y), \forall y \in [k]$ where $g_2^* := \mathbb{E}[Y|X_2]$ in Theorem 4.6.1, we get the following corollary:

Corollary 4.6.4 (Guarantees with small Bayes error)**.** *Under the same setting and algorithm as Corollary 4.6.2, if additionally we assume α-Bayes error (Assumption 4.6.3), we have that the generalization error also satisfies:*

$$ER_\psi(\hat{\boldsymbol{W}}) \leq \tilde{O}\left(\frac{\alpha}{1-\lambda} + \sigma^2 \frac{k}{n_2}\right),$$

where λ is the k-th maximal correlation between X_1 and X_2.

When the joint distribution of X_1, X_2 is non-degenerate, $\lambda < 1$. Therefore when Bayes error is small, the learned representation will yield a good downstream performance.

This corollary and the clustering setting is inspired by Theorem 3.7 in HaoChen et al. [2021], which showed a similar result for a spectral contrastive loss. Our corollary here shows that non-linear CCA achieves similar guarantees as spectral contrastive loss, without needing any negative samples.

Remark 4.6.5. *All the results in this section holds in the same way when replacing Y with the more fine-grained labels $\tilde{Y} = [Y, Z]$ as discussed in the previous section, and by replacing k by the cardinality of $\tilde{Y}$.*

4.6.2 Connection to ACE algorithm and maximal correlation

In this section, we review the variational formulation of our problem, and a closer look at the Breiman and Friedman's alternating conditional expectation (ACE) algorithm Makur et al. [2015], Breiman and Friedman [1985], Buja [1990]. Recall $L^2(X_1)$ and $L^2(X_2)$ are the square integrable function with respect to the marginal distribution of X_1 and X_2. We will understand the maximal correlation and the ACE algorithm on the operator $\mathcal{T} : L^2(X_2) \to L^2(X_1)$, where $(\mathcal{T} \circ g)(x_1) := \mathbb{E}[g(X_2)|X_1 = x_1]$ for any $g \in L^2(X_2)$. We will show that ACE algorithm decomposes the operator $\mathcal{T}$ and also implicitly defines the maximal correlation between the two random variables X_1 and X_2.

Due to Courant–Fischer–Weyl min-max principle, the top singular value of $\mathcal{T}$ can be computed by the variational problem

$$\max_{\|u\|_{L^2(X_1)}=1, \|v\|_{L^2(X_2)}=1} \left\{ \langle u, \mathcal{T} v \rangle \equiv \int p(x_1, x_2) u(x_1) v(x_2) dx_1 dx_2 \right\}.$$

The top k singular vectors of $\mathcal{T}$ can be computed by the variational problem

$$\{\psi_i\}_{i=1}^k, \{\eta_i\}_{i=1}^k \leftarrow \arg\max_{\psi,\eta} \left\{ \sum_{i=1}^k \int \langle \psi_i, \mathcal{T}\eta_i \rangle \equiv \mathbb{E}_{X_1,X_2} \left[\psi(X_1)^\top \eta(X_2) \right] \right\},$$
$$\text{s.t. } \mathbf{\Sigma}_{\psi,\psi} = \mathbf{\Sigma}_{\eta,\eta} = I_k. \tag{4.5}$$

Lemma 4.6.6. *ACE algorithm (Eqn. Equation* (4.5)*) with k-dimensional vector-valued functions solves the $(k+1)$-SVD of $\mathcal{T}$, and the top singular vectors of $\mathcal{T}$ is always achieved by constant functions $u(x_1) \equiv 1$ and $v(x_2) \equiv 1$.*

Proof. Observe that the top singular value $\sigma_1(\mathcal{T})$ is achieved by the top singular functions $u_1(x_1) = 1 \in L^2(X_1)$ and $v_1(x_2) = 1 \in L^2(X_2)$. The constraint $\mathbb{E}f(X_1) = 0$ corresponds to $\langle u_1, f \rangle_{X_1} = 0$, i.e., f being in the complement subspace of the top left singular vector of $\mathcal{T}$, and vice versa for X_2. By the Courant-Fischer characterization of singular values, ρ_1 is the variational problem corresponding to $\sigma_2(\mathcal{T})$. Similarly, ψ_k, η_k are the $(k+1)$-th singular vectors of $\mathcal{T}$ since they since $\rho_k = \langle \mathcal{T}\eta_k, \psi_k \rangle$. □

The second proposition shows that the variational form can be solved by the famous ACE algorithm of Breiman and Friedman Makur et al. [2015], Breiman and Friedman [1985], Buja [1990].

Proposition 4.6.7. *The generalized ACE algorithm solves Equation* (4.4)*, and is equivalent to the solution of non-linear CCA as in Equation* (4.5)*.*

Proof.

$$\begin{aligned}
&\mathbb{E} \sum_{i=1}^k (\eta_i(X_2) - \psi_i(X_1))^2 \\
=& \int_{x_1,x_2} p(x_1,x_2) \sum_{i=1}^k (\eta_i(x_2) - \psi_i(x_1))^2 \\
=& \sum_{i=1}^k \int_{x_1,x_2} (\eta_i^2(x_2) + \psi_i^2(x_1)) p(x_1,x_2) dx_1 dx_2 - 2 \sum_{i=1}^k \int_{x_1,x_2} p(x_1,x_2) \eta_i(x_2) \psi_i(x_1) dx_1 dx_2 \\
=& \sum_i \left(\mathbb{E}_{X_1}[\psi_i^2(X_1)] + E_{X_2}[\eta_i^2(X_2)] - 2\langle \psi_i, \mathcal{T}\eta_i \rangle \right)
\end{aligned}$$

$$=2k - 2\sum_{i=1}^{k}\langle \psi_i, \mathcal{T}\eta_i\rangle. \qquad \text{(Due to the orthogonality constraints)}$$

Therefore the solution of ACE is equivalent to that of non-linear CCA.

□

In summary, these two propositions show that calculating the SVD of $\mathcal{T}$ corresponds to conducting the alternating conditional expectation algorithm Makur et al. [2015], Breiman and Friedman [1985], Buja [1990].

Finally, the generalized maximal correlation between X_1 and X_2 is associated with the singular values of $\mathcal{T}$.

Definition 4.6.8 (k-th maximal correlation)**.** *For every $k \geq 1$, we define the k-th maximal correlation between X_1 and X_2 as:*

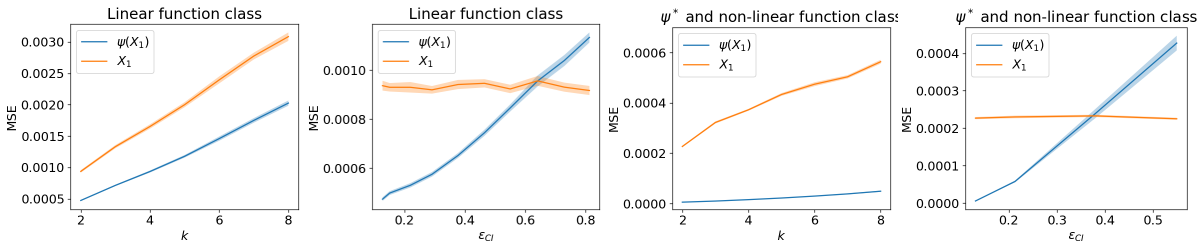
$$\lambda_k = \max_{f_i, g_i, i\in[k]} \min_{1\leq i\leq k} \mathbb{E}[f_i(X_1)g_i(X_2)],$$
$$s.t.\ \boldsymbol{\Sigma}_{f,f} = \boldsymbol{I}, \boldsymbol{\Sigma}_{g,g} = \boldsymbol{I}, \mathbb{E}[f_i(X_1)] = 0, \mathbb{E}[g_i(X_2)] = 0.$$

As shown in Propostion 3 and Theorem 2 of Makur et al. [2015], the k-th maximal correlation is the $(k+1)$-th singular value of $\mathcal{T}$ and therefore can be calculated from the ACE algorithm: $\lambda_k = \mathbb{E}[\psi_k(X_1)\eta_k(X_2)]$ when ψ, η solves Eq. Equation (4.4). One can also refer to Makur et al. [2015] for more geometric interpretation for the maximal correlation between two random variables.

4.7 Experiments

In this section, we empirically verify our claim that SSL performs well when ACI is satisfied. More details for experiments can be found in Section 4.17, including experiments in the text domain.

Simulations. With synthetic data, we verify how excess risk (ER) scales with the cardinality/feature dimension of $\mathcal{Y}$ (k), and ACI (ϵ_{CI} in Definition 4.4.2). We consider a mixture of Gaussian data and conduct experiments with both linear function space ($\mathcal{H}_1$ with ϕ_1 as identity map) and universal function space



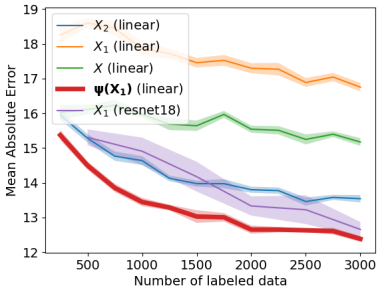
(a) (b) (c) (d)

Figure 4.1: **Left two**: how MSE scales with k (the dimension of Y) and ϵ_{CI} (Definition 4.4.2) with the linear function class. **Right two**: how MSE scales with k and ϵ with ψ^* and non-linear function class. Mean of 30 trials are shown in solid line and one standard error is shown by shadow.

$\mathcal{H}_u$. We sample the label Y uniformly from $\{1, ..., k\}$. For i-th class, the centers $\mu_{1i} \in \mathbb{R}^{d_1}$ and $\mu_{2i} \in \mathbb{R}^{d_2}$ are uniformly sampled from $[0, 10)$. Given $Y = i$, $\alpha \in [0, 1]$, let $X_1 \sim \mathcal{N}(\mu_{1i}, \mathbf{I})$, $\hat{X}_2 \sim \mathcal{N}(\mu_{2i}, \mathbf{I})$, and $X_2 = (1 - \alpha)\hat{X}_2 + \alpha X_1$. Therefore α is a correlation coefficient: $\alpha = 0$ ensures X_2 being CI with X_1 given Y and when $\alpha = 1$, X_2 fully depends on X_1. (if $d_1 \neq d_2$, we append zeros or truncate to fit accordingly).

We first conduct experiments with linear function class. We learn a linear representation ψ with n_1 samples and the linear prediction of Y from ψ with n_2 samples. We set $d_1 = 50$, $d_2 = 40$, $n_1 = 4000$, $n_2 = 1000$ and ER is measured with Mean Squared Error (MSE). As shown in Figure Figure 4.1(a)(b), the MSE of learning with $\psi(X_1)$ scales linearly with k as indicated in Theorem 4.3.10, and scales linearly with ϵ_{CI} associated with linear function class as indicated in Theorem 4.4.4. Next we move on to general function class, i.e., $\psi^* = \mathbb{E}[Y|X_1]$ with a closed form solution (see example Example 4.3.3). We use the same parameter settings as above. For baseline method, we use kernel linear regression to predict Y using X_1 (we use RBF kernel which also has universal approximation power). As shown in Figure Figure 4.1(c)(d), the phenomenon is the same as what we observe in the linear function class setting, and hence they respectively verify Theorem 4.3.5 and Theorem 4.4.4 with $\mathcal{H}_u$.

Computer Vision Task. We verify if learning from ψ is more effective than learning directly from X_1, in a realistic setting (without enforcing conditional independence). Specifically, we test on the Yearbook dataset [Ginosar et al., 2015], and try to predict the date when the portraits are taken (denoted as Y_D), which ranges from 1905 to 2013. We resize all the portraits to be 128 by 128. We crop out the center 64 by 64 pixels (the face), and treat it as X_2, and treat the outer rim as X_1 as shown in Figure Figure 4.2. Our task is to predict Y_D, which is the year when the portraits are taken, and the year ranges from 1905 to 2013. For ψ, we learn X_2 from X_1 with standard image inpainting techniques [Pathak et al., 2016], and full set of training



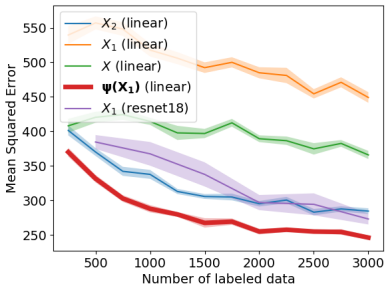


Figure 4.2: **Left**: Example of the X_2 (in the red box of the 1st row), the X_1 (out of the red box of the 1st row), the input to the inpainting task (the second row), $\psi(X_1)$ (the 3 row in the red box), and in this example $Y = 1967$. **Middle**: Mean Squared Error comparison of yearbook regression predicting dates. **Right**: Mean Absolute Error comparison of yearbook regression predicting dates. Experiments are repeated 10 times, with mean shown in solid line and one standard deviation in shadow.

data (without labels). After that we fix the learned ψ and learn a linear model to predict Y_D from ψ using a smaller set of data (with labels). Besides linear model on X_1, another strong baseline that we compare with is using ResNet18 [He et al., 2016] to predict Y_D from X_1. With the full set of training data, this model is able to achieve a Mean Absolute Difference of 6.89, close to what state-of-the-art can achieve [Ginosar et al., 2015]. ResNet18 has similar amount of parameters as our generator, and hence roughly in the same function class. We show the MSE result as in Figure Figure 4.2. Learning from ψ is more effective than learning from X_1 or X_2 directly, with linear model as well as with ResNet18. Practitioner usually fine-tune ψ with the downstream task, which leads to more competitive performance [Pathak et al., 2016].

4.8 Conclusion

In this work we theoretically quantify how an approximate conditional independence assumption that connects pretext and downstream task data distributions can give sample complexity benefits of self-supervised learning on downstream tasks. Our theoretical findings are also supported by experiments on simulated data and also on real CV and NLP tasks. We would like to note that approximate CI is only a sufficient condition for a useful pretext task. We leave it for future work to investigate other mechanisms by which pretext tasks help with downstream tasks.

4.9 Some useful facts

4.9.1 Relation of inverse covariance matrix and partial correlation

For a covariance matrix of joint distribution for variables X, Y, the covariance matrix is

$$\begin{bmatrix} \mathbf{\Sigma}_{XX} & \mathbf{\Sigma}_{XY} \\ \mathbf{\Sigma}_{YX} & \mathbf{\Sigma}_{YY} \end{bmatrix} = \begin{bmatrix} \mathbf{\Sigma}_{X_1X_1} & \mathbf{\Sigma}_{X_1X_2} & \mathbf{\Sigma}_{X_1Y} \\ \mathbf{\Sigma}_{X_2X_1} & \mathbf{\Sigma}_{X_2X_2} & \mathbf{\Sigma}_{X_2Y} \\ \mathbf{\Sigma}_{YX_1} & \mathbf{\Sigma}_{X_2Y} & \mathbf{\Sigma}_{YY} \end{bmatrix}.$$

Its inverse matrix $\mathbf{\Sigma}^{-1}$ satisfies

$$\mathbf{\Sigma}^{-1} = \begin{bmatrix} \boldsymbol{A} & \rho \\ \rho^\top & \boldsymbol{B} \end{bmatrix}.$$

Here $\boldsymbol{A}^{-1} = \mathbf{\Sigma}_{XX} - \mathbf{\Sigma}_{XY}\mathbf{\Sigma}_{YY}^{-1}\mathbf{\Sigma}_{YX} \equiv \mathrm{cov}(X - \mathbb{E}^L[X|Y], X - \mathbb{E}^L[X|Y]) := \mathbf{\Sigma}_{XX\cdot Y}$, the partial covariance matrix of X given Y.

4.9.2 Relation to conditional independence

Proof of Lemma 4.12.6.

Fact 4.9.1. *When $X_1 \perp X_2 | Y$, the partial covariance between X_1, X_2 given Y is 0:*

$$\begin{aligned} \mathbf{\Sigma}_{X_1X_2\cdot Y} :=& \mathrm{cov}(X_1 - \mathbb{E}^L[X_1|Y], X_2 - \mathbb{E}^L[X_2|Y]) \\ \equiv& \mathbf{\Sigma}_{X_1X_2} - \mathbf{\Sigma}_{X_1Y}\mathbf{\Sigma}_{YY}^{-1}\mathbf{\Sigma}_{YX_2} = 0. \end{aligned}$$

The derivation comes from the following:

Lemma 4.9.2 (Conditional independence (Adapted from Huang [2010]))**.** *For random variables X_1, X_2 and a random variable Y with finite values, conditional independence $X_1 \perp X_2 | Y$ is equivalent to:*

$$\sup_{f \in N_1, g \in N_2} \mathbb{E}[f(X_1)g(X_2)|Y] = 0. \tag{4.6}$$

Here $N_i = \{f : \mathbb{R}^{d_i} \to R : E[f(X_i)|Y] = 0\}$, $i = 1, 2$.

Notice for arbitrary function f, $\mathbb{E}[f(X)|Y] = \mathbb{E}^L[f(X)|\phi_y(Y)]$ with one-hot encoding of discrete variable Y. Therefore for any feature map we can also get that conditional independence ensures:

$$\begin{aligned}\boldsymbol{\Sigma}_{\phi_1(X_1)\phi_2(X_2)|Y} :=& \mathrm{cov}(\phi_1(X_1) - \mathbb{E}^L[\phi_1(X_1)|\phi_y(Y)], \phi_2(X_2) - \mathbb{E}^L[\phi_2(X_2)|\phi_y(Y)]) \\ =& \mathbb{E}[\bar{\phi}_1(X_1)\bar{\phi}_2(X_2)^\top] = 0.\end{aligned}$$

Here $\bar{\phi}_1(X_1) = \phi_1(X_1) - \mathbb{E}[\phi_1(X_1)|\phi_y(Y)]$ is mean zero given Y, and vice versa for $\bar{\phi}_2(X_2)$. This thus finishes the proof for Lemma 4.12.6. □

4.9.3 Technical facts for matrix concentration

We include this covariance concentration result that is adapted from Claim A.2 in Du et al. [2020]:

Claim 4.9.3 (covariance concentration for gaussian variables)**.** *Let $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots \boldsymbol{x}_n]^\top \in \mathbb{R}^{n\times d}$ where each $x_i \sim \mathcal{N}(0, \boldsymbol{\Sigma}_X)$. Suppose $n \gg k + \log(1/\delta)$ for $\delta \in (0,1)$. Then for any given matrix $B \in \mathbb{R}^{d\times m}$ that is of rank k and is independent of $\boldsymbol{X}$, with probability at least $1 - \frac{\delta}{10}$ over $\boldsymbol{X}$ we have*

$$0.9\boldsymbol{B}^\top\boldsymbol{\Sigma}_X\boldsymbol{B} \preceq \frac{1}{n}\boldsymbol{B}^\top\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{B} \preceq 1.1\boldsymbol{B}^\top\boldsymbol{\Sigma}_X\boldsymbol{B}. \tag{4.7}$$

And we will also use Claim A.2 from Du et al. [2020] for concentrating subgaussian random variable.

Claim 4.9.4 (covariance concentration for subgaussian variables)**.** *Let $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots \boldsymbol{x}_n]^\top \in \mathbb{R}^{n\times d}$ where each $\boldsymbol{x}_i$ is ρ^2-sub-gaussian. Suppose $n \gg \rho^4(k + \log(1/\delta))$ for $\delta \in (0,1)$. Then for any given matrix $B \in \mathbb{R}^{d\times m}$ that is of rank k and is independent of $\boldsymbol{X}$, with probability at least $1 - \frac{\delta}{10}$ over $\boldsymbol{X}$ we have*

$$0.9\boldsymbol{B}^\top\boldsymbol{\Sigma}_X\boldsymbol{B} \preceq \frac{1}{n}\boldsymbol{B}^\top\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{B} \preceq 1.1\boldsymbol{B}^\top\boldsymbol{\Sigma}_X\boldsymbol{B}. \tag{4.8}$$

Claim 4.9.5. *Let $\boldsymbol{Z} \in \mathbb{R}^{n\times k}$ be a matrix with row vectors sampled from i.i.d Gaussian distribution $\mathcal{N}(0, \boldsymbol{\Sigma}_Z)$. Let $\mathbf{P} \in \mathbb{R}^{n\times n}$ be a fixed projection onto a space of dimension d. Then with a fixed $\delta \in (0,1)$, we have:*

$$\|\mathbf{P}\boldsymbol{Z}\|_F^2 \lesssim \mathrm{Tr}(\boldsymbol{\Sigma}_Z)(d + \log(k/\delta)),$$

with probability at least $1 - \delta$.

Claim Claim 4.9.5. Each t-th column of Z is an n-dim vector that is i.i.d sampled from Gaussian distribution $\mathcal{N}(0, \mathbf{\Sigma}_{tt})$.

$$\begin{aligned}\|\mathbf{P}\boldsymbol{Z}\|_F^2 &= \sum_{t=1}^{k} \|\mathbf{P}\boldsymbol{z}_t\|^2 \\ &= \sum_{t=1}^{k} \boldsymbol{z}_t^\top \mathbf{P}\boldsymbol{z}_t.\end{aligned}$$

Each term satisfy $\mathbf{\Sigma}_{kk}^{-1}\|\mathbf{P}\boldsymbol{z}_t\|^2 \sim \chi^2(d)$, and therefore with probability at least $1-\delta'$ over $\boldsymbol{z}_t$,

$$\mathbf{\Sigma}_{kk}^{-1}\|\mathbf{P}\boldsymbol{z}_t\|^2 \lesssim d + \log(1/\delta').$$

Using union bound, take $\delta' = \delta/k$ and summing over $t \in [k]$ we get:

$$\|\mathbf{P}\boldsymbol{Z}\|_F^2 \lesssim \mathrm{Tr}(\mathbf{\Sigma}_Z)(d + \log(k/\delta)).$$

□

Theorem 4.9.6 (Vector Bernstein Inequality (Theorem 12 in Gross [2011]))**.** *Let $X_1, \cdots, X_m$ be independent zero-mean vector-valued random variables. Let*

$$N = \|\sum_{i=1}^{m} X_i\|_2.$$

Then

$$\mathbb{P}[N \geq \sqrt{V} + t] \leq \exp\left(\frac{-t^2}{4V}\right),$$

where $V = \sum_i \mathbb{E}\|X_i\|_2^2$ and $t \leq V/(\max \|X_i\|_2)$.

Lemma 4.9.7. *Let $\boldsymbol{Z} \in \mathbb{R}^{n\times d}$ be a matrix whose row vectors are n independent mean-zero (conditional on $\mathbf{P}$ being a rank-d projection matrix) σ-sub-Gaussian random vectors. With probability $1-\delta$:*

$$\|\mathbf{P}\boldsymbol{Z}\|_F^2 \lesssim \sigma^2(d + \log(d/\delta)).$$

Proof of Lemma 4.9.7. Write $\mathbf{P} = \boldsymbol{U}\boldsymbol{U}^\top = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_d]$ where $\boldsymbol{U}$ is orthogonal matrix in $\mathbb{R}^{n\times d}$ where

$\boldsymbol{U}^\top \boldsymbol{U} = I$. Notice $\|\boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{Z}\|_F^2 = \mathrm{Tr}(\boldsymbol{Z}^\top UU^\top UU^\top \boldsymbol{Z}) = \mathrm{Tr}(\boldsymbol{Z}^\top UU^\top \boldsymbol{Z})$. Therefore:

$$\begin{aligned}\|\mathbf{P}\boldsymbol{Z}\|_F^2 =& \|\boldsymbol{U}^\top \boldsymbol{Z}\|_F^2 \\ =& \sum_{j=1}^{d} \|\boldsymbol{u}_j^\top \boldsymbol{Z}\|^2 \\ =& \sum_{j=1}^{d} \|\sum_{i=1}^{n} \boldsymbol{u}_{ji}\boldsymbol{z}_i\|^2,\end{aligned}$$

where each $\boldsymbol{z}_i \in \mathbb{R}^k$ being the i-th row of $\boldsymbol{Z}$ is a centered independent σ sub-Gaussian random vectors. To use vector Bernstein inequality, we let $X := \sum_{i=1}^n X_i$ with X_i taking the value of $\boldsymbol{u}_{ji}\boldsymbol{z}_i$. We have X_i is zero mean: $\mathbb{E}[X_i] = \mathbb{E}[\boldsymbol{u}_{ji}\mathbb{E}[\boldsymbol{z}_i|\boldsymbol{u}_{ji}]] = \mathbb{E}[\boldsymbol{u}_{ji} \cdot 0] = 0$.

$$\begin{aligned}V :=& \sum_i \mathbb{E}\|X_i\|_2^2 \\ =& \sum_i \mathbb{E}[\boldsymbol{u}_{ji}^2 \boldsymbol{z}_i^\top \boldsymbol{z}_i] \\ =& \sum_i \mathbb{E}_{\boldsymbol{u}_{ji}}[\boldsymbol{u}_{ji}^2 \mathbb{E}[\|\boldsymbol{z}_i\|_2^2|\boldsymbol{u}_{ji}]] \\ \leq& \sigma^2 \sum_i \mathbb{E}_{\boldsymbol{u}_{ji}}[\boldsymbol{u}_{ji}^2] \\ =& \sigma^2.\end{aligned}$$

Therefore by vector Bernstein Inequality, with probability at least $1 - \delta/d$, $\|X\| \leq \sigma(1 + \sqrt{\log(d/\delta)})$. Then by taking union bound, we get that $\|\mathbf{P}\boldsymbol{Z}\|^2 = \sum_{j=1}^d \|\boldsymbol{u}_j^\top \boldsymbol{Z}\|^2 \lesssim \sigma^2 d(1 + \log(d/\delta))$ with probability $1 - \delta$.

□

4.10 Warm-up: jointly Gaussian variables

We assume X_1, X_2, Y are jointly Gaussian, and so the optimal regression functions are all linear, i.e., $\mathbb{E}[Y|X_1] = \mathbb{E}^L[Y|X_1]$. We also assume data is centered: $\mathbb{E}[X_i] = 0$ and $\mathbb{E}[Y] = 0$. Non-centered data can easily be handled by learning an intercept. All relationships between random variables can then be captured by the (partial) covariance matrix. Therefore it is easy to quantify the CI property and establish the necessary and sufficient conditions that make X_2 a reasonable pretext task.

Assumption 4.10.1 (Jointly Gaussian)**.** *X_1, X_2, Y are jointly Gaussian.*

Assumption 4.10.2 (Conditional independence)**.** *$X_1 \perp X_2 | Y$.*

Claim 4.10.3 (Closed-form solution)**.** *Under Assumption 4.10.1, the representation function and optimal prediction that minimize the population risk can be expressed as follows:*

$$\psi^*(\boldsymbol{x}_1) := \mathbb{E}^L[X_2|X_1 = \boldsymbol{x}_1] = \boldsymbol{\Sigma}_{X_2X_1}\boldsymbol{\Sigma}_{X_1X_1}^{-1}\boldsymbol{x}_1 \tag{4.9}$$

$$\textit{Our target } f^*(\boldsymbol{x}_1) := \mathbb{E}^L[Y|X_1 = \boldsymbol{x}_1] = \boldsymbol{\Sigma}_{YX_1}\boldsymbol{\Sigma}_{X_1X_1}^{-1}\boldsymbol{x}_1. \tag{4.10}$$

Our prediction for downstream task with representation ψ^* will be: $g(\cdot) := \mathbb{E}^L[Y|\psi^*(X_1)]$. Recall from Equation (4.1) that the partial covariance matrix between X_1 and X_2 given Y is $\boldsymbol{\Sigma}_{X_1X_2|Y} \equiv \boldsymbol{\Sigma}_{X_1X_2} - \boldsymbol{\Sigma}_{X_1Y}\boldsymbol{\Sigma}_{YY}^{-1}\boldsymbol{\Sigma}_{YX_2}$. This partial covariance matrix captures the correlation between X_1 and X_2 given Y. For jointly Gaussian random variables, CI is equivalent to $\boldsymbol{\Sigma}_{X_1X_2|Y} = 0$. We first analyze the approximation error based on the property of this partial covariance matrix.

Lemma 4.10.4 (Approximation error)**.** *Under Assumption 4.10.1, Assumption 4.10.2, if $\boldsymbol{\Sigma}_{X_2Y}$ has rank k, we have $f^*(\boldsymbol{x}_1) \equiv \boldsymbol{W}^*\psi^*(\boldsymbol{x}_1)$, i.e., $e_{apx}(\psi^*) = 0$.*

Remark 4.10.5. *$\boldsymbol{\Sigma}_{X_2Y}$ being full column rank implies that $\mathbb{E}[X_2|Y]$ has rank k, i.e., X_2 depends on all directions of Y and thus captures all directions of information of Y. This is a necessary assumption for X_2 to be a reasonable pretext task for predicting Y. $e_{apx}(\psi^*) = 0$ means f^* is linear in ψ^*. Therefore ψ^* selects d_2 out of d_1 features that are sufficient to predict Y.*

Next we consider the estimation error that characterizes the number of samples needed to learn a prediction function $f(\boldsymbol{x}_1) = \hat{\boldsymbol{W}}\psi^*(\boldsymbol{x}_1)$ that generalizes.

Theorem 4.10.6 (Excess risk)**.** *Fix a failure probability $\delta \in (0, 1)$. Under Assumption 4.10.1,Assumption 4.10.2, if $n_2 \gg k + \log(1/\delta)$, excess risk of the learned predictor $\boldsymbol{x}_1 \to \hat{\boldsymbol{W}}\psi^*(\boldsymbol{x}_1)$ on the target task satisfies*

$$\mathrm{ER}_{\psi^*}(\hat{\boldsymbol{W}}) \leq \mathcal{O}\left(\frac{\mathrm{Tr}(\boldsymbol{\Sigma}_{YY|X_1})(k + \log(k/\delta))}{n_2}\right),$$

with probability at least $1 - \delta$.

Here $\boldsymbol{\Sigma}_{YY|X_1} \equiv \boldsymbol{\Sigma}_{YY} - \boldsymbol{\Sigma}_{YX_1}\boldsymbol{\Sigma}_{X_1X_1}^{-1}\boldsymbol{\Sigma}_{X_1Y}$ captures the noise level and is the covariance matrix of the residual

term $Y - f^*(X_1) = Y - \mathbf{\Sigma}_{YX_1}\mathbf{\Sigma}_{X_1X_1}^{-1}X_1$. Compared to directly using X_1 to predict Y, self-supervised learning reduces the sample complexity from $\tilde{\mathcal{O}}(d_1)$ to $\tilde{\mathcal{O}}(k)$. We generalize these results even when only a weaker form of CI holds.

Assumption 4.10.7 (Conditional independence given latent variables)**.** *There exists some latent variable $Z \in \mathbb{R}^m$ such that $X_1 \perp X_2 | \bar{Y}$, and $\mathbf{\Sigma}_{X_2\bar{Y}}$ is of rank $k+m$, where $\bar{Y} = [Y, Z]$.*

This assumption lets introduce some reasonable latent variables that capture the information between X_1 and X_2 apart from Y. $\mathbf{\Sigma}_{X_2\bar{Y}}$ being full rank says that all directions of $\bar{Y}$ are needed to predict X_2, and therefore Z is not redundant. For instance, when $Z = X_1$, the assumption is trivially true but Z is not the minimal latent information we want to add. Note it implicitly requires $d_2 \geq k + m$.

Corollary 4.10.8. *Under Assumption 4.10.1, Assumption 4.10.7, we have $f^*(\boldsymbol{x}_1) \equiv \boldsymbol{W}^*\psi^*(\boldsymbol{x}_1)$, i.e., the approximation error $e_{apx}(\psi^*)$ is 0. We can also generalize Theorem 4.10.6 by replacing k by $k+m$.*

4.11 Omitted proofs with conditional independence

Proof of Lemma 4.10.4.

$$\operatorname{cov}(X_1|Y, X_2|Y) = \mathbf{\Sigma}_{X_1X_2} - \mathbf{\Sigma}_{X_1Y}\mathbf{\Sigma}_{YY}^{-1}\mathbf{\Sigma}_{YX_2} = 0.$$

By plugging it into the expression of $\mathbb{E}^L[X_2|X_1]$, we get that

$$\begin{aligned}\psi(x_1) := \mathbb{E}^L[X_2|X_1 = x_1] &= \mathbf{\Sigma}_{X_2X_1}\mathbf{\Sigma}_{X_1X_1}^{-1}x_1 \\ &= \mathbf{\Sigma}_{X_2Y}\mathbf{\Sigma}_{YY}^{-1}\mathbf{\Sigma}_{YX_1}\mathbf{\Sigma}_{X_1X_1}^{-1}x_1 \\ &= \mathbf{\Sigma}_{X_2Y}\mathbf{\Sigma}_{YY}^{-1}\mathbb{E}^L[Y|X_1].\end{aligned}$$

Therefore, as long as $\mathbf{\Sigma}_{X_2Y}$ is rank k, it has left inverse matrix and we get: $\mathbb{E}^L[Y|X_1 = x_1] = \mathbf{\Sigma}_{X_2Y}^{\dagger}\mathbf{\Sigma}_{YY}\psi(x_1)$. Therefore there's no approximation error in using ψ to predict Y.

□

Proof of Corollary 4.10.8. Let selector operator $\boldsymbol{S}_y$ be the mapping such that $\boldsymbol{S}_y\bar{Y} = Y$, we overload it as the matrix that ensure $\boldsymbol{S}_y\mathbf{\Sigma}_{\bar{Y}X} = \mathbf{\Sigma}_{YX}$ for any random variable X as well.

From Lemma 4.10.4 we get that there exists W such that $\mathbb{E}^L[\bar{Y}|X_1] = \boldsymbol{W}\mathbb{E}^L[X_2|X_1]$, just plugging in $\boldsymbol{S}_y$ we get that $\mathbb{E}^L[Y|X_1] = (\boldsymbol{S}_y\boldsymbol{W})\mathbb{E}^L[X_2|X_1]$.

□

Proof of Theorem 4.10.6. Write $f^*(X_1) = \mathbb{E}[Y|X_1] = (\boldsymbol{A}^*)^\top X_1$. $\mathbb{E}^L[Y|X_1 = x_1] = \boldsymbol{\Sigma}^\dagger_{X_2Y}\boldsymbol{\Sigma}_{YY}\psi(x_1)$. Let $\boldsymbol{W}^* = \boldsymbol{\Sigma}_{YY}\boldsymbol{\Sigma}^\dagger_{YX_2}$. From Lemma 4.10.4 we know $f^* = \boldsymbol{W}^*\psi$. Recall noise $N = Y - f^*(X_1)$ is mean zero conditional on X_1. We write $\boldsymbol{N} = \boldsymbol{Y} - f^*(\boldsymbol{X}_1)$.

First we have the basic inequality,

$$\begin{aligned}\frac{1}{2n_2}\|\boldsymbol{Y} - \psi(\boldsymbol{X}_1)\hat{\boldsymbol{W}}\|_F^2 \leq& \frac{1}{2n_2}\|\boldsymbol{Y} - \boldsymbol{X}_1A^*\|_F^2\\ =&\frac{1}{2n_2}\|\boldsymbol{Y} - \psi(\boldsymbol{X}_1)\boldsymbol{W}^*\|_F^2 = \frac{1}{2n_2}\|\boldsymbol{N}\|_F^2.\end{aligned}$$

Therefore by rearranging both sides, we have:

$$\begin{aligned}\|\psi(\boldsymbol{X}_1)\boldsymbol{W}^* - \psi(\boldsymbol{X}_1)\hat{\boldsymbol{W}}\|^2 \leq& 2\langle \boldsymbol{N}, \psi(\boldsymbol{X}_1)\boldsymbol{W}^* - \psi(\boldsymbol{X}_1)\hat{\boldsymbol{W}}\rangle\\ =&2\langle P_{\psi(\boldsymbol{X}_1)}\boldsymbol{N}, \psi(\boldsymbol{X}_1)\boldsymbol{W}^* - \psi(\boldsymbol{X}_1)\hat{W}\rangle\\ \leq&2\|P_{\psi(\boldsymbol{X}_1)}\boldsymbol{N}\|_F\|\psi(\boldsymbol{X}_1)\boldsymbol{W}^* - \psi(\boldsymbol{X}_1)\hat{W}\|_F\\ \Rightarrow \|\psi(\boldsymbol{X}_1)\boldsymbol{W}^* - \psi(\boldsymbol{X}_1)\hat{\boldsymbol{W}}\| \leq& 2\|P_{\psi(\boldsymbol{X}_1)}\boldsymbol{N}\|_F\\ \lesssim&\sqrt{\mathrm{Tr}(\boldsymbol{\Sigma}_{YY|X_1})(k + \log k/\delta)}. \qquad \text{(from Claim Claim 4.9.5)}\end{aligned}$$

The last inequality is derived from Claim Claim 4.9.5 and the fact that each row of $\boldsymbol{N}$ follows gaussian distribution $\mathcal{N}(0, \boldsymbol{\Sigma}_{YY|X_1})$. Therefore

$$\frac{1}{n_2}\|\psi(\boldsymbol{X}_1)W^* - \psi(\boldsymbol{X}_1)\hat{W}\|_F^2 \lesssim \frac{\mathrm{Tr}(\boldsymbol{\Sigma}_{YY|X_1})(k + \log k/\delta)}{n_2}.$$

Next we need to concentrate $1/n\boldsymbol{X}_1^\top\boldsymbol{X}_1$ to $\boldsymbol{\Sigma}_X$. Suppose $\mathbb{E}^L[X_2|X_1] = \boldsymbol{B}^\top X_1$, i.e., $\psi(x_1) = \boldsymbol{B}^\top x_1$, and $\psi(\boldsymbol{X}_1) = \boldsymbol{X}_1\boldsymbol{B}$. With Claim Claim 4.9.3 we have $1/n\psi(\boldsymbol{X}_1)^\top\psi(\boldsymbol{X}_1) = 1/n\boldsymbol{B}^\top\boldsymbol{X}_1^\top\boldsymbol{X}_1\boldsymbol{B}$ satisfies:

$$0.9\boldsymbol{B}^\top\boldsymbol{\Sigma}_X\boldsymbol{B} \preceq 1/n_2\psi(\boldsymbol{X}_1)^\top\psi(\boldsymbol{X}_1) \preceq 1.1\boldsymbol{B}^\top\boldsymbol{\Sigma}_X\boldsymbol{B}$$

Therefore we also have:

$$\begin{aligned}
&\mathbb{E}[\|(\boldsymbol{W}^* - \hat{\boldsymbol{W}})^\top \psi(x_1)\|^2] \\
=&\|\boldsymbol{\Sigma}_X^{1/2}\boldsymbol{B}(\boldsymbol{W}^* - \hat{\boldsymbol{W}})\|_F^2 \\
\leq&\frac{1}{0.9n_2}\|\psi(\boldsymbol{X}_1)\boldsymbol{W}^* - \psi(\boldsymbol{X}_1)\hat{\boldsymbol{W}}\|_F^2 \lesssim \frac{\mathrm{Tr}(\boldsymbol{\Sigma}_{YY|X_1})(k + \log k/\delta)}{n_2}.
\end{aligned}$$

□

4.11.1 Omitted proof for general random variables

Proof of Lemma 4.3.2. Let the representation function ψ be defined as:

$$\begin{aligned}
\psi(\cdot) := \mathbb{E}[X_2|X_1] =&\mathbb{E}[\mathbb{E}[X_2|X_1, Y]|X_1] \\
=&\mathbb{E}[\mathbb{E}[X_2|Y]|X_1] && \text{(uses CI)} \\
=&\sum_y P(Y = y|X_1)\mathbb{E}[X_2|Y = y] \\
=:&f(X_1)^\top \boldsymbol{A},
\end{aligned}$$

where $f : \mathbb{R}^{d_1} \to \Delta_{\mathcal{Y}}$ satisfies $f(x_1)_y = P(Y = y|X_1 = x_1)$, and $\boldsymbol{A} \in \mathbb{R}^{\mathcal{Y}\times d_2}$ satisfies $\boldsymbol{A}_{y,:} = \mathbb{E}[X_2|Y = y]$. Here Δ_d denotes simplex of dimension d, which represents the discrete probability density over support of size d.

Let $\boldsymbol{B} = \boldsymbol{A}^\dagger \in \mathbb{R}^{\mathcal{Y}\times d_2}$ be the pseudoinverse of matrix A, and we get $\boldsymbol{B}\boldsymbol{A} = \boldsymbol{I}$ from our assumption that $\boldsymbol{A}$ is of rank $|\mathcal{Y}|$. Therefore $f(x_1) = \boldsymbol{B}\psi(x_1), \forall x_1$. Next we have:

$$\begin{aligned}
\mathbb{E}[Y|X_1 = x_1] =&\sum_y P(Y = y|X_1 = x_1) \times y \\
=&\boldsymbol{Y} f(x_1) \\
=&(\boldsymbol{Y}\boldsymbol{B}) \cdot \psi(X_1).
\end{aligned}$$

Here we denote by $\boldsymbol{Y} \in \mathbb{R}^{k\times\mathcal{Y}}, \boldsymbol{Y}_{:,y} = y$ that spans the whole support $\mathcal{Y}$. Therefore let $\boldsymbol{W}^* = \boldsymbol{Y}\boldsymbol{B}$ will finish the proof.

□

Proof of Theorem 4.3.5. With Lemma 4.3.2 we know $e_{\text{apx}} = 0$, and therefore $\boldsymbol{W}^*\psi(X_1) \equiv f^*(X_1)$. Next from basic inequality and the same proof as in Theorem 4.10.6 we have:

$$\|\psi(\boldsymbol{X}_1)\boldsymbol{W}^* - \psi(\boldsymbol{X}_1)\hat{\boldsymbol{W}}\| \leq 2\|\mathbf{P}_{\psi(\boldsymbol{X}_1)}\boldsymbol{N}\|_F$$

Notice $\mathcal{N}$ is a random noise matrix whose row vectors are independent samples from some centered distribution. Note we assumed $\mathbb{E}[\|N\|^2|\boldsymbol{X}_1] \leq \sigma^2$. $P_{\psi(\boldsymbol{X}_1)}$ is a projection to dimension k. From Lemma 4.9.7 we have:

$$\|f^*(\boldsymbol{X}_1) - \psi(\boldsymbol{X}_1)\hat{\boldsymbol{W}}\| \leq \sigma\sqrt{k(1+\log k/\delta)}.$$

Next, with Claim Claim 4.9.4 we have when $n \gg \rho^4(k + \log(1/\delta))$, since $\boldsymbol{W}^* - \hat{\boldsymbol{W}} \in \mathbb{R}^{d_2\times k}$,

$$\begin{aligned}
&0.9(\boldsymbol{W}^* - \hat{\boldsymbol{W}})^\top\boldsymbol{\Sigma}_\psi(\boldsymbol{W}^* - \hat{\boldsymbol{W}}) \\
\preceq&\frac{1}{n_2}(\boldsymbol{W}^* - \hat{\boldsymbol{W}})^\top\sum_i \psi(x_1^{(i)})\psi(x_1^{(i)})^\top(\boldsymbol{W}^* - \hat{\boldsymbol{W}}) \preceq 1.1(\boldsymbol{W}^* - \hat{\boldsymbol{W}})^\top\boldsymbol{\Sigma}_\psi(\boldsymbol{W}^* - \hat{\boldsymbol{W}})
\end{aligned}$$

And therefore we could easily conclude that:

$$\mathbb{E}\|\hat{\boldsymbol{W}}^\top\psi(X_1) - f^*(X_1)\|^2 \lesssim \sigma^2\frac{k(1+\log(k/\delta))}{n_2}.$$

□

4.11.2 Omitted proof of linear model with approximation error

Proof of Theorem 4.3.10. First we note that $Y = f^*(X_1) + N$, where $\mathbb{E}[N|X_1] = 0$ but $Y - (\boldsymbol{A}^*)^\top X_1$ is not necessarily mean zero, and this is where additional difficulty lies. Write approximation error term $a(X_1) := f^*(X_1) - (\boldsymbol{A}^*)^\top X_1$, namely $Y = a(X_1) + (\boldsymbol{A}^*)^\top X_1 + N$. Also, $(\boldsymbol{A}^*)^\top X_1 \equiv (\boldsymbol{W}^*)^\top\psi(X_1)$ with conditional independence.

Second, with KKT condition on the training data, we know that $\mathbb{E}[a(X_1)X_1^\top] = 0$.

Recall $\hat{\boldsymbol{W}} = \arg\min_{\boldsymbol{W}} \|\boldsymbol{Y} - \psi(\boldsymbol{X}_1)\boldsymbol{W}\|_F^2$. We have the basic inequality,

$$\begin{aligned}\frac{1}{2n_2}\|\boldsymbol{Y} - \psi(\boldsymbol{X}_1)\hat{\boldsymbol{W}}\|_F^2 \leq & \frac{1}{2n_2}\|\boldsymbol{Y} - \boldsymbol{X}_1\boldsymbol{A}^*\|_F^2 \\ = & \frac{1}{2n_2}\|\boldsymbol{Y} - \psi(\boldsymbol{X}_1)\boldsymbol{W}^*\|_F^2. \\ \text{i.e., } \frac{1}{2n_2}\|\psi(\boldsymbol{X}_1)\boldsymbol{W}^* + a(\boldsymbol{X}_1) + \boldsymbol{N} - \psi(\boldsymbol{X}_1)\hat{\boldsymbol{W}}\|_F^2 \leq & \frac{1}{2n_2}\|a(\boldsymbol{X}_1) + \boldsymbol{N}\|_F^2.\end{aligned}$$

Therefore

$$\begin{aligned}&\frac{1}{2n_2}\|\psi(\boldsymbol{X}_1)\boldsymbol{W}^* - \psi(\boldsymbol{X}_1)\hat{\boldsymbol{W}}\|^2 \\ \leq & -\frac{1}{n_2}\langle a(\boldsymbol{X}_1) + \boldsymbol{N}, \psi(\boldsymbol{X}_1)\boldsymbol{W}^* - \psi(\boldsymbol{X}_1)\hat{\boldsymbol{W}}\rangle \\ = & -\frac{1}{n_2}\langle a(\boldsymbol{X}_1), \psi(\boldsymbol{X}_1)\boldsymbol{W}^* - \psi(\boldsymbol{X}_1)\hat{\boldsymbol{W}}\rangle - \langle \boldsymbol{N}, \psi(\boldsymbol{X}_1)\boldsymbol{W}^* - \psi(\boldsymbol{X}_1)\hat{\boldsymbol{W}}\rangle \end{aligned} \tag{4.11}$$

With Assumption 4.3.9 and by concentration $0.9\frac{1}{n_2}\boldsymbol{X}_1\boldsymbol{X}_1^\top \preceq \boldsymbol{\Sigma}_{X_1} \preceq 1.1\frac{1}{n_2}\boldsymbol{X}_1\boldsymbol{X}_1^\top$, we have

$$\frac{1}{\sqrt{n_2}}\|a(\boldsymbol{X}_1)\boldsymbol{X}_1^\top\boldsymbol{\Sigma}_{X_1}^{-1/2}\|_F \leq 1.1b_0\sqrt{k} \tag{4.12}$$

Denote $\psi(\boldsymbol{X}_1) = \boldsymbol{X}_1\boldsymbol{B}$, where $\boldsymbol{B} = \boldsymbol{\Sigma}_{X_1}^{-1}\boldsymbol{\Sigma}_{X_1X_2}$ is rank k under exact CI since $\boldsymbol{\Sigma}_{X_1X_2} = \boldsymbol{\Sigma}_{X_1Y}\boldsymbol{\Sigma}_Y^{-1}\boldsymbol{\Sigma}_{YX_2}$. We have

$$\begin{aligned}&\frac{1}{n_2}\langle a(\boldsymbol{X}_1), \psi(\boldsymbol{X}_1)\boldsymbol{W}^* - \psi(\boldsymbol{X}_1)\hat{\boldsymbol{W}}\rangle \\ = & \frac{1}{n_2}\langle a(\boldsymbol{X}_1), \boldsymbol{X}_1\boldsymbol{B}\boldsymbol{W}^* - \boldsymbol{X}_1\boldsymbol{B}\hat{\boldsymbol{W}}\rangle \\ = & \frac{1}{n_2}\langle \boldsymbol{\Sigma}_{X_1}^{-1/2}\boldsymbol{X}_1^\top a(\boldsymbol{X}_1), \boldsymbol{\Sigma}_{X_1}^{1/2}(\boldsymbol{B}\boldsymbol{W}^* - \boldsymbol{B}\hat{\boldsymbol{W}})\rangle \\ \leq & 1.1b_0\sqrt{\frac{k}{n_2}}\|\boldsymbol{\Sigma}_{X_1}^{1/2}(\boldsymbol{B}\boldsymbol{W}^* - \boldsymbol{B}\hat{\boldsymbol{W}})\|_F \qquad \text{(from Ineq. Equation (4.12))}\end{aligned}$$

Back to Eqn. Equation (4.11), we get

$$\begin{aligned}&\frac{1}{2n_2}\|\psi(\boldsymbol{X}_1)\boldsymbol{W}^* - \psi(\boldsymbol{X}_1)\hat{\boldsymbol{W}}\|_F^2 \\ \lesssim & \sqrt{\frac{k}{n_2}}\|\boldsymbol{\Sigma}_{X_1}^{1/2}(\boldsymbol{B}\boldsymbol{W}^* - \boldsymbol{B}\hat{\boldsymbol{W}})\|_F + \frac{1}{n_2}\|P_{\boldsymbol{X}_1}\boldsymbol{N}\|_F\|\boldsymbol{X}_1(\boldsymbol{B}\boldsymbol{W}^* - \boldsymbol{B}\hat{\boldsymbol{W}})\|_F\end{aligned}$$

$$\lesssim \left(\frac{\sqrt{k}}{n_2} + \frac{1}{n_2}\|P_{\boldsymbol{X}_1}\boldsymbol{N}\|_F\right)\|\boldsymbol{X}_1(\boldsymbol{B}\boldsymbol{W}^* - \boldsymbol{B}\hat{\boldsymbol{W}})\|_F$$

$$\Longrightarrow \frac{1}{\sqrt{n_2}}\|\psi(\boldsymbol{X}_1)\boldsymbol{W}^* - \psi(\boldsymbol{X}_1)\hat{\boldsymbol{W}}\|_F \lesssim \sqrt{\frac{k(1+\log k/\delta)}{n_2}}. \qquad \text{(from Lemma 4.9.7)}$$

Finally, by concentration we transfer the result from empirical loss to excess risk and get:

$$\mathbb{E}[\|\psi(X_1)\boldsymbol{W}^* - \psi(X_1)\hat{\boldsymbol{W}}\|^2] \lesssim \frac{k(1+\log(k/\delta))}{n_2}.$$

□

4.11.3 Argument on denoising auto-encoder or context encoder

Remark 4.11.1. *We note that since $X_1 \perp X_2|Y$ ensures $X_1 \perp h(X_2)|Y$ for any deterministic function h, we could replace X_2 by $h(X_2)$ and all results hold. Therefore in practice, we could use $h(\psi(X_1))$ instead of $\psi(X_1)$ for downstream task. Specifically with denoising auto-encoder or context encoder, one could think about h as the inverse of decoder D ($h = D^{-1}$) and use $D^{-1}\psi \equiv E$ the encoder function as the representation for downstream tasks, which is more commonly used in practice.*

This section explains what we claim in Remark Remark 4.11.1. For context encoder, the reconstruction loss targets to find the encoder E^* and decoder D^* that achieve

$$\min_E \min_D \mathbb{E}\|X_2 - D(E(X_1))\|_F^2, \tag{4.13}$$

where X_2 is the masked part we want to recover and X_1 is the remainder.

If we naively apply our theorem we should use $D^*(E^*(\cdot))$ as the representation, while in practice we instead use only the encoder part $E^*(\cdot)$ as the learned representation. We argue that our theory also support this practical usage if we view the problem differently. Consider the pretext task to predict $(D^*)^{-1}(X_2)$ instead of X_2 directly, namely,

$$\bar{E} \leftarrow \arg\min_E \mathbb{E}\|(D^*)^{-1}(X_2) - E(X_1)\|^2, \tag{4.14}$$

and then we should indeed use $E(X_1)$ as the representation. On one hand, when $X_1 \perp X_2|Y$, it also satisfies $X_1 \perp (D^*)^{-1}(X_2)|Y$ since $(D^*)^{-1}$ is a deterministic function of X_2 and all our theory applies. On the other

hand, the optimization on Equation (4.13) or Equation (4.14) give us similar result. Let

$$E^* = \arg\min_E \mathbb{E}[\|X_2 - D^*(E(X_1))\|^2],$$

and $\mathbb{E}\|X_2 - D^*(E^*(X_1))\|^2 \leq \epsilon$, then with pretext task as in Equation (4.14) we have that:

$$\begin{aligned}\mathbb{E}\|(D^*)^{-1}(X_2) - E^*(X_1)\|^2 =& \mathbb{E}\|(D^*)^{-1}(X_2) - (D^*)^{-1} \circ D^*(E^*(X_1))\|^2 \\ \leq& \|(D^*)^{-1}\|_{\text{Lip}}^2 \mathbb{E}\|X_2 - D^*(E^*(X_1))\|^2 \\ \leq& L^2\epsilon,\end{aligned}$$

where $L := \|(D^*)^{-1}\|_{\text{Lip}}$ is the Lipschitz constant for function $(D^*)^{-1}$. This is to say, in practice, we optimize over Equation (4.13), and achieves a good representation $E^*(X_1)$ such that $\epsilon_{\text{pre}} \leq L\sqrt{\epsilon}$ and thus performs well for downstream tasks. (Recall ϵ_{pre} is defined in Theorem 4.4.4 that measures how well we have learned the pretext task.)

4.12 Omitted Proofs Beyond Conditional Independence

4.12.1 Warm-up: Jointly Gaussian Variables

As before, for simplicity we assume all data is centered in this case.

Assumption 4.12.1 (Approximate Conditional Independent Given Latent Variables)**.** *Assume there exists some latent variable $Z \in \mathbb{R}^m$ such that*

$$\|\boldsymbol{\Sigma}_{X_1}^{-1/2}\boldsymbol{\Sigma}_{X_1,X_2|\bar{Y}}\|_F \leq \epsilon_{CI},$$

$\sigma_{k+m}(\boldsymbol{\Sigma}_{Y\bar{Y}}^{\dagger}\boldsymbol{\Sigma}_{\bar{Y}X_2}) = \beta > 0$ [8] *and $\boldsymbol{\Sigma}_{X_2,\bar{Y}}$ is of rank $k+m$, where $\bar{Y} = [Y, Z]$.*

When X_1 is not exactly CI of X_2 given Y and Z, the approximation error depends on the norm of $\|\boldsymbol{\Sigma}_{X_1}^{-1/2}\boldsymbol{\Sigma}_{X_1,X_2|\bar{Y}}\|_2$. Let $\hat{\boldsymbol{W}}$ be the solution from Equation (uses CI).

[8] $\sigma_k(\boldsymbol{A})$ denotes k-th singular value of $\boldsymbol{A}$, and $\boldsymbol{A}^{\dagger}$ is the pseudo-inverse of $\boldsymbol{A}$.

Theorem 4.12.2. *Under Assumption 4.12.1 with constant ϵ_{CI} and β, then the excess risk satisfies*

$$\mathrm{ER}_{\psi^*}[\hat{\boldsymbol{W}}] := \mathbb{E}[\|\hat{\boldsymbol{W}}^\top \psi^*(X_1) - f^*(X_1)\|_F^2] \lesssim \frac{\epsilon_{CI}^2}{\beta^2} + \mathrm{Tr}(\boldsymbol{\Sigma}_{YY|X_1}) \frac{d_2 + \log(d_2/\delta)}{n_2}.$$

Proof of Theorem 4.12.2. Let $\boldsymbol{V} := f^*(\boldsymbol{X}_1) \equiv \boldsymbol{X}_1 \boldsymbol{\Sigma}_{X_1X_1}^{-1} \boldsymbol{\Sigma}_{1Y}$ be our target direction. Denote the optimal representation matrix by $\Psi := \psi(\boldsymbol{X}_1) \equiv \boldsymbol{X}_1 \boldsymbol{A}$ (where $\boldsymbol{A} := \boldsymbol{\Sigma}_{X_1X_1}^{-1} \boldsymbol{\Sigma}_{X_1X_2}$).

Next we will make use of the conditional covariance matrix:

$$\boldsymbol{\Sigma}_{X_1X_2|\bar{Y}} := \boldsymbol{\Sigma}_{X_1X_2} - \boldsymbol{\Sigma}_{X_1\bar{Y}} \boldsymbol{\Sigma}_{\bar{Y}}^{-1} \boldsymbol{\Sigma}_{\bar{Y}X_2},$$

and plug it in into the definition of Ψ:

$$\begin{aligned}\Psi =& \boldsymbol{X}_1 \boldsymbol{\Sigma}_{X_1X_1}^{-1} \boldsymbol{\Sigma}_{X_1\bar{Y}} \boldsymbol{\Sigma}_{\bar{Y}}^{-1} \boldsymbol{\Sigma}_{\bar{Y}X_2} + \boldsymbol{X}_1 \boldsymbol{\Sigma}_{X_1X_1}^{-1} \boldsymbol{\Sigma}_{X_1X_2|\bar{Y}} \\ =:& \boldsymbol{L} + \boldsymbol{E},\end{aligned}$$

where $\boldsymbol{L} := \boldsymbol{X}_1 \boldsymbol{\Sigma}_{X_1X_1}^{-1} \boldsymbol{\Sigma}_{X_1\bar{Y}} \boldsymbol{\Sigma}_{\bar{Y}}^{-1} \boldsymbol{\Sigma}_{\bar{Y}X_2}$ and $\boldsymbol{E} := \boldsymbol{X}_1 \boldsymbol{\Sigma}_{X_1X_1}^{-1} \boldsymbol{\Sigma}_{X_1X_2|\bar{Y}}$. We analyze these two terms respectively.

For $\boldsymbol{L}$, we note that $\mathrm{span}(\boldsymbol{V}) \subseteq \mathrm{span}(\boldsymbol{L})$: $\boldsymbol{L}\boldsymbol{\Sigma}_{X_2\bar{Y}}^\dagger \boldsymbol{\Sigma}_{\bar{Y}} = \boldsymbol{X}_1 \boldsymbol{\Sigma}_{X_1X_1}^{-1} \boldsymbol{\Sigma}_{X_1\bar{Y}}$. By right multiplying the selector matrix S_Y we have: $\boldsymbol{L}\boldsymbol{\Sigma}_{X_2\bar{Y}}^\dagger \boldsymbol{\Sigma}_{\bar{Y}Y} = \boldsymbol{X}_1 \boldsymbol{\Sigma}_{X_1X_1}^{-1} \boldsymbol{\Sigma}_{X_1Y}$, i.e., $\boldsymbol{L}\bar{\boldsymbol{W}} = \boldsymbol{V}$, where $\bar{\boldsymbol{W}} := \boldsymbol{\Sigma}_{X_2\bar{Y}}^\dagger \boldsymbol{\Sigma}_{\bar{Y}Y}$. From our assumption that $\sigma_r(\boldsymbol{\Sigma}_{\bar{Y}Y}^\dagger \boldsymbol{\Sigma}_{\bar{Y}X_2}) = \beta$, we have $\|\bar{\boldsymbol{W}}\|_2 \le \|\boldsymbol{\Sigma}_{X_2\bar{Y}}^\dagger \boldsymbol{\Sigma}_{\bar{Y}}\|_2 \le 1/\beta$. (Or we could directly define β as $\sigma_k(\boldsymbol{\Sigma}_{Y\bar{Y}}^\dagger \boldsymbol{\Sigma}_{\bar{Y}X_2}) \equiv \|\bar{\boldsymbol{W}}\|_2$.)

By concentration, we have $\boldsymbol{E} = \boldsymbol{X}_1 \boldsymbol{\Sigma}_{X_1X_1}^{-1} \boldsymbol{\Sigma}_{X_1X_2|\bar{Y}}$ converges to $\boldsymbol{\Sigma}_{X_1X_1}^{-1/2} \boldsymbol{\Sigma}_{X_1X_2|\bar{Y}}$. Specifically, when $n \gg k + \log 1/\delta$, $\|\boldsymbol{E}\|_F \le 1.1 \|\boldsymbol{\Sigma}_{X_1X_1}^{-1/2} \boldsymbol{\Sigma}_{X_1X_2|\bar{Y}}\|_F \le 1.1\epsilon_{\mathrm{CI}}$ (by using Claim 4.9.3). Together we have $\|\boldsymbol{E}\bar{\boldsymbol{W}}\|_F \lesssim \epsilon_{\mathrm{CI}}/\beta$.

Let $\hat{\boldsymbol{W}} = \arg\min_{\boldsymbol{W}} \|\boldsymbol{Y} - \Psi\boldsymbol{W}\|^2$. We note that $\boldsymbol{Y} = \boldsymbol{N} + \boldsymbol{V} = \boldsymbol{N} + \Psi\bar{\boldsymbol{W}} - \boldsymbol{E}\bar{\boldsymbol{W}}$ where $\boldsymbol{V}$ is our target direction and $\boldsymbol{N}$ is random noise (each row of $\boldsymbol{N}$ has covariance matrix $\boldsymbol{\Sigma}_{YY|X_1}$).

From basic inequality, we have:

$$\begin{aligned}\|\Psi\hat{\boldsymbol{W}} - \boldsymbol{Y}\|_F^2 \le& \|\Psi\bar{\boldsymbol{W}} - \boldsymbol{Y}\|_F^2 = \|\boldsymbol{N} - \boldsymbol{E}\bar{\boldsymbol{W}}\|_F^2. \\ \implies \|\Psi\hat{\boldsymbol{W}} - \boldsymbol{V} - \boldsymbol{E}\bar{\boldsymbol{W}}\|^2 \le& 2\langle \Psi\hat{\boldsymbol{W}} - \boldsymbol{V} - \boldsymbol{E}\bar{\boldsymbol{W}}, \boldsymbol{N} - \boldsymbol{E}\bar{\boldsymbol{W}} \rangle \\ \implies \|\Psi\hat{\boldsymbol{W}} - \boldsymbol{V} - \boldsymbol{E}\bar{\boldsymbol{W}}\| \le& \|P_{[\Psi,E,V]}\boldsymbol{N}\| + \|\boldsymbol{E}\bar{\boldsymbol{W}}\|\end{aligned}$$

$$\implies \|\Psi\hat{\boldsymbol{W}} - \boldsymbol{V}\| \lesssim \|\boldsymbol{E}\|_F \|\bar{\boldsymbol{W}}\| + (\sqrt{d_2} + \sqrt{\log 1/\delta})\sqrt{\mathrm{Tr}(\boldsymbol{\Sigma}_{YY|X_1})}. \qquad \text{(from Claim 4.9.5)}$$

$$\leq \sqrt{n_2}\frac{\epsilon_{\mathrm{CI}}}{\beta} + (\sqrt{d_2} + \sqrt{\log 1/\delta})\sqrt{\mathrm{Tr}(\boldsymbol{\Sigma}_{YY|X_1})}. \qquad \text{(from Assumption 4.12.1)}$$

Next, by the same procedure that concentrates $\frac{1}{n_2}\boldsymbol{X}_1^\top \boldsymbol{X}_1$ to $\boldsymbol{\Sigma}_{X_1X_1}$ with Claim Claim 4.9.3, we could easily get

$$\mathrm{ER}[\hat{\boldsymbol{W}}] := \mathbb{E}[\|\hat{\boldsymbol{W}}^\top \psi(X_1) - f^*(X_1)\|^2] \lesssim \frac{\epsilon_{\mathrm{CI}}^2}{\beta^2} + \mathrm{Tr}(\boldsymbol{\Sigma}_{YY|X_1})\frac{d_2 + \log 1/\delta}{n_2}.$$

□

4.12.2 Measuring conditional dependence with cross-covariance operator

$L^2(P_X)$ denotes the Hilbert space of square integrable function with respect to the measure P_X, the marginal distribution of X. We are interested in some function class $\mathcal{H}_x \subset L^2(P_X)$ that is induced from some feature maps:

Definition 4.12.3 (General and Universal feature Map)**.** *We denote feature map $\phi : \mathcal{X} \to \mathcal{F}$ that maps from a compact input space $\mathcal{X}$ to the feature space $\mathcal{F}$. $\mathcal{F}$ is a Hilbert space associated with inner product: $\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle_{\mathcal{F}}$. The associated function class is: $\mathcal{H}_x = \{h : \mathcal{X} \to \mathbb{R} | \exists w \in \mathcal{F}, h(\boldsymbol{x}) = \langle w, \phi(\boldsymbol{x}) \rangle_{\mathcal{F}}, \forall \boldsymbol{x} \in \mathcal{X}\}$. We call ϕ universal if the induced $\mathcal{H}_x$ is dense in $L^2(P_X)$.*

Linear model is a special case when feature map $\phi = Id$ is identity mapping and the inner product is over Euclidean space. A feature map with higher order polynomials correspondingly incorporate high order moments [Fukumizu et al., 2004, Gretton et al., 2005]. For discrete variable Y we overload ϕ as the one-hot embedding.

Remark 4.12.4. *For continuous data, any universal kernel like Gaussian kernel or RBF kernel induce the universal feature map that we require [Micchelli et al., 2006]. Two-layer neural network with infinite width also satisfy it, i.e., $\forall \boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d, \phi_{NN}(\boldsymbol{x}) : \mathcal{S}^{d-1} \times \mathbb{R} \to \mathbb{R}, \phi_{NN}(\boldsymbol{x})[\boldsymbol{w}, b] = \sigma(\boldsymbol{w}^\top \boldsymbol{x} + b)$ [Barron, 1993].*

When there's no ambiguity, we overload ϕ_1 as the random variable $\phi_1(X_1)$ over domain $\mathcal{F}_1$, and $\mathcal{H}_1$ as the function class over X_1. Next we characterize CI using the cross-covariance operator.

Definition 4.12.5 (Cross-covariance operator)**.** *For random variables $X \in \mathcal{X}, Y \in \mathcal{Y}$ with joint distribution*

$P : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, *and associated feature maps* ϕ_x *and* ϕ_y, *we denote by* $\mathcal{C}_{\phi_x\phi_y} = \mathbb{E}[\phi_x(X) \otimes \phi_y(Y)] = \int_{\mathcal{X}\times\mathcal{Y}} \phi_x(x) \otimes \phi_y(y) dP(x,y)$, *the (un-centered) cross-covariance operator. Similarly we denote by* $\mathcal{C}_{X\phi_y} = \mathbb{E}[X \otimes \phi_y(Y)] : \mathcal{F}_y \to \mathcal{X}$.

To understand what $\mathcal{C}_{\phi_x\phi_y}$ is, we note it is of the same shape as $\phi_x(x) \otimes \phi_y(y)$ for each individual $x \in \mathcal{X}, y \in \mathcal{Y}$. It can be viewed as an operator: $\mathcal{C}_{\phi_x\phi_y} : \mathcal{F}_y \to \mathcal{F}_x$, $\mathcal{C}_{\phi_x\phi_y} f = \int_{\mathcal{X}\times\mathcal{Y}} \langle \phi_y(y), f\rangle \phi_x(x) dP(x,y), \forall f \in \mathcal{F}_y$. For any $f \in \mathcal{H}_x$ and $g \in \mathcal{H}_y$, it satisfies: $\langle f, \mathcal{C}_{\phi_x\phi_y} g\rangle_{\mathcal{H}_x} = \mathbb{E}_{XY}[f(X)g(Y)]$[Baker, 1973, Fukumizu et al., 2004]. CI ensures $\mathcal{C}_{\phi_1 X_2|\phi_y} = 0$ for arbitrary ϕ_1, ϕ_2:

Lemma 4.12.6. *With one-hot encoding map* ϕ_y *and arbitrary* ϕ_1, $X_1 \perp X_2 | Y$ *ensures:*

$$\mathcal{C}_{\phi_1 X_2|\phi_y} := \mathcal{C}_{\phi_1 X_2} - \mathcal{C}_{\phi_1\phi_y}\mathcal{C}^{-1}_{\phi_y\phi_y}\mathcal{C}_{\phi_y X_2} = 0. \tag{4.15}$$

A more complete discussion of cross-covariance operator and CI can be found in [Fukumizu et al., 2004]. Also, recall that an operator $\mathcal{C} : \mathcal{F}_y \to \mathcal{F}_x$ is Hilbert-Schmidt (HS) [Reed, 2012] if for complete orthonormal systems (CONSs) $\{\zeta_i\}$ of $\mathcal{F}_x$ and $\{\eta_i\}$ of $\mathcal{F}_y$, $\|\mathcal{C}\|^2_{\text{HS}} := \sum_{i,j} \langle \zeta_j, \mathcal{C}\eta_i\rangle^2_{\mathcal{F}_x} < \infty$. The Hilbert-Schmidt norm generalizes the Frobenius norm from matrices to operators, and we will later use $\|\mathcal{C}_{\phi_1 X_2|\phi_y}\|$ to quantify approximate CI.

We note that covariance operators [Fukumizu et al., 2009, 2004, Baker, 1973] are commonly used to capture conditional dependence of random variables. In this work, we utilize the covariance operator to quantify the performance of the algorithm even when the algorithm is *not a kernel method*.

4.12.3 Omitted Proof in General Setting

Claim 4.12.7. *For feature maps* ϕ_1 *with universal property, we have:*

$$\begin{aligned}
\psi^*(X_1) :=& \mathbb{E}[X_2|X_1] = \mathbb{E}^L[X_2|\phi_1] \\
=& \mathcal{C}_{X_2\phi_1}\mathcal{C}^{-1}_{\phi_1\phi_1}\phi_1(X_1). \\
\textit{Our target } f^*(X_1) :=& \mathbb{E}[Y|X_1] = \mathbb{E}^L[Y|\phi_1] \\
=& \mathcal{C}_{Y\phi_1}\mathcal{C}^{-1}_{\phi_1\phi_1}\phi_1(X_1).
\end{aligned}$$

For general feature maps, we instead have:

$$\begin{aligned}\psi^*(X_1) :=& \arg\min_{f\in\mathcal{H}_1^{d_2}} \mathbb{E}_{X_1X_2}\|X_2 - f(X_1)\|_2^2\\ =& \mathcal{C}_{X_2\phi_1}\mathcal{C}_{\phi_1\phi_1}^{-1}\phi_1(X_1).\\ \textit{Our target } f^*(X_1) :=& \arg\min_{f\in\mathcal{H}_1^{k}} \mathbb{E}_{X_1Y}\|Y - f(X_1)\|_2^2\\ =& \mathcal{C}_{Y\phi_1}\mathcal{C}_{\phi_1\phi_1}^{-1}\phi_1(X_1).\end{aligned}$$

To prove Claim Claim 4.12.7, we show the following lemma:

Lemma 4.12.8. *Let $\phi : \mathcal{X} \to \mathcal{F}_x$ be a universal feature map, then for random variable $Y \in \mathcal{Y}$ we have:*

$$\mathbb{E}[Y|X] = \mathbb{E}^L[Y|\phi(X)].$$

Proof of Lemma 4.12.8. Denote by $\mathbb{E}[Y|X = x] =: f(x)$. Since ϕ is dense in $\mathcal{X}$, there exists a linear operator $a : \mathcal{X} \to \mathbb{R}$ such that $\int_{x\in\mathcal{X}} a(x)\phi(x)[\cdot]dx = f(\cdot)$ a.e. Therefore the result comes directly from the universal property of ϕ. □

Proof of Claim Claim 4.12.7. We want to show that for random variables Y, X, where X is associated with a universal feature map ϕ_x, we have $\mathbb{E}[Y|X] = \mathcal{C}_{Y\phi_x(X)}\mathcal{C}^{-1}_{\phi_x(X)\phi_x(X)}\phi_x(X)$.

First, from Lemma 4.12.8, we have that $\mathbb{E}[Y|X] = \mathbb{E}^L[Y|\phi_x(X)]$. Next, write $A^* : \mathcal{F}_x \to \mathcal{Y}$ as the linear operator that satisfies

$$\begin{aligned}\mathbb{E}[Y|X] &= A^*\phi_x(X)\\ \text{s.t. } A^* &= \arg\min_A \mathbb{E}[\|Y - A\phi_x(X)\|^2].\end{aligned}$$

Therefore from the stationary condition we have $A^*\mathbb{E}_X[\phi_x(X)\otimes\phi_x(X)] = \mathbb{E}_{XY}[Y\otimes\phi_x(X)]$. Or namely we get $A^* = \mathcal{C}_{Y\phi_x}\mathcal{C}^{-1}_{\phi_x\phi_x}$ simply from the definition of the cross-covariance operator $\mathcal{C}$. □

Claim 4.12.9. $\|\mathcal{C}^{-1/2}_{\phi_1\phi_1}\mathcal{C}_{\phi_1X_2|\phi_{\bar{y}}}\|^2_{HS} = \mathbb{E}_{X_1}[\|\mathbb{E}[X_2|X_1] - \mathbb{E}_{\bar{Y}}[\mathbb{E}[X_2|\bar{Y}]|X_1]\|^2] = \epsilon^2_{CI}$.

Proof.

$$
\begin{aligned}
&\|\mathcal{C}_{\phi_1\phi_1}^{-1/2}\mathcal{C}_{\phi_1 X_2|\phi_{\bar{Y}}}\|_{\mathrm{HS}}^2 \\
=&\int_{X_1}\left\|\int_{X_2}\left(\frac{p_{X_1X_2}(\boldsymbol{x}_1,\boldsymbol{x}_2)}{p_{X_1}(\boldsymbol{x}_1)}-\frac{p_{X_1\perp X_2|Y}(\boldsymbol{x}_1,\boldsymbol{x}_2)}{p_{X_1}(\boldsymbol{x}_1)}\right)X_2 dp_{\boldsymbol{x}_2}\right\|^2 dp_{\boldsymbol{x}_1} \\
=&\mathbb{E}_{X_1}[\|\mathbb{E}[X_2|X_1]-\mathbb{E}_{\bar{Y}}[\mathbb{E}[X_2|\bar{Y}]|X_1]\|^2].
\end{aligned}
$$

□

4.12.4 Omitted Proof for Main Results

We first prove a simpler version without approximation error.

Theorem 4.12.10. *For a fixed $\delta \in (0,1)$, under Assumption 4.4.1, Assumption 4.3.4, if there is no approximation error, i.e., there exists a linear operator A such that $f^*(X_1) \equiv A\phi_1(X_1)$, if $n_1, n_2 \gg \rho^4(d_2 + \log 1/\delta)$, and we learn the pretext tasks such that:*

$$\mathbb{E}\|\tilde{\psi}(X_1) - \psi^*(X_1)\|_F^2 \le \epsilon_{pre}^2.$$

Then we are able to achieve generalization for downstream task with probability $1-\delta$:

$$\mathbb{E}[\|f_{\mathcal{H}_1}^*(X_1) - \hat{\boldsymbol{W}}^\top\tilde{\psi}(X_1)\|^2] \le \tilde{\mathcal{O}}\{\sigma^2\frac{d_2}{n_2} + \frac{\epsilon_{CI}^2}{\beta^2} + \frac{\epsilon_{pre}^2}{\beta^2}\}. \tag{4.16}$$

Proof of Theorem 4.12.10. We follow the similar procedure as Theorem 4.12.2. For the setting of no approximation error, we have $f^* = f_{\mathcal{H}_1}^*$, and the residual term $N := Y - f^*(X_1)$ is a mean-zero random variable with $\mathbb{E}[\|N\|^2|X_1] \lesssim \sigma^2$ according to our data assumption in Section 4.3. $\boldsymbol{N} = \boldsymbol{Y} - f^*(\boldsymbol{X}_1^{\mathrm{down}})$ is the collected n_2 samples of noise terms. We write $Y \in \mathbb{R}^{d_3}$. For classification task, we have $Y \in \{\boldsymbol{e}_i, i \in [k]\} \subset \mathbb{R}^k$ (i.e, $d_3 = k$) is one-hot encoded random variable. For regression problem, Y might be otherwise encoded. For instance, in the yearbook dataset, Y ranges from 1905 to 2013 and represents the years that the photos are taken. We want to note that our result is general for both cases: the bound doesn't depend on d_3, but only depends on the variance of N.

Let $\Psi^*, \boldsymbol{L}, \boldsymbol{E}, \boldsymbol{V}$ be defined as follows:

Let $\boldsymbol{V} = f^*(\boldsymbol{X}_1^{\text{down}}) \equiv f^*_{\mathcal{H}_1}(\boldsymbol{X}_1^{\text{down}}) \equiv \phi(\boldsymbol{X}_1^{\text{down}})\mathcal{C}_{\phi_1}^{-1}\mathcal{C}_{\phi_1 Y}$ be our target direction. Denote the optimal representation matrix by

$$\begin{aligned}\Psi^* :=& \psi^*(\boldsymbol{X}_1^{\text{down}}) \\ =& \phi(\boldsymbol{X}_1^{\text{down}})\mathcal{C}_{\phi_1\phi_1}^{-1}\mathcal{C}_{\phi_1 X_2} \\ =& \phi(\boldsymbol{X}_1^{\text{down}})\mathcal{C}_{\phi_1\phi_1}^{-1}\mathcal{C}_{\phi_1\phi_{\bar{y}}}\mathcal{C}_{\phi_{\bar{y}}}^{-1}\boldsymbol{\Sigma}_{\phi_{\bar{y}}X_2} + \phi(\boldsymbol{X}_1^{\text{down}})\mathcal{C}_{\phi_1\phi_1}^{-1}\mathcal{C}_{\phi_1 X_2|\phi_{\bar{y}}} \\ =:& \boldsymbol{L} + \boldsymbol{E},\end{aligned}$$

where $\boldsymbol{L} = \phi(\boldsymbol{X}_1^{\text{down}})\mathcal{C}_{\phi_1\phi_1}^{-1}\mathcal{C}_{\phi_1\phi_{\bar{y}}}\mathcal{C}_{\phi_{\bar{y}}}^{-1}\mathcal{C}_{\phi_{\bar{y}}X_2}$ and $\boldsymbol{E} = \phi(\boldsymbol{X}_1^{\text{down}})\mathcal{C}_{\phi_1\phi_1}^{-1}\mathcal{C}_{\phi_1 X_2|\bar{Y}}$.

In this proof, we denote S_Y as the matrix such that $S_Y\phi_{\bar{y}} = Y$. Specifically, if Y is of dimension d_3, S_Y is of size $d_3 \times |\mathcal{Y}||\mathcal{Z}|$. Therefore $S_Y\boldsymbol{\Sigma}_{\phi_y A} = \boldsymbol{\Sigma}_{YA}$ for any random variable A.

Therefore, similarly we have:

$$\boldsymbol{L}\boldsymbol{\Sigma}^{\dagger}_{X_2\phi_{\bar{y}}}\boldsymbol{\Sigma}_{\phi_{\bar{y}}\phi_{\bar{y}}}S_Y^{\top} = \boldsymbol{L}\boldsymbol{\Sigma}^{\dagger}_{X_2\phi_{\bar{y}}}\boldsymbol{\Sigma}_{\phi_{\bar{y}}Y} = \boldsymbol{L}\bar{\boldsymbol{W}} = \boldsymbol{V}$$

where $\bar{\boldsymbol{W}} := \boldsymbol{\Sigma}^{\dagger}_{X_2\phi_{\bar{y}}}\boldsymbol{\Sigma}_{\phi_{\bar{y}}Y}$ satisfies $\|\bar{\boldsymbol{W}}\|_2 = 1/\beta$. Therefore $\text{span}(\boldsymbol{V}) \subseteq \text{span}(\boldsymbol{L})$ since we have assumed that $\boldsymbol{\Sigma}^{\dagger}_{X_2\phi_{\bar{y}}}\boldsymbol{\Sigma}_{\phi_{\bar{y}}Y}$ to be full rank.

On the other hand, $\boldsymbol{E} = \phi_1(\boldsymbol{X}_1^{\text{down}})\mathcal{C}_{\phi_1\phi_1}^{-1}\mathcal{C}_{\phi_1 X_2|\bar{Y}}$ concentrates to $\mathcal{C}_{\phi_1\phi_1}^{-1/2}\mathcal{C}_{\phi_1 X_2|\phi_{\bar{y}}}$. Specifically, when $n \gg k + \log 1/\delta$, $\frac{1}{n_2}\|\boldsymbol{E}\|_F^2 \leq 1.1\|\mathcal{C}_{\phi_1\phi_1}^{-1/2}\mathcal{C}_{\phi_1 X_2|\phi_{\bar{y}}}\|_F^2 \leq 1.1\epsilon_{\text{CI}}^2$ (by using Claim 4.9.4). Together we have $\|\boldsymbol{E}\bar{\boldsymbol{W}}\|_F \lesssim \epsilon_{\text{CI}}/\beta$.

We also introduce the error from not learning ψ^* exactly: $\boldsymbol{E}^{\text{pre}} = \Psi - \Psi^* := \tilde{\psi}(\boldsymbol{X}_1^{\text{down}}) - \psi^*(\boldsymbol{X}_1^{\text{down}})$. With proper concentration and our assumption, we have that $\mathbb{E}\|\psi(X_1) - \psi^*(X_1)\|^2 \leq \epsilon_{\text{pre}}$ and $\frac{1}{\sqrt{n_2}}\|\psi(\boldsymbol{X}_1^{\text{down}}) - \psi^*(\boldsymbol{X}_1^{\text{down}})\|^2 \leq 1.1\epsilon_{\text{pre}}$.

Also, the noise term after projection satisfies $\|P_{[\Psi,\boldsymbol{E},\boldsymbol{V}]}\boldsymbol{N}\| \lesssim \sqrt{d_2(1+\log d_2/\delta)}\sigma$ as using Lemma 4.9.7. Therefore $\Psi = \Psi^* - \boldsymbol{E}^{\text{pre}} = \boldsymbol{L} + \boldsymbol{E} - \boldsymbol{E}^{\text{pre}}$.

Recall that $\hat{\boldsymbol{W}} = \arg\min_{\boldsymbol{W}} \|\psi(\boldsymbol{X}_1^{\text{down}})\boldsymbol{W} - \boldsymbol{Y}\|_F^2$. And with exactly the same procedure as Theorem 4.12.2 we also get that:

$$\begin{aligned}\|\Psi\hat{\boldsymbol{W}} - \boldsymbol{V}\| \leq & 2\|\boldsymbol{E}\bar{\boldsymbol{W}}\| + 2\|\boldsymbol{E}^{\text{pre}}\bar{\boldsymbol{W}}\| + \|P_{[\Psi,\boldsymbol{E},\boldsymbol{V},\boldsymbol{E}^{\text{pre}}]}\boldsymbol{N}\| \\ \lesssim & \sqrt{n_2}\frac{\epsilon_{\text{CI}} + \epsilon_{\text{pre}}}{\beta} + \sigma\sqrt{d_2(1+\log(d_2/\delta))}.\end{aligned}$$

With the proper concentration we also get:

$$\mathbb{E}[\|\hat{\boldsymbol{W}}^\top\psi(X_1) - f^*_{\mathcal{H}_1}(X_1)\|^2] \lesssim \frac{\epsilon^2_{\text{CI}} + \epsilon^2_{\text{pre}}}{\beta^2} + \sigma^2\frac{d_2(1+\log(d_2/\delta))}{n_2}.$$

□

Next we move on to the proof of our main result Theorem 4.4.4 where approximation error occurs.

Proof of Theorem 4.4.4. The proof is a combination of Theorem 4.3.10 and Theorem 4.12.10. We follow the same notation as in Theorem 4.12.10. Now the only difference is that an additional term $a(\boldsymbol{X}_1^{\text{down}})$ is included in $\boldsymbol{Y}$:

$$\begin{aligned}\boldsymbol{Y} =& \boldsymbol{N} + f^*(\boldsymbol{X}_1^{\text{down}}) \\ =& \boldsymbol{N} + \Psi^*\bar{\boldsymbol{W}} + a(\boldsymbol{X}_1^{\text{down}}) \\ =& \boldsymbol{N} + (\Psi + \boldsymbol{E}^{\text{pre}})\bar{\boldsymbol{W}} + a(\boldsymbol{X}_1^{\text{down}}) \\ =& \Psi\bar{\boldsymbol{W}} + (\boldsymbol{N} + \boldsymbol{E}^{\text{pre}}\bar{\boldsymbol{W}} + a(\boldsymbol{X}_1^{\text{down}})).\end{aligned}$$

From re-arranging $\frac{1}{2n_2}\|\boldsymbol{Y} - \Psi\hat{\boldsymbol{W}}\|_F^2 \leq \frac{1}{2n_2}\|\boldsymbol{Y} - \Psi\bar{\boldsymbol{W}}\|_F^2$,

$$\frac{1}{2n_2}\|\Psi(\bar{\boldsymbol{W}} - \hat{\boldsymbol{W}}) + (\boldsymbol{N} + \boldsymbol{E}^{\text{pre}} + a(\boldsymbol{X}_1^{\text{down}}))\|_F^2 \leq \frac{1}{2n_2}\|\boldsymbol{N} + \boldsymbol{E}^{\text{pre}}\bar{\boldsymbol{W}} + a(\boldsymbol{X}_1^{\text{down}})\|_F^2 \tag{4.17}$$

$$\Rightarrow \frac{1}{2n_2}\|\Psi(\bar{\boldsymbol{W}} - \hat{\boldsymbol{W}})\|_F^2 \leq \frac{1}{n_2}\langle\Psi(\bar{\boldsymbol{W}} - \hat{\boldsymbol{W}}), \boldsymbol{N} + \boldsymbol{E}^{\text{pre}}\bar{\boldsymbol{W}} + a(\boldsymbol{X}_1^{\text{down}})\rangle. \tag{4.18}$$

Then with similar procedure as in the proof of Theorem 4.3.10, and write Ψ as $\phi(X_1^{\text{down}})\boldsymbol{B}$, we have:

$$\begin{aligned}&\frac{1}{n_2}\langle\Psi(\bar{\boldsymbol{W}} - \hat{\boldsymbol{W}}), a(\boldsymbol{X}_1^{\text{down}})\rangle \\ =&\frac{1}{n_2}\langle\boldsymbol{B}(\bar{\boldsymbol{W}} - \hat{\boldsymbol{W}}), \phi(\boldsymbol{X}_1^{\text{down}})^\top a(\boldsymbol{X}_1^{\text{down}})\rangle\end{aligned}$$

$$
\begin{aligned}
=&\frac{1}{n_2}\langle \mathcal{C}_{\phi_1}^{1/2}\boldsymbol{B}(\bar{\boldsymbol{W}}-\hat{\boldsymbol{W}}), \mathcal{C}_{\phi_1}^{-1/2}\phi(\boldsymbol{X}_1^{\text{down}})^\top a(\boldsymbol{X}_1^{\text{down}})\rangle\\
\leq&\sqrt{\frac{d_2}{n_2}}\|\mathcal{C}_{\phi_1}^{1/2}\boldsymbol{B}(\bar{\boldsymbol{W}}-\hat{\boldsymbol{W}})\|_F\\
\leq&1.1\frac{1}{\sqrt{n_2}}\sqrt{\frac{d_2}{n_2}}\|\phi(\boldsymbol{X}_1^{\text{down}})\boldsymbol{B}(\bar{\boldsymbol{W}}-\hat{\boldsymbol{W}})\|_F\\
=&1.1\frac{\sqrt{d_2}}{n_2}\|\Psi(\bar{\boldsymbol{W}}-\hat{\boldsymbol{W}})\|_F.
\end{aligned}
$$

Therefore plugging back to Equation (4.18) we get:

$$
\begin{aligned}
&\frac{1}{2n_2}\|\Psi(\bar{\boldsymbol{W}}-\hat{\boldsymbol{W}})\|_F^2 \leq \frac{1}{n_2}\langle\Psi(\bar{\boldsymbol{W}}-\hat{\boldsymbol{W}}), \boldsymbol{N}+\boldsymbol{E}^{\text{pre}}\bar{\boldsymbol{W}}+a(\boldsymbol{X}_1^{\text{down}})\rangle\\
&\Rightarrow \frac{1}{2n_2}\|\Psi(\bar{\boldsymbol{W}}-\hat{\boldsymbol{W}})\|_F \leq \frac{1}{2n_2}\|\boldsymbol{E}^{\text{pre}}\bar{\boldsymbol{W}}\|_F+\frac{1}{2n_2}\|P_\Psi \boldsymbol{N}\|_F+1.1\frac{\sqrt{d_2}}{n_2}.\\
&\Rightarrow \frac{1}{2\sqrt{n_2}}\|\Psi\hat{\boldsymbol{W}}-f^*_{\mathcal{H}_1}(\boldsymbol{X}_1^{\text{down}})\|_F-\|\boldsymbol{E}\bar{\boldsymbol{W}}\|_F \leq \frac{1}{\sqrt{n_2}}(1.1\sqrt{d_2}+\|\boldsymbol{E}^{\text{pre}}\bar{\boldsymbol{W}}\|+\sqrt{d_2+\log(d_2/\delta)})\\
&\Rightarrow \frac{1}{2\sqrt{n_2}}\|\Psi\hat{\boldsymbol{W}}-f^*_{\mathcal{H}_1}(\boldsymbol{X}_1^{\text{down}})\|_F \lesssim \sqrt{\frac{d_2(1+\log d_2/\delta)}{n_2}}+\frac{\epsilon_{\text{CI}}+\epsilon_{\text{pre}}}{\beta}.
\end{aligned}
$$

Finally by concentrating $\frac{1}{n_2}\Psi^\top\Psi$ to $\mathbb{E}[\tilde{\psi}(X_1)\tilde{\psi}(X_1)^\top]$ we get:

$$
\mathbb{E}[\|\hat{\boldsymbol{W}}^\top\tilde{\psi}(X_1)-f^*_{\mathcal{H}_1}(X_1)\|_2^2] \lesssim \frac{d_2(1+\log d_2/\delta}{n_2})+\frac{\epsilon_{\text{CI}}^2+\epsilon_{\text{pre}}^2}{\beta^2},
$$

with probability $1-\delta$. □

4.12.5 Principal Component Regression

Claim 4.12.11 (Approximation Error of Principle Component Analysis)**.** *Let matrix $\boldsymbol{A}=\boldsymbol{L}+\boldsymbol{E}\in\mathbb{R}^{n\times d}$ where $\boldsymbol{L}$ has rank r <size of $\boldsymbol{A}$. Let $\boldsymbol{A}_r$ be the rank-r PCA of A. Then we have: $\|\boldsymbol{A}_r-\boldsymbol{L}\|_F\leq 2\|\boldsymbol{E}\|_F$, and $\|\boldsymbol{A}_r-\boldsymbol{L}\|_2\leq 2\|\boldsymbol{E}\|_2$.*

Proof. Due to the property of PCA, $\|\boldsymbol{A}_r-\boldsymbol{A}\|_F\leq\|\boldsymbol{E}\|_F$ and $\|\boldsymbol{A}_r-\boldsymbol{A}\|_2\leq\|\boldsymbol{E}\|_2$.

$$
\begin{aligned}
\|\boldsymbol{A}_r-\boldsymbol{L}\|_2=&\|\boldsymbol{A}_r-\boldsymbol{A}+\boldsymbol{A}-\boldsymbol{L}\|_2\\
\leq&\|\boldsymbol{A}_r-\boldsymbol{A}\|_F+\|\boldsymbol{E}\|_F\\
\leq&2\|\boldsymbol{E}\|_2.
\end{aligned}
$$

Similarly we have $\|\boldsymbol{A}_r - \boldsymbol{L}\|_F \leq 2\|\boldsymbol{E}\|_F$. $\square$

This technical fact could be used to complete the proof for Remark Remark 4.4.5.

Proof of Remark Remark 4.4.5. We replace the key steps of Theorem 4.12.10.

Recall $\Psi^*, \boldsymbol{L}, \boldsymbol{E}, \boldsymbol{V}$ are defined as follows:

$\Psi^* := \psi^*(\boldsymbol{X}_1^{\text{down}})$ is the optimal representation matrix. Ψ_r is the features obtained from r-PCA of Ψ^*. $\Psi^* = \boldsymbol{L} + \boldsymbol{E}$ which is low rank plus small norm. ($\boldsymbol{L} = \phi(\boldsymbol{X}_1^{\text{down}})\mathcal{C}_{\phi_1\phi_1}^{-1}\mathcal{C}_{\phi_1\phi_{\bar{y}}}\mathcal{C}_{\phi_{\bar{y}}}^{-1}\mathcal{C}_{\phi_{\bar{y}}X_2}$ and $\boldsymbol{E} = \phi(\boldsymbol{X}_1^{\text{down}})\mathcal{C}_{\phi_1\phi_1}^{-1}\mathcal{C}_{\phi_1 X_2|\bar{Y}}$. Suppose $r = |\mathcal{Y}||\mathcal{Z}|$.) Let $\boldsymbol{V} = f^*(\boldsymbol{X}_1^{\text{down}}) \equiv f_{\mathcal{H}_1}^*(\boldsymbol{X}_1^{\text{down}}) \equiv \phi(\boldsymbol{X}_1^{\text{down}})\mathcal{C}_{\phi_1}^{-1}\mathcal{C}_{\phi_1 Y} = \boldsymbol{L}\bar{\boldsymbol{W}}$ be our target direction, where $\bar{\boldsymbol{W}} := \boldsymbol{\Sigma}_{X_2\phi_{\bar{y}}}^{\dagger}\boldsymbol{\Sigma}_{\phi_{\bar{y}}Y}$.

Due to representation learning error (finite sample in the first stage) and approximate conditional independence, the target direction $\boldsymbol{V}$ is not perfectly linear in Ψ^* or its r-PCA features Ψ.

Now with PCR we learn the linear model with $\hat{\boldsymbol{W}} \leftarrow \arg\min_{\boldsymbol{W}} \|\Psi_r\boldsymbol{W} - \boldsymbol{Y}\|_F^2$. Together with Claim 4.12.11 and the same procedure as Theorem 4.12.10 we also get that:

Let $\bar{\boldsymbol{E}} = \boldsymbol{L} - \Psi_r$ is of rank at most $2r$.

$$
\begin{aligned}
\|\Psi_r\hat{\boldsymbol{W}} - \boldsymbol{Y}\|_F^2 \leq& \|\Psi_r\bar{\boldsymbol{W}} - \boldsymbol{Y}\|_F^2 = \|\boldsymbol{N} - \bar{\boldsymbol{E}}\bar{\boldsymbol{W}}\|_F^2. \\
\implies \|\Psi_r\hat{\boldsymbol{W}} - \boldsymbol{V} - \bar{\boldsymbol{E}}\bar{\boldsymbol{W}}\|^2 \leq& 2\langle \Psi_r\hat{\boldsymbol{W}} - \boldsymbol{V} - \bar{\boldsymbol{E}}\bar{\boldsymbol{W}}, \boldsymbol{N} - \bar{\boldsymbol{E}}\bar{\boldsymbol{W}}\rangle \\
\implies \|\Psi_r\hat{\boldsymbol{W}} - \boldsymbol{V} - \bar{\boldsymbol{E}}\bar{\boldsymbol{W}}\| \leq& \|P_{[\Psi_r, \boldsymbol{L}]}\boldsymbol{N}\| + \|\bar{\boldsymbol{E}}\bar{\boldsymbol{W}}\| \\
\implies \|\Psi_r\hat{\boldsymbol{W}} - \boldsymbol{V}\| \leq& 2\|\bar{\boldsymbol{E}}\|_F\|\bar{\boldsymbol{W}}\| + \|P_{2r}\boldsymbol{N}\| \\
\lesssim& \|\boldsymbol{E}\|_F\|\bar{\boldsymbol{W}}\| + \sigma\sqrt{r}(1 + \sqrt{\log(r/\delta)}).
\end{aligned}
$$

With concentration on the downstream labeled samples we also get the result in Remark Remark 4.4.5:

$$
\mathbb{E}[\|\hat{\boldsymbol{W}}^\top\psi_r(X_1) - f_{\mathcal{H}_1}^*(X_1)\|^2] \lesssim \frac{\epsilon_{\text{CI}}^2 + \epsilon_{\text{pre}}^2}{\beta^2} + \sigma^2\frac{r(1 + \log(r/\delta))}{n_2}.
$$

Here $r = |\mathcal{Y}||\mathcal{Z}|$. $\square$

4.12.6 Proof for topic modeling example

Proof for Corollary 4.5.1. We will construct a latent variable $\bar{Y}$ such that $\epsilon_{\text{CI}} = 0$. We pick the domain of $\bar{Y}$ to be $[k]$ and the distribution $P(\bar{Y}|X_1)$ to be the distribution $\mathbb{E}[\mu|X_1] \in \Delta_{[k]}$, and define $P\left(X_2|\bar{Y}=i\right) = P\left(X_2|\mu = e_i\right)$. More specifically we have

$$P(\bar{Y}=i|X_1) = \mathbb{E}[\mu|X_1](i) = \mathbb{E}[\mu(i)|X_1] \text{ and thus } \mathbb{E}\left[\bar{Y}|X_1\right] = \mathbb{E}[\mu|X_1]$$
$$P\left(X_2|\bar{Y}=i\right) = P\left(X_2|\mu=e_i\right) \text{ and thus } \mathbb{E}\left[X_2|\bar{Y}=i\right] = \mathbb{E}[X_2|\mu=e_i]$$

To show $\epsilon_{\text{CI}} = 0$, from Definition 4.4.2 we need to show $\mathbb{E}[X_2|X_1] = \mathbb{E}\left[\mathbb{E}\left[X_2|\bar{Y}\right]|X_1\right]$. Since X_2 is the bag of words representation, we know that $X_2 = \frac{2}{N}\sum_{i=N/2+1}^{N} e_{w_i}$. So for any $\mu \in \Delta_{[k]}$ we get

$$\mathbb{E}[X_2|\mu] =^{(a)} \frac{2}{N}\sum_{i=N/2+1}^{N}\mathbb{E}[e_{w_i}|\mu] =^{(b)} \frac{2}{N}\sum_{i=N/2+1}^{N} A\mu = A\mu$$

where (a) follows from linearity of expectation and (b) follows from the linearity of the probability distribution of each word given μ for topic models. Thus from the definition of $\bar{Y}$, $\mathbb{E}\left[X_2|\bar{Y}=i\right] = \mathbb{E}[X_2|\mu=e_i] = Ae_i$. To check if $\epsilon_{\text{CI}} = 0$, we compute the following

$$\begin{aligned}\mathbb{E}\left[\mathbb{E}\left[X_2|\bar{Y}\right]|X_1\right] &= \sum_{i=1}^{k}\mathbb{E}\left[X_2|\bar{Y}=i\right]P(\bar{Y}=i|X_1)\\ &= \sum_{i=1}^{k} Ae_i\ \mathbb{E}[\mu(i)|X_1] = A\sum_{i=1}^{k}\mathbb{E}[\mu(i)e_i|X_1]\\ &= \mathbb{E}[A\mu|X_1] = \mathbb{E}[\mathbb{E}[X_2|\mu]|X_1]\end{aligned}$$

Due to the topic modeling assumption and the independent sampling of words given μ, we know that $X_1 \perp X_2|\mu$ and thus $\mathbb{E}[X_2|X_1] = \mathbb{E}[\mathbb{E}[X_2|\mu]|X_1]$. Combining with the above calculation, we get that $\mathbb{E}\left[\mathbb{E}\left[X_2|\bar{Y}\right]|X_1\right] = \mathbb{E}[X_2|X_1]$, thus giving $\epsilon_{\text{CI}} = 0$. This proves points 1. and 2.

For point 3., note that $\mathbb{E}[Y|X_1] = \mathbb{E}[w^\top\mu|X_1] = w^\top\mathbb{E}[\mu|X_1] = w^\top\mathbb{E}[\bar{Y}|X_1]$.

Finally for point 4., we use the definition $1/\beta = \|\mathbf{\Sigma}_{Y\phi_{\bar{y}}}\mathbf{\Sigma}^{\dagger}_{X_2\phi_{\bar{y}}}\|_2$. For the first term, we note that $\mathbb{E}[\phi_{\bar{Y}}|\mu] = \mathbb{E}[\mathbb{E}[\phi_{\bar{Y}}|X_1]|\mu] = \mathbb{E}[\mathbb{E}[\bar{\mu}|X_1]|\mu] = \mu$

$$\mathbf{\Sigma}_{Y\phi_{\bar{y}}} = \mathbb{E}_{\mu\sim\tau}\left[Y\phi_{\bar{Y}}^\top\right] = \mathbb{E}_{\mu\sim\tau}\left[w^\top\mu\phi_{\bar{Y}}^\top\right]$$

$$= \mathbb{E}_{\mu\sim\tau}\left[w^\top \mu \mathbb{E}\left[\phi_{\bar{Y}}^\top|\mu\right]\right] = \mathbb{E}_{\mu\sim\tau}\left[w^\top \mu\mu^\top\right]$$
$$= w^\top \Gamma$$

where Γ was defined as the topic covariance $\Gamma = \mathbb{E}_{\mu\sim\tau}\left[\mu\mu^\top\right]$. The second term is

$$\mathbf{\Sigma}_{X_2\phi_{\bar{y}}} = \mathbb{E}_{\mu\sim\tau}\left[\mathbb{E}\left[X_2|\mu\right]\mathbb{E}\left[\phi_{\bar{Y}}^\top|\mu\right]\right] = \mathbb{E}_{\mu\sim\tau}\left[A\mu\mu^\top\right] = A\Gamma$$

The upper bound for $1/\beta$ can be computed as follows

$$\begin{aligned}
1/\beta &= \left\|\mathbf{\Sigma}_{Y\phi_{\bar{y}}}\mathbf{\Sigma}_{X_2\phi_{\bar{y}}}^\dagger\right\|_2 = \left\|w^\top\Gamma\left(A\Gamma\right)^\dagger\right\|_2 \\
&\leq \|w\|_2\ \lambda_{\max}(\Gamma)\ \lambda_{\max}\left(\left(A\Gamma\right)^\dagger\right) = \|w\|_2\ \lambda_{\max}(\Gamma)\ \lambda_{\min}\left(A\Gamma\right)^{-1} \\
&\leq \|w\|_2\ \lambda_{\max}(\Gamma)\ \lambda_{\min}\left(A\right)^{-1}\ \lambda_{\min}\left(\Gamma\right)^{-1} \\
&= \|w\|_2\ \frac{\lambda_{\max}(\Gamma)}{\lambda_{\min}\left(\Gamma\right)}\ \lambda_{\min}\left(A\right)^{-1} = \frac{\kappa\|w\|_2}{\lambda_{\min}\left(A\right)}
\end{aligned}$$

□

4.13 Omitted proofs on learning the conditional distribution

4.13.1 Introducing the operators on the Hilbert spaces

We first introduce all the operators. They will help us to present all the theorem of Section 4.6 in a more compact way. We let $L^2(X)$ denotes the Hilbert space of square integrable function with respect to the measure P_X, the marginal distribution of X. For instance, in our context of SSL, $L^2(X_2) = \{g : \mathbb{R}^{d_2} \to \mathbb{R}|\int g^2(x_2)dP_{X_2}(x_2) < \infty.\}$.

- Representation operator $\mathcal{T} : L^2(X_2) \to L^2(X_1)$,

$$(\mathcal{T}g)(x_1) := \mathbb{E}[g(X_2)|X_1 = x_1], \forall g \in L^2(X_2).$$

- Low rank approximation operator $\mathcal{L} : L^2(X_2) \to L^2(X_1)$,

$$(\mathcal{L}g)(x_1) = \mathbb{E}_Y[\mathbb{E}_{X_2}[g(X_2)|Y]|X_1 = x_1].$$

Under conditional independence $X_1 \perp X_2 | Y, \mathcal{T} = \mathcal{L}$.

- From the definition of $\mathcal{L}$ we can decompose it into the following two operators $\mathcal{L} = \mathcal{B} \circ \mathcal{A}$:
- $\mathcal{A} : L^2(X_2) \to L^2(Y), (\mathcal{A}g)(y) := \mathbb{E}[g(X_2)|Y = y]$
- $\mathcal{B} : L^2(Y) \to L^2(X_1), (\mathcal{B}h)(x_1) := \mathbb{E}[h(Y)|X_1 = x_1]$. Our final goal is to compute $\mathcal{B} \circ \boldsymbol{I} = \mathbb{E}[Y|X_1 = x_1]$, where $\boldsymbol{I}(y) = y$ is the identity map on $L^2(Y)$.
- $\mathcal{A}^\dagger : L^2(Y) \to L^2(X_2)$ is the inverse operator of $\mathcal{A}$. Let $\tilde{\beta} := 1/\|\mathcal{A}^\dagger\|_{\text{HS}}$. This $\tilde{\beta} \in [\sigma_k(\mathcal{A})/\sqrt{k}, \sigma_k(\mathcal{A})]$ where $\sigma_k(\mathcal{A})$ is the $(k-1)$-th maximal correlation between X_2 and Y.

- Operator that measures conditional independence: $\mathcal{E} := \mathcal{T} - \mathcal{L}$,

$$\|\mathcal{E}\|_{\text{op}} := \max_{\|g\|_{L^2(X_2)}=1} \mathbb{E}_{X_1}(\mathbb{E}[g(X_2)|X_1] - \mathbb{E}[\mathbb{E}[g(X_2)|Y]|X_1])^2 =: \tilde{\epsilon}_{\text{CI}}.$$

Theorem 4.13.1 (Theorem 4.6.1 restated)**.** *Conduct SVD on $\mathcal{T}$: find k orthonormal function $u_1, \cdots u_k$ in $L^2(X_1)$ and orthonormal function $v_1, \cdots v_k \in L^2(X_2)$ and scalars $\sigma_1, \cdots \sigma_k \in \mathbb{R}$ that minimizes:*

$$L(\{u_i\}, \{v_i\}, \{\sigma_i\}) := \max_{\|g\|_{L^2(X_2)}=1} \|\mathcal{T}g - \mathcal{T}_k g\|_{L^2(X_1)}, \text{ where } \mathcal{T}_k g := \sum_{i=1}^{k} \sigma_i \langle v_i, g \rangle_{L^2(X_2)} u_i.$$

Now treat $\psi(x_1) = [u_1(x_1), \cdots u_k(x_1)] : \mathcal{X}_1 \to \mathbb{R}^k$ as the representation. Then the approximation error of ψ satisfies:

$$\begin{aligned} e_{apx}(\psi) &:= \min_{\boldsymbol{W} \in \mathbb{R}^{k \times k}} \mathbb{E}[\|f^*(X_1) - \boldsymbol{W}^\top \psi(X_1)\|^2] \\ &\leq \sum_{y=1}^{k} \min_{g_y \in L^2(X_2)} 2(\|(\mathcal{T}_k - \mathcal{L}) \circ g_y\|^2_{L^2(X_1)} + \|\mathcal{L} \circ g_y - f_y^*\|^2_{L^2(X_1)}). \end{aligned}$$

Here f^* is the optimal function to predict the one-hot encoder of Y with X_2, i.e., $f_y^*(x_1) = \mathbb{E}[1(Y = y)|X_1 = x_1] = P(Y = y|X_1 = x_1)$.

When we set $g_y(x_2) = \mathcal{A}^\dagger \circ 1(Y = y)$, we have the following corollary:

Corollary 4.13.2 (Corollary 4.6.2 restated)**.** *In the same setting of Theorem 4.13.1, suppose the $(k-1)$-th*

maximal correlation between X_2 and Y is not zero, then we have:

$$ER_\psi(\hat{\boldsymbol{W}}) \le \tilde{O}(\frac{\tilde{\epsilon}_{CI}^2}{\tilde{\beta}^2} + \sigma^2 \frac{k}{n_2}).$$

Next we present the proof of Theorem 4.13.1, Corollary 4.6.2 and Corollary 4.6.4.

4.13.2 Proof of Theorem 4.13.1

Proof of Theorem 4.13.1. First note that the representation function $\psi : \mathcal{X}_1 \to \mathbb{R}^k$ is formed by the left singular vectors of $\mathcal{T}_k$, therefore for any vector $\mathbf{w} \in \mathbb{R}^k$, there exists a corresponding $g_{\boldsymbol{w}} \in L^2(X_2)$ such that $\psi(x_1)^\top \mathbf{w} \equiv (\Psi \circ g_{\boldsymbol{w}})(x_1)$. In the same way, $\mathcal{T}_k \circ g = \sum_{i=1}^k \sigma_i \langle v_i, g\rangle u_i = \psi^\top \boldsymbol{w}$ where $\boldsymbol{w} = \sigma_i \langle v_i, g\rangle$. Therefore for any $g \in L^2(X_2)$, there also exists a $\boldsymbol{w}$ such that $\psi(x_1)^\top \boldsymbol{w} \equiv (\mathcal{T}_k \circ g)(x_1)$.

$$\begin{aligned}
\mathrm{apx}(\psi) :=& \min_{\boldsymbol{W} \in \mathbb{R}^{k\times k}} \mathbb{E}[\|f^*(X_1) - \psi(X_1)\boldsymbol{W}\|^2] \\
=& \sum_{y=1}^k \min_{\boldsymbol{W} \in \mathbb{R}^{k\times k}} \mathbb{E}[\|f_y^*(X_1) - \psi(X_1)^\top \boldsymbol{w}_y\|^2] && (\boldsymbol{w}_y \text{ is the } y\text{-th column vector of } \boldsymbol{W}) \\
=& \sum_{y=1}^k \mathbb{E}[\|f_y^*(X_1) - (\mathcal{T}_k \circ g_{\boldsymbol{w}_y})(X_1)\|^2] \\
=& \sum_{y=1}^k \min_{g_y \in L^2(X_2)} \mathbb{E}[\|f_y^*(X_1) - (\mathcal{T}_k \circ g_y)(X_1)\|^2] \\
=& \sum_{y=1}^k \min_{g_y \in L^2(X_2)} \mathbb{E}[\|(f_y^*(X_1) - \mathcal{L} \circ g_y) - ((\mathcal{T}_k - \mathcal{L}) \circ g_y)(X_1)\|^2] \\
\le& \sum_{y=1}^k \min_{g_y \in L^2(X_2)} 2(\|(\mathcal{T}_k - \mathcal{L}) \circ g_y\|_{L^2(X_1)}^2 + \|\mathcal{L} \circ g_y - f_y^*\|_{L^2(X_1)}^2). && (\text{By AM-GM})
\end{aligned}$$

□

Claim 4.13.3. *The joint distribution $p_{X_1,X_2}(x_1, x_2)$ satisfies:*

$$\int_{X_1,X_2} p_{X_1,X_2}(x_1, x_2) 1(g_1^*(x_1) \neq g_2^*(x_2)) \le 2\alpha.$$

Let functions $w_{1,y}(x_1) = 1(g_1^*(x_1) = y) \in L^2(\mathcal{X}_1)$, and $w_{2,y}(x_2) = 1(g_2^*(x_2) = y) \in L^2(\mathcal{X}_2), \forall y \in [k]$. Then we

have that:

$$\sum_y \langle \mathcal{T} w_{2,y}, w_{1,y} \rangle \geq 1 - 2\alpha.$$

Proof.

$$\begin{aligned}
&\int_{X_1,X_2} p_{X_1,X_2}(x_1,x_2) 1(g(x_1) \neq g(x_2)) \\
=&\int_{X_1,X_2} \int_Y p_{X_1,x_2,Y}(x_1,x_2,y) 1(g_1^*(x_1) \neq g_2^*(x_2)) \\
\leq&\int_{X_1,X_2} \int_Y p_{X_1,x_2,Y}(x_1,x_2,y) \left(1(g_1^*(x_1) \neq y) + 1(g_2^*(x_2) \neq y)\right) \\
=&\int_{X_1,Y} p_{X_1,Y}(x_1,y) 1(g_1^*(x_1 \neq y)) + \int_{X_2,Y} p_{X_2,Y}(x_2,y) 1(g_2^*(x_2) \neq y) \\
=&P(g_1^*(x_1) \neq y) + P(g_2^*(x_2) \neq y) \leq 2\alpha. \qquad (4.19)
\end{aligned}$$

Meanwhile,

$$\begin{aligned}
&\sum_y \langle \mathcal{T} w_{2,y}, w_{1,y} \rangle \\
=&\sum_y \int_{X_1} \left(\int_{X_2} T(x_1,x_2) w_{2,y}(x_2) p_{X_2}(x_2) dx_2 \right) w_{1,y}(x_1) p_{X_1}(x_1) dx_1 \\
=&\sum_y \int_{X_1,X_2} 1(g_1^*(x_1) = y) 1(g_2^*(x_2) = y) p_{X_1,X_2}(x_1,x_2) \qquad \text{(since } T(x_1,x_2) := \tfrac{p_{X_1,X_2}(x_1,x_2)}{p_{X_1}(x_1) p_{X_2}(x_2)}\text{)} \\
=&\int_{g_1^*(X_1)=g_2^*(X_2)} p_{X_1,X_2}(x_1,x_2) \\
=&1 - \int_{X_1,X_2} p_{X_1,X_2}(x_1,x_2) 1(g(x_1) \neq g(x_2)) \\
\geq&1 - 2\alpha. \qquad \text{(from Ineq. Equation (4.19))}
\end{aligned}$$

□

Claim 4.13.4. *The top eigenvalue of T is* 1.

Proof. First we show that $\|\mathcal{T}\|_{op} := \max_{u \neq 0} \frac{\|\mathcal{T}u\|_{L^2(X_1)}}{\|u\|_{L^2(X_2)}} \leq 1$. For any $u \in L^2(R^d)$, we have that

$$\|\mathcal{T}u\|^2 = \|\mathbb{E}[u(X_2)|X_1]\|^2_{L^2(X_1)}$$

$$
\begin{aligned}
&= \int_{x_1} \mathbb{E}[u(X_2)|X_1]^2 p_{X_1}(x_1) dx_1 \\
&\le \int_{x_1} \mathbb{E}[u^2(X_2)|X_1] p_{X_1}(x_1) dx_1 && \text{(Jensen's inequality that } \mathbb{E}^2[X] \le \mathbb{E}[X^2]\text{)} \\
&= \mathbb{E}[u^2(X_2)] = \|u\|^2_{L^2(X_2)}.
\end{aligned}
$$

Second, let $u(x_2) \equiv 1$ and $v(x_1) \equiv 1$, we have $\int_{x_1} T(x_1, x_2) u(x_2) dx_2 = 1 = v(x_1)$. Therefore we have $\|Tu\|_{L^2(X_1)} = 1$ for $u = 1$ and $\|u\|_{L^2(X_2)} = 1$. Therefore $\|T\|_{\text{op}} = 1$. □

Lemma 4.13.5. *Let $w_{1,y}, w_{2,y}, \forall y \in [k]$ be the same from Claim 4.13.3. Then we have:*

$$\sum_y \langle \mathcal{L} w_{2,y}, w_{1,y} \rangle \ge 1 - 2\alpha.$$

Therefore $\sum_y \|\mathcal{L} w_{2,y} - w_{1,y}\|^2 \le 4\alpha$.

Proof.

$$
\begin{aligned}
&\sum_y \langle \mathcal{L} w_{2,y}, w_{1,y} \rangle \\
=&\sum_y \sum_h \int_{x_1,x_2} p(x_1|h) p(x_2|h) p(h) 1(g_1^*(x_1) = y) 1(g_2^*(x_2) = y) dx 2 dx_1 \\
=&\sum_h \int_{x_1,x_2} p(x_1|h) p(x_2|h) p(h) 1(g_1^*(x_1) = g_2^*(x_2)) dx 2 dx_1 \\
=&\sum_h \int_{x_1,x_2} p(x_1|h) p(x_2|h) p(h) (1 - 1(g_1^*(x_1) \ne g_2^*(x_2))) dx 2 dx_1 \\
=&\sum_h \int_{x_1,x_2} p(x_1|h) p(x_2|h) p(h) dx_2 dx_1 - \sum_h \int_{x_1,x_2} p(x_1|h) p(x_2|h) p(h) 1(g_1^*(x_1) \ne g_2^*(x_2)) dx_2 dx_1 \\
=&1 - \sum_h \int_{x_1,x_2} p(x_1|h) p(x_2|h) p(h) 1(g_1^*(x_1) \ne g_2^*(x_2)) dx_2 dx_1.
\end{aligned}
$$

$$
\begin{aligned}
&\sum_h \int_{x_1,x_2} p(x_1|h) p(x_2|h) p(h) 1(g_1^*(x_1) \ne g_2^*(x_2)) dx_2 dx_1 \\
\le&\sum_y \int_{x_1,x_2} p(x_1|y) p(x_2|y) p(y) (1(g_1^*(x_1) \ne y) + 1(g_2^*(x_2) \ne y)) dx 2 dx_1 \\
=&\sum_y \left(\int_{x_1} p(x_1|y) \int_{x_2} p(x_2, h) 1(g_2^*(x_2) \ne y) dx_1 + \int_{x_2} p(x_2|y) \int_{x_1} p(x_1, h) 1(g_1^*(x_1) \ne y) dx_2 \right)
\end{aligned}
$$

$$= \sum_y (P_{X_1,Y}(g_1^*(x_1 \neq y)) + P_{X_1,Y}(g_1^*(x_1 \neq y)))$$

$$\leq 2\alpha.$$

Therefore $\sum_y \langle \mathcal{L} w_{2,y}, w_{1,y} \rangle \geq 1 - 2\alpha$. $\sum_y \|\mathcal{L} w_{2,y} - w_{1,y}\|^2 = \sum_y (\|\mathcal{L} w_{2,y}\|^2 + \|w_{1,y}\|^2 - 2\langle w_{1,y}, \mathcal{L} w_{2,y} \rangle) \leq 2 - 2(1 - 2\alpha) = 4\alpha$. □

Lemma 4.13.6. *Let $T_k(x_1, x_2)$ be the rank-k approximation of $T(x_1, x_2)$, i.e., $T_k(x_1, x_2) = \sum_{i=1}^k \sigma_i u_i(x_1) v_i(x_2)$, where $u_i \in L^2(\mathcal{X}_1), v_i \in L^2(\mathcal{X}_2)$. Then with the same definition of $w_{1,y}$ and $w_{2,y}$ as Claim Claim 4.13.3, we have that:*

$$\sum_{y=1}^k \|\mathcal{T}_k w_{2,y} - w_{1,y}\|^2 \leq \frac{16\alpha}{1 - \lambda_{k+1}^2},$$

where λ_{k+1} is the $(k+1)$-th singular value of $\mathcal{T}$, i.e., the k-th maximal correlation between X_1 and X_2

Proof. First, we have that $\sum_y \mathbb{E}[w_{2,y}^2(X_2)] = \sum_y P_{X_2}(g_2^*(X_2) = y) = 1$.

Second from Claim Claim 4.13.4 we know that $\|T\|_{op} := \max_{\|u\|=1} \|Tu\| = 1$. Also, as we defined that $T = L + E$ with L of rank k and $\tilde{\epsilon}_{\text{CI}} := \|E\|$, we have $|\lambda_{k+1}| \leq \epsilon_{\text{CI}}$.

Write the full decomposition of T as $T(x_1, x_2) = \sum_{i=1}^\infty \lambda_i u_i(x_1) v_i(x_2)$. We have that:

$$\begin{aligned} 1 - 2\alpha \leq & \sum_y \langle \mathcal{T} w_{2,y}, w_{1,y} \rangle \\ \leq & \sqrt{\sum_y \|\mathcal{T} w_{2,y}\|^2} \sqrt{\sum_y \|w_{1,y}\|^2}. \end{aligned}$$

Therefore $\sqrt{\sum_y \|\mathcal{T} w_{2,y}\|^2} \geq 1 - 2\alpha$.

Meanwhile,

$$\begin{aligned} \sum_y \|\mathcal{T} w_{2,y}\|^2 = & \sum_y (\|\mathcal{T}_k w_{2,y}\|^2 + \|(\mathcal{T} - \mathcal{T}_k) w_{2,y}\|^2) \\ = & \sum_y (\|\mathcal{T}_k P_{\mathcal{T}_k} w_{2,y}\|^2 + \|(\mathcal{T} - \mathcal{T}_k) P_{\mathcal{T}_k}^\perp w_{2,y}\|^2) \end{aligned}$$

$$\leq \sum_y (\|P_{\mathcal{T}_k} w_{2,y}\|^2 + \lambda_{k+1}^2(\|w_{2,y}\|^2 - \|P_{\mathcal{T}_k} w_{2,y}\|^2) \quad (\text{since } \|\mathcal{T}\|_{\text{op}} = 1 \text{ and } \|\mathcal{T} - \mathcal{T}_k\| = \lambda_{k+1})$$

$$=(1-\lambda_{k+1}^2)(\sum_y \|P_{\mathcal{T}_k} w_{2,y}\|^2) + \lambda_{k+1}^2 \qquad (\text{since } \textstyle\sum_y \|w_{2,y}\|^2 = 1.)$$

Therefore $\sum_y \|P_{\mathcal{T}_k} w_{2,y}\|^2 \geq \frac{(1-2\alpha)^2 - \lambda_{k+1}^2}{1-\lambda_{k+1}^2}$ and

$$\begin{aligned}\sum_y \|(\mathcal{T} - \mathcal{T}_k) w_{2,y}\|^2 \leq& \lambda_{k+1}^2(1 - \sum_y \|P_{\mathcal{T}_k} w_{2,y}\|^2) \\ \leq& \lambda_{k+1}^2(1 - \frac{(1-2\alpha)^2 - \lambda_{k+1}^2}{1-\lambda_{k+1}^2}) \\ =& \frac{4\alpha(1-\alpha)\lambda_{k+1}^2}{1-\lambda_{k+1}^2}.\end{aligned}$$

Finally, on one hand we have

$$\begin{aligned}&\sum_y \|\mathcal{T} w_{2,y} - w_{1,y}\|^2 = \sum_y \|\mathcal{T} w_{2,y}\|^2 + \|w_{1,y}\|^2 - 2\langle \mathcal{T} w_{2,y}, w_{1,y}\rangle \\ &\leq 2 - 2(1-2\alpha) = 4\alpha.\end{aligned}$$

On the other hand we have:

$$\begin{aligned}\sqrt{\sum_y \|\mathcal{T}_k w_{2,y} - w_{1,y}\|^2} \leq& \sqrt{\sum_y \|\mathcal{T}_k w_{2,y} - w_{1,y}\|^2} + \sqrt{\sum_y \|(\mathcal{T} - \mathcal{T}_k) w_{2,y}\|^2} \\ \leq& 2\sqrt{\alpha} + \sqrt{\frac{4\alpha(1-\alpha)}{1-\lambda_{k+1}^2}} \\ \leq& \frac{4\sqrt{\alpha}}{\sqrt{1-\lambda_{k+1}^2}}.\end{aligned}$$

Therefore $\sum_y \|\mathcal{T}_k w_{2,y} - w_{1,y}\|^2 \leq \frac{16\alpha}{1-\lambda_{k+1}^2}$. □

Proof of Corollary 4.13.2. This is the corollary from Theorem 4.13.1 by taking $g_i^*(y) = \mathcal{A}^\dagger \circ 1(y = i)$ such that $\mathcal{L} \circ g_i^* \equiv f_i^*, \forall i \in [k]$. This is because $\mathcal{L} = \mathcal{B} \circ \mathcal{A}$, and $\mathcal{L} \circ \mathcal{A}^\dagger \circ 1(y = i) = \mathcal{B} \circ \boldsymbol{I} = \mathbb{E}[Y = i|X_1] = f_y^*$.

Therefore the second term is 0 in Theorem 4.13.1 and it remains to prove that the first term is small.

Notice

$$\begin{aligned}
&\mathbb{E}_{X_1}\|\bar{\mathcal{E}} \circ g^*(X_1)\|^2 \\
=&\|\bar{\mathcal{E}} \circ g^*\|^2_{L^2(X_1)} \\
\leq&\|\bar{\mathcal{E}}\|^2_{\text{op}}\|\mathcal{A}^\dagger\|^2_{\text{op}}\sum_y \|1(Y=y)\|^2_{L^2(Y)} \\
\lesssim&\tilde{\epsilon}^2_{\text{CI}}/\tilde{\beta}^2.
\end{aligned}$$

Therefore the approximation error is upper bounded by $\tilde{\epsilon}^2_{\text{CI}}/\tilde{\beta}^2$.

□

Proof of Corollary 4.6.4. With Theorem 4.13.1 and we take $g_y(x_2) = w_{2,y}(x_2) = 1(g_2^*(x_2) = y), \forall y \in [k]$ as in Lemma 4.13.5. We only need to upper bound

$$\mathbb{E}_{X_1}\|f_y^* - \mathcal{L} \circ w_{2,y}\|^2 + \|(\mathcal{L} - \mathcal{T}_k) \circ w_{2,y}\|^2.$$

Notice that

$$\begin{aligned}
&\sum_y \mathbb{E}_{X_1}\|(\mathcal{L} - \mathcal{T}_k)w_{2,y}\|^2 \\
=&\sum_y \mathbb{E}_{X_1}\|(\mathcal{L} \circ w_{2,y} - w_{1,y}) + (w_{1,y} - \mathcal{T}_k \circ w_{2,y})\|^2 \\
\leq&2\sum_y \mathbb{E}_{X_1}(\|\mathcal{L} \circ w_{2,y} - w_{1,y}\|^2 + \|(w_{1,y} - \mathcal{T}_k \circ w_{2,y})\|^2) \\
\leq&\frac{16\alpha}{1-\lambda_k^2} + 4\alpha. && \text{(from Lemma 4.13.6 and Lemma 4.13.5)}
\end{aligned}$$

Meanwhile, the other term is

$$\begin{aligned}
&\sum_y \mathbb{E}_{X_1}\|f_y^* - \mathcal{L} \circ w_{2,y}\|^2 \\
\leq&2\sum_y \mathbb{E}_{X_1}\|f_y^* - w_{1,y}\|^2 + \|w_{1,y} - \mathcal{L} \circ w_{2,y}\|^2 \\
\leq&2\sum_y \mathbb{E}_{X_1}\|f_y^* - w_{1,y}\|^2 + 8\alpha && \text{(from Lemma 4.13.5)}
\end{aligned}$$

$$
\begin{aligned}
=&8\alpha + 2\sum_y \int_{x_1} (p(y|x_1) - 1(g_1^*(x_1) = y))^2 p_{X_1}(x_1) dx_1 \\
=&8\alpha + 2\sum_y \int_{x_1} p^2(y|x_1) p_{X_1}(x_1) + 1(g_1^*(x_1) = y)^2 p_{X_1}(x_1) - 2 \cdot 1(g_1^*(x_1) = y) p(y|x_1) p_{X_1}(x_1) dx_1 \\
\leq&8\alpha + 2\sum_y \int_{x_1} p(y|x_1) p_{X_1}(x_1) + 1(g_1^*(x_1) = y) p_{X_1}(x_1) - 2 \cdot 1(g_1^*(x_1) = y) p(y|x_1) p_{X_1}(x_1) dx_1 \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\text{ since } p(y|x_1) \leq 1) \\
=&8\alpha + 2(2 - 2\sum_y 1(g_1^*(x_1) = y) p(y|x_1) p_{X_1}(x_1) dx_1) = 8\alpha + 4P_{X_1,Y}(g_1^*(x_1) \neq y) \\
\leq&12\alpha \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\text{since Bayes error is bounded by } \alpha.)
\end{aligned}
$$

Altogether we have the approximation error is upper bounded by $O(\frac{\alpha}{1-\lambda_k^2})$.

□

4.14 General results and comparison to multi-view redundancy

We now show a more general form of our results and also connect the multi-view redundancy assumption from Tosh et al. [2021a] to ours.

4.14.1 General results

We first note that all our results hold for a generalized version of Assumption 4.4.1 and Definition 4.4.2 that we state below.

Assumption 4.14.1. *Suppose $\bar{Y}$ with $|\bar{Y}| \leq m$ is a discrete latent variable that satisfies*

1. *$\bar{Y}$ makes X_1 and X_2 approximately CI as in Definition 4.4.2, i.e.*

$$\epsilon_{CI}^2 := \mathbb{E}_{X_1}\left[\|\mathbb{E}[X_2|X_1] - \mathbb{E}_{\bar{Y}}[\mathbb{E}[X_2|\bar{Y}]|X_1]\|^2\right]$$

2. *$\bar{Y}$ also makes X_1 and Y approximately CI with*

$$\epsilon_{\bar{Y}}^2 := \mathbb{E}_{X_1}\left[\|\mathbb{E}[Y|X_1] - \mathbb{E}_{\bar{Y}}[\mathbb{E}[Y|\bar{Y}]|X_1]\|^2\right]$$

3. $\mathbf{\Sigma}_{\phi_{\bar{Y}} X_2}$ *is full column rank and* $\|\mathbf{\Sigma}_{Y\phi_{\bar{Y}}}\mathbf{\Sigma}^{\dagger}_{X_2\phi_{\bar{Y}}}\|_2 = 1/\beta$*, where* $A^{\dagger}$ *is pseudo-inverse, and* $\phi_{\bar{Y}}$ *is the one-hot embedding for* $\bar{Y}$*.*

Note that our assumptions from the main paper are a special case of Assumption 4.14.1, with $\epsilon_{\bar{Y}} = 0$ being satisfied automatically as $\bar{Y} = [Y, Z]$ is explicitly defined to contain Y in it. Unlike Assumption 4.4.1, we do not need Y to be a discrete variable, but just need $\bar{Y}$ to be discrete. We state the generalization of Theorem 4.4.4 below

Theorem 4.14.2. *For a fixed* $\delta \in (0,1)$*, under Assumptions Assumption 4.14.1, Assumption 4.4.3 for* $\tilde{\psi}$ *and* ψ^* *and Assumption 4.3.4 for non-universal feature maps, if* $n_1, n_2 \gg \rho^4(d_2 + \log 1/\delta)$*, and we learn the pretext tasks such that:* $\mathbb{E}\|\tilde{\psi}(X_1) - \psi^*(X_1)\|_F^2 \leq \epsilon^2_{pre}$*. Then the generalization error for downstream task w.p.* $1-\delta$ *is:*

$$\mathbb{E}_{X_1}\left[\|\mathbb{E}[Y|X_1] - \hat{\boldsymbol{W}}^{\top}\tilde{\psi}(X_1)\|_2^2\right] \leq \tilde{\mathcal{O}}\left(\sigma^2\frac{d_2}{n_2} + \frac{\epsilon^2_{CI}}{\beta^2} + \frac{\epsilon^2_{pre}}{\beta^2} + \epsilon^2_{\bar{Y}}\right) \tag{4.20}$$

The result is pretty much the same as Theorem 4.4.4, except for an additional term of $\epsilon^2_{\bar{Y}}$. The proof is also very similar, the difference being that $\mathbb{E}[\mathbb{E}[Y|\bar{Y}]|X_1]$ can now be expressed as a linear function of ψ^* instead of $\mathbb{E}[Y|X_1]$, and the additional error incurred during to the mismatch between $\mathbb{E}[Y|X_1]$ and $\mathbb{E}[\mathbb{E}[Y|\bar{Y}]|X_1]$ that is $\epsilon^2_{\bar{Y}}$ will be incurred.

4.14.2 Multi-view redundancy

We show guarantees for our algorithm under the assumption from Tosh et al. [2021a] in the following special case that satisfies: (1) X_1 and X_2 are *exactly* CI given $\bar{Y}$ (thus $\epsilon_{\text{CI}} = 0$), (2) the variation in the target Y is small given X_1 and X_2. The assumption from Tosh et al. [2021a], in our setting, is equivalent to saying that ϵ_{X_1} and ϵ_{X_2} are small, where

$$\epsilon^2_{X_i} = \mathbb{E}\left[\|\mathbb{E}[Y|X_i] - \mathbb{E}[Y|X_1, X_2]\|^2\right], \quad i \in \{1,2\}$$

A similar assumption of multi-view redundancy also appears in Tsai et al. [2020]; however they state it in terms of information-theoretic quantities instead. We will show that these assumptions are also almost sufficient to show results in our setting. In particular we show that if $Y|X_1, X_2$ is almost deterministic (which makes sense for a many regression tasks) and if $\epsilon^2_{X_2}$ is small, then $\epsilon_{\bar{Y}}$ defined in the previous subsection will

be small and thus we have meaningful guarantees.

Lemma 4.14.3. *Let $\sigma_Y^2 = \mathrm{Var}[Y|X_1, X_2]$ be the variance of Y. $\bar{Y}$ is as defined in Assumption 4.14.1 with the extra condition that X_1 and X_2 are exactly CI given $\bar{Y}$. Then we have*

$$\epsilon_{\bar{Y}} \leq \sqrt{2}(\sigma_Y + \epsilon_{X_2})$$

Plugging this into Theorem 4.14.2 will give us the desired result. Note however that we did not even use the fact that ϵ_{X_1} is small. Using this part of the assumption, we can get an even stronger result that shows that even though our learned representation will only X_1, if will still predict $Y|X_1, X_2$ well.

Corollary 4.14.4. *For a fixed $\delta \in (0,1)$, under Assumptions Assumption 4.14.1, Assumption 4.4.3 for $\tilde{\psi}$ and ψ^* and Assumption 4.3.4 for non-universal feature maps, if $n_1, n_2 \gg \rho^4(d_2 + \log 1/\delta)$, and we learn the pretext tasks such that: $\mathbb{E}\|\tilde{\psi}(X_1) - \psi^*(X_1)\|_F^2 \leq \epsilon_{pre}^2$. Then the generalization error for downstream task w.p. $1-\delta$ is:*

$$\mathbb{E}_{X_1,X_2}\left[\|\mathbb{E}[Y|X_1,X_2] - \hat{\boldsymbol{W}}^\top \tilde{\psi}(X_1)\|_2^2\right] \leq \tilde{\mathcal{O}}\left(\sigma^2 \frac{d_2}{n_2} + \frac{\epsilon_{pre}^2}{\beta^2} + \epsilon_{\bar{X}_1}^2 + \epsilon_{\bar{X}_2}^2 + \sigma_Y^2\right)$$

Thus we see that the assumption from Tosh et al. [2021a] is strong enough for us to be able to show stronger results than just our assumption. We complete this section by proving Lemma 4.14.3

Lemma 4.14.3. We will also make use of the following lemma that is easily proved using Cauchy-Schwarz inequality

Lemma 4.14.5. *For random variables $Z_1, \ldots, Z_n$ for which $\mathbb{E}[\|Z_i\|^2] < \infty$ for every $i \in [n]$, we have*

$$\mathbb{E}[\|Z_1 + \cdots + Z_n\|^2] \leq \left(\sqrt{\mathbb{E}[\|Z_1\|^2]} + \cdots + \sqrt{\mathbb{E}[\|Z_n\|^2]}\right)^2$$

The proof follows from the following sequence of inequalities that uses Jensen's inequality, conditional independence of X_1 and X_2 and the above lemma. For simplicity we assume that Y is a scalar random variable, the proof is the same for vector values Y, except squared values will replaced by norm squared

values.

$$\begin{aligned}
\epsilon_{\bar{Y}}^2 &= \mathbb{E}_{X_1}\left[(\mathbb{E}[Y|X_1] - \mathbb{E}_{\bar{Y}}[\mathbb{E}[Y|\bar{Y}]|X_1])^2\right] = \mathbb{E}_{X_1}\left[(\mathbb{E}_{\bar{Y}}[\mathbb{E}[Y|\bar{Y}, X_1]|X_1] - \mathbb{E}_{\bar{Y}}[\mathbb{E}[Y|\bar{Y}]|X_1])^2\right] \\
&\leq \mathbb{E}_{X_1,\bar{Y}}\left[(\mathbb{E}[Y|X_1, \bar{Y}] - \mathbb{E}[Y|\bar{Y}])^2\right] \\
&= \mathbb{E}_{\bar{Y}}\mathbb{E}_{X_1|\bar{Y}}\mathbb{E}_{X_1'|\bar{Y}}\left[(\mathbb{E}[Y|X_1, \bar{Y}] - \mathbb{E}[Y|X_1', \bar{Y}])^2\right] \\
&= \frac{1}{2}\mathbb{E}_{\bar{Y}}\mathbb{E}_{X_1|\bar{Y}}\mathbb{E}_{X_1'|\bar{Y}}\left[(\mathbb{E}_{X_2}[\mathbb{E}[Y|X_1, X_2, \bar{Y}]|\bar{Y}] - \mathbb{E}_{X_2}[\mathbb{E}[Y|X_1', X_2, \bar{Y}]|\bar{Y}])^2\right] \\
&\leq \frac{1}{2}\mathbb{E}_{\bar{Y}}\mathbb{E}_{X_1|\bar{Y}}\mathbb{E}_{X_1'|\bar{Y}}\mathbb{E}_{X_2|\bar{Y}}\left[(\mathbb{E}[Y|X_1, X_2, \bar{Y}] - \mathbb{E}[Y|X_1', X_2, \bar{Y}])^2\right] \\
&= \frac{1}{2}\mathbb{E}\left[(Z_1 + Z_2 + Z_3 + Z_4)^2\right]
\end{aligned}$$

where $Z_1 = \mathbb{E}[Y|X_1, X_2, \bar{Y}] - \mathbb{E}[Y|X_1, X_2]$, $Z_2 = -\mathbb{E}[Y|X_1', X_2, \bar{Y}] + \mathbb{E}[Y|X_1', X_2]$, $Z_3 = \mathbb{E}[Y|X_1, X_2] - \mathbb{E}[Y|X_2]$ and $Z_4 = -\mathbb{E}[Y|X_1', X_2] + \mathbb{E}[Y|X_2]$. The first and third inequality follow from Jensen's inequality, second inequality follows from $\mathbb{E}[(X - \mathbb{E}[X])^2] = \frac{1}{2}\mathbb{E}[(X - X')^2]$, and the third equality follows from the CI assumption.

We will bound $\mathbb{E}[Z_1^2] = \mathbb{E}[Z_2^2] \leq \mathbb{E}[(\mathbb{E}[Y|X_1, X_2, \bar{Y}] - \mathbb{E}[Y|X_1, X_2])^2] \leq \mathbb{E}[(Y - \mathbb{E}[Y|X_1, X_2])^2] = \sigma_Y^2$ again from Jensen's inequality. Z_3 and Z_4 can be handled by observing that $\mathbb{E}[Z_3^2] = \mathbb{E}[Z_4^2] = \mathbb{E}[(\mathbb{E}[Y|X_1, X_2] - \mathbb{E}[Y|X_2])^2] = \epsilon_{X_2}^2$.

Thus using the above lemma, we get the desired upper bound on $\epsilon_{\bar{Y}}$. □

4.14.3 Showing $\mathbb{E}[Y|X_1] \approx \mathbb{E}[Y|X_1, X_2]$

Our main result Theorem 4.4.4 shows that self-supervised learning can help approximate $\mathbb{E}[Y|X_1]$ as a linear function of the learned features $\tilde{\psi}$. In practice, however, it is more common to predict the label Y using the entire input $X = (X_1, X_2)$ rather than just X_1. We show here that learning $\mathbb{E}[Y|X_1]$ is sufficient, under mild assumptions on the task being solved: the Bayes error of the classification task (X_1, Y) is low. We first upper bound the discrepancy between $\mathbb{E}[Y|X_1]$ and $\mathbb{E}[Y|X_1, X_2]$ based on the Bayes error rate.

Lemma 4.14.6. *Suppose* $\|Y\| \leq 1$ *and* $k = |\mathcal{Y}|$. *Denote the Bayes error for distribution* $P_{X_1,Y}$ *to be* *Bayes-error*$(P_{X_1,Y}) = \mathbb{E}_{X_1}\left[1 - \max_y P(y|X_1)\right]$[9]. *Then we have*

$$\mathbb{E}_{X_1,X_2}\left[\|\mathbb{E}[Y|X_1] - \mathbb{E}[Y|X_1, X_2]\|^2\right] \leq 2k \text{ Bayes-error}(P_{X_1,Y})$$

[9]We abuse notation and use $P(y|X_1)$ instead of $P_{X_1,Y}(y|X_1)$.

We will show below (for $\mathcal{H} = \mathcal{H}_u$) that if $P_{X_1,Y}$ has low Bayes error, then predicting $\mathbb{E}[Y|X_1]$ is as good as predicting $\mathbb{E}[Y|X_1, X_2]$ up to this small additive error.

Theorem 4.14.7. *Suppose $\epsilon_{Bayes} = $ Bayes-error$(P_{X_1,Y})$ and that $\tilde{\psi}$ is ϵ^2_{pre}-optimal on the SSL task (as in Theorem 4.4.4). Under the same conditions as Theorem 4.4.4, with probability $1 - \delta$ we have*

$$\mathbb{E}_{X_1,X_2}\left[\|\mathbb{E}[Y|X_1, X_2] - \hat{\boldsymbol{W}}^\top \tilde{\psi}(X_1)\|_2^2\right] \leq \tilde{\mathcal{O}}\left(\sigma^2 \frac{d_2}{n_2} + \frac{\epsilon^2_{CI}}{\beta^2} + \frac{\epsilon^2_{pre}}{\beta^2}\right) + 2\epsilon_{Bayes}$$

Proof. The law of total expectation gives $\mathbb{E}_{X_2}[\mathbb{E}[Y|X_1, X_2]|X_1] = \mathbb{E}[Y|X_1]$, thus it is easy to obtain the following decomposition

$$\begin{aligned}\mathbb{E}_{X_1,X_2}\left[\|\mathbb{E}[Y|X_1, X_2] - \hat{\boldsymbol{W}}^\top \tilde{\psi}(X_1)\|_2^2\right] =& \mathbb{E}_{X_1}\left[\|\mathbb{E}[Y|X_1] - \hat{\boldsymbol{W}}^\top \tilde{\psi}(X_1)\|_2^2\right] \\ &+ \mathbb{E}_{X_1,X_2}\left[\|\mathbb{E}[Y|X_1] - \mathbb{E}[Y|X_1, X_2]\|_2^2\right]\end{aligned}$$

The first term can be upper bounded using Theorem 4.4.4: $\mathbb{E}_{X_1}\left[\|\mathbb{E}[Y|X_1] - \hat{\boldsymbol{W}}^\top \tilde{\psi}(X_1)\|_2^2\right] = \mathrm{ER}_{\tilde{\psi}}(\hat{\boldsymbol{W}}) \leq \tilde{\mathcal{O}}\left(\sigma^2 \frac{d_2}{n_2} + \frac{\epsilon^2_{\mathrm{CI}}}{\beta^2} + \frac{\epsilon^2_{\mathrm{pre}}}{\beta^2}\right)$. The second term is upper bounded by $2\epsilon_{\mathrm{Bayes}}$ by invoking Lemma 4.14.6, and this completes the proof □

Proof of Lemma 4.14.6. Notice the following inequality

$$\begin{aligned}\mathbb{E}_{X_1,X_2}\left[\|\mathbb{E}[Y|X_1] - \mathbb{E}[Y|X_1, X_2]\|^2\right] &= \mathbb{E}_{X_1,X_2}\left[\left\|\sum_{y\in\mathcal{Y}} y\left(P(y|X_1) - P(y|X_1, X_2)\right)\right\|^2\right] \\ &\leq |\mathcal{Y}|(\max_y \|y\|^2)\mathbb{E}_{X_1,X_2}\left[\sum_y \left(P(y|X_1) - P(y|X_1, X_2)\right)^2\right] \\ &\leq k\mathbb{E}_{X_1}\left[\mathbb{E}_{X_2}\left[\sum_y \left(P(y|X_1) - P(y|X_1, X_2)\right)^2 \mid X_1\right]\right]\end{aligned}$$

where the first inequality follows from Cauchy-Schwartz and second inequality follows from $\|Y\| \leq 1$. Thus the problem reduces to bounding the inner expectation for every X_1. We first note that for every X_1, y, we have $P(y|X_1) = \mathbb{E}_{X_2}[P(y|X_1, X_2)|X_1]$ from the law of total expectation. This gives

$$\mathbb{E}_{X_2}\left[\sum_y \left(P(y|X_1) - P(y|X_1, X_2)\right)^2 \mid X_1\right] = \sum_y \mathbb{E}_{X_2}\left[P(y|X_1, X_2)^2|X_1\right] - P(y|X_1)^2$$

$$\leq \sum_y \mathbb{E}_{X_2}\left[P(y|X_1,X_2)|X_1\right] - P(y|X_1)^2 = \mathbb{E}_{X_2}\left[\sum_y P(y|X_1,X_2)|X_1\right] - \sum_y P(y|X_1)^2$$
$$= 1 - \sum_y P(y|X_1)^2 \leq 1 - \max_y P(y|X_1)^2 \leq 2(1 - \max_y P(y|X_1))$$

where the first inequality follows because $P(y|X_1, X_2) \in [0, 1]$ and second follows trivially and third follows from $1 - x^2 \leq 2(1 - x)$ for $x \in [0, 1]$. Combining everything, we get $\mathbb{E}_{X_1,X_2}\left[\|\mathbb{E}[Y|X_1] - \mathbb{E}[Y|X_1, X_2]\|^2\right] \leq 2k\mathbb{E}_{X_1}\left[1 - \max_y P(y|X_1)\right] = 2k$ Bayes-error$(P_{X_1,Y})$, thus proving the result. □

4.15 Theoretical analysis for classification tasks

4.15.1 Classification tasks

We now consider the benefit of learning ψ from a class $\mathcal{H}_1$ on linear classification task for label set $\mathcal{Y} = [k]$. The performance of a classifier is measured using the standard logistic loss

Definition 4.15.1. *For a task with $\mathcal{Y} = [k]$, classification loss for a predictor $f : \mathcal{X}_1 \to \mathbb{R}^k$ is*

$$\ell_{clf}(f) = \mathbb{E}[\ell_{log}(f(X_1), Y)] \text{ , where } \ell_{log}(\hat{y}, y) = \left[-\log\left(\frac{e^{\hat{y}_y}}{\sum_{y'} e^{\hat{y}_{y'}}}\right)\right]$$

The loss for representation $\psi : \mathcal{X}_1 \to \mathbb{R}^{d_1}$ and linear classifier $\boldsymbol{W} \in \mathbb{R}^{k \times d_1}$ is denoted by $\ell_{clf}(\boldsymbol{W}\psi)$.

We note that the function $\ell_{\log}$ is 1-Lipschitz in the first argument. The result will also hold for the hinge loss $\ell_{\text{hinge}}(\hat{y}, y) = (1 - \hat{y}_y + \max_{y' \neq y} \hat{y}_{y'})_+$ which is also 1-Lipschitz, instead of $\ell_{\log}$.

We assume that the optimal regressor $f^*_{\mathcal{H}_1}$ for one-hot encoding also does well on linear classification.

Assumption 4.15.2. *The best regressor for 1-hot encodings in $\mathcal{H}_1$ does well on classification, i.e. $\ell_{clf}(\gamma f^*_{\mathcal{H}_1}) \leq \epsilon_{one\text{-}hot}$ is small for some scalar γ.*

Remark 4.15.3. *Note that if $\mathcal{H}_1$ is universal, then $f^*_{\mathcal{H}_1}(\boldsymbol{x}_1) = \mathbb{E}[Y|X_1 = \boldsymbol{x}_1]$ and we know that $f^*_{\mathcal{H}_1}$ is the Bayes-optimal predictor for binary classification. In general one can potentially predict the label by looking at $\arg\max_{i \in [k]} f^*_{\mathcal{H}_1}(\boldsymbol{x}_1)_i$. The scalar γ captures the margin in the predictor $f^*_{\mathcal{H}_1}$.*

We now show that using the classifier $\hat{\boldsymbol{W}}$ obtained from linear regression on one-hot encoding with learned

representations $\tilde{\psi}$ will also be good on linear classification. The proof is in Section 4.15

Theorem 4.15.4. *For a fixed $\delta \in (0,1)$, under the same setting as Theorem 4.4.4 and Assumption 4.15.2, we have:*

$$\ell_{clf}\left(\gamma \hat{\boldsymbol{W}} \tilde{\psi}\right) \leq \tilde{\mathcal{O}}\left(\gamma \sqrt{\sigma^2 \frac{d_2}{n_2} + \frac{\epsilon^2}{\beta^2} + \frac{\epsilon_{pre}^2}{\beta^2}}\right) + \epsilon_{one\text{-}hot},$$

with probability $1-\delta$.

Proof of Theorem 4.15.4. We simply follow the following sequence of steps

$$\begin{aligned}
\ell_{\text{clf}}\left(\gamma \hat{\boldsymbol{W}} \tilde{\psi}\right) &= \mathbb{E}[\ell_{\log}\left(\gamma \hat{\boldsymbol{W}} \tilde{\psi}(X_1), Y\right)] \\
&\leq^{(a)} \mathbb{E}\left[\ell_{\log}\left(\gamma f^*_{\mathcal{H}_1}(X_1), Y\right) + \gamma \|\hat{\boldsymbol{W}} \tilde{\psi}(X_1) - f^*_{\mathcal{H}_1}(X_1)\|\right] \\
&\leq^{(b)} \epsilon_{\text{one-hot}} + \gamma \sqrt{\mathbb{E}\left[\|\hat{\boldsymbol{W}} \tilde{\psi}(X_1) - f^*_{\mathcal{H}_1}(X_1)\|^2\right]} \\
&= \epsilon_{\text{one-hot}} + \gamma \sqrt{\text{ER}_{\tilde{\psi}}[\hat{\boldsymbol{W}}]}
\end{aligned}$$

where (a) follows because $\ell_{\log}$ is 1-Lipschitz and (b) follows from Assumption 4.15.2 and Jensen's inequality. Plugging in Theorem 4.4.4 completes the proof. □

4.16 Four different ways to use CI

In this section we propose four different ways to use conditional independence to prove zero approximation error, i.e.,

Claim 4.16.1 (informal)**.** *When conditional independence is satisfied: $X_1 \perp X_2 | Y$, and some non-degeneracy is satisfied, there exists some matrix $\boldsymbol{W}$ such that $\mathbb{E}[Y|X_1] = \boldsymbol{W}\mathbb{E}[X_2|X_1]$.*

We note that for simplicity, most of the results are presented for the jointly Gaussian case, where everything could be captured by linear conditional expectation $\mathbb{E}^L[Y|X_1]$ or the covariance matrices. When generalizing the results for other random variables, we note just replace X_1, X_2, Y by $\phi_1(X_1), \phi_2(X_2), \phi_y(Y)$ will suffice the same arguments.

4.16.1 Inverse covariance matrix

Write $\mathbf{\Sigma}$ as the covariance matrix for the joint distribution $P_{X_1X_2Y}$.

$$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}_{XX} & \mathbf{\Sigma}_{XY} \\ \mathbf{\Sigma}_{YY}^\top & \mathbf{\Sigma}_{YY} \end{bmatrix}, \quad \mathbf{\Sigma}^{-1} = \begin{bmatrix} \boldsymbol{A} & \rho \\ \rho^\top & \boldsymbol{B} \end{bmatrix}$$

where $\boldsymbol{A} \in \mathbb{R}^{(d_1+d_2)\times(d_1+d_2)}, \rho \in \mathbb{R}^{(d_1+d_2)\times k}, \boldsymbol{B} \in \mathbb{R}^{k\times k}$. Furthermore

$$\rho = \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix}; \quad \boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix}$$

for $\rho_i \in \mathbb{R}^{d_i\times k}, i = 1, 2$ and $\boldsymbol{A}_{ij} \in \mathbb{R}^{d_i\times d_j}$ for $i, j \in \{1, 2\}$.

Claim 4.16.2. *When conditional independence is satisfied, $\boldsymbol{A}$ is block diagonal matrix, i.e., $\boldsymbol{A}_{12}$ and $\boldsymbol{A}_{21}$ are zero matrices.*

Lemma 4.16.3. *We have the following*

$$\mathbb{E}[X_1|X_2] = (\boldsymbol{A}_{11} - \bar{\rho}_1\bar{\rho}_1^\top)^{-1}(\bar{\rho}_1\bar{\rho}_2{}^\top - \boldsymbol{A}_{12})X_2 \tag{4.21}$$

$$\mathbb{E}[X_2|X_1] = (\boldsymbol{A}_{22} - \bar{\rho}_2\bar{\rho}_2^\top)^{-1}(\bar{\rho}_2\bar{\rho}_1{}^\top - \boldsymbol{A}_{21})X_1 \tag{4.22}$$

$$\mathbb{E}[Y|X] = -B^{-\frac{1}{2}}(\bar{\rho}_1^\top X_1 + \bar{\rho}_2^\top X_2) \tag{4.23}$$

where $\bar{\rho}_i = \rho_i\boldsymbol{B}^{-\frac{1}{2}}$ for $i \in \{1, 2\}$. Also,

$$(\boldsymbol{A}_{11} - \bar{\rho}_1\bar{\rho}_1^\top)^{-1}\bar{\rho}_1\bar{\rho}_2^\top = \frac{1}{1 - \bar{\rho}_1^\top\boldsymbol{A}_{11}^{-1}\bar{\rho}_1}\boldsymbol{A}_{11}^{-1}\bar{\rho}_1\bar{\rho}_2^\top$$
$$(\boldsymbol{A}_{22} - \bar{\rho}_2\bar{\rho}_2^\top)^{-1}\bar{\rho}_2\bar{\rho}_1^\top = \frac{1}{1 - \bar{\rho}_2^\top\boldsymbol{A}_{22}^{-1}\bar{\rho}_2}\boldsymbol{A}_{22}^{-1}\bar{\rho}_2\bar{\rho}_1^\top$$

Proof. We know that $\mathbb{E}[X_1|X_2] = \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}X_2$ and $\mathbb{E}[X_2|X_1] = \mathbf{\Sigma}_{21}\mathbf{\Sigma}_{11}^{-1}x_1$, where

$$\mathbf{\Sigma}_{XX} = \begin{bmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22} \end{bmatrix}$$

First using $\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1} = I$, we get the following identities

$$\boldsymbol{\Sigma}_{XX}\boldsymbol{A} + \boldsymbol{\Sigma}_{XY}\rho^\top = \boldsymbol{I} \tag{4.24}$$

$$\boldsymbol{\Sigma}_{XY}^\top\boldsymbol{A} + \boldsymbol{\Sigma}_{YY}\rho^\top = 0 \tag{4.25}$$

$$\boldsymbol{\Sigma}_{XX}\rho + \boldsymbol{\Sigma}_{XY}\boldsymbol{B} = 0 \tag{4.26}$$

$$\boldsymbol{\Sigma}_{XY}^\top\rho + \boldsymbol{\Sigma}_{YY}\boldsymbol{B} = \boldsymbol{I} \tag{4.27}$$

From Equation (4.26) we get that $\boldsymbol{\Sigma}_{XY} = -\boldsymbol{\Sigma}_{XX}\rho\boldsymbol{B}^{-1}$ and plugging this into Equation (4.24) we get

$$\begin{aligned}
&\boldsymbol{\Sigma}_{XX}\boldsymbol{A} - \boldsymbol{\Sigma}_{XX}\rho\boldsymbol{B}^{-1}\rho^\top = \boldsymbol{I} \\
\implies &\boldsymbol{\Sigma}_{XX} = (\boldsymbol{A} - \rho\boldsymbol{B}^{-1}\rho^\top)^{-1} = (\boldsymbol{A} - \bar{\rho}\bar{\rho}^\top)^{-1} \\
\implies &\begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix} = \left(\begin{bmatrix} \boldsymbol{A}_{11} - \bar{\rho}_1\bar{\rho}_1^\top & \boldsymbol{A}_{12} - \bar{\rho}_1\bar{\rho}_2^\top \\ \boldsymbol{A}_{21} - \bar{\rho}_2\bar{\rho}_1^\top & \boldsymbol{A}_{22} - \bar{\rho}_2\bar{\rho}_2^\top \end{bmatrix} \right)^{-1}
\end{aligned}$$

We now make use of the following expression for inverse of a matrix that uses Schur complement: $\boldsymbol{M}/\alpha = \delta - \gamma\alpha^{-1}\beta$ is the Schur complement of α for $\boldsymbol{M}$ defined below

$$\text{If } \boldsymbol{M} = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}, \text{ then, } \boldsymbol{M}^{-1} = \begin{bmatrix} \alpha^{-1} + \alpha^{-1}\beta(\boldsymbol{M}/\alpha)^{-1}\gamma\alpha^{-1} & -\alpha^{-1}\beta(\boldsymbol{M}/\alpha)^{-1} \\ -(\boldsymbol{M}/\alpha)^{-1}\gamma\alpha^{-1} & (\boldsymbol{M}/\alpha)^{-1} \end{bmatrix}$$

For $\boldsymbol{M} = (\boldsymbol{A} - \bar{\rho}\bar{\rho}^\top)$, we have that $\boldsymbol{\Sigma}_{XX} = \boldsymbol{M}^{-1}$ and thus

$$\begin{aligned}
\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1} &= -\alpha^{-1}\beta(\boldsymbol{M}/\alpha)^{-1}((\boldsymbol{M}/\alpha)^{-1})^{-1} \\
&= -\alpha^{-1}\beta \\
&= (\boldsymbol{A}_{11} - \bar{\rho}_1\bar{\rho}_1^\top)^{-1}(\bar{\rho}_1\bar{\rho}_2^\top - \boldsymbol{A}_{12})
\end{aligned}$$

This proves Equation (4.21) and similarly Equation (4.22) can be proved.

For Equation (4.23), we know that $\mathbb{E}[Y|X = (X_1, X_2)] = \boldsymbol{\Sigma}_{YX}\boldsymbol{\Sigma}_{XX}^{-1}X = \boldsymbol{\Sigma}_{XY}^\top\boldsymbol{\Sigma}_{XX}^{-1}X$. By using Equation (4.26) we get $\boldsymbol{\Sigma}_{XY} = -\boldsymbol{\Sigma}_{XX}\rho\boldsymbol{B}^{-1}$ and thus

$$\mathbb{E}[Y|X = (X_1, X_2)] = -\boldsymbol{B}^{-1}\rho^\top\boldsymbol{\Sigma}_{XX}\boldsymbol{\Sigma}_{XX}^{-1}X$$

$$= -\boldsymbol{B}^{-1}\rho^\top X = \boldsymbol{B}^{-1}(\rho_1^\top X_1 + \rho_2^\top X_2)$$
$$= -\boldsymbol{B}^{-\frac{1}{2}}(\bar{\rho}_1^\top X_1 + \bar{\rho}_2^\top X_2)$$

For the second part, we will use the fact that $(\boldsymbol{I} - \boldsymbol{a}\boldsymbol{b}^\top)^{-1} = \boldsymbol{I} + \frac{1}{1-\boldsymbol{a}^\top \boldsymbol{b}}\boldsymbol{a}\boldsymbol{b}^\top$. Thus

$$\begin{aligned}
(\boldsymbol{A}_{11} - \bar{\rho}_1\bar{\rho}_1^\top)^{-1}\bar{\rho}_1\bar{\rho}_2 &= (\boldsymbol{I} - \boldsymbol{A}_{11}^{-1}\bar{\rho}_1\bar{\rho}_1^\top)\boldsymbol{A}_{11}^{-1}\bar{\rho}_1\bar{\rho}_2^\top \\
&= (\boldsymbol{I} + \frac{1}{1-\bar{\rho}_1^\top \boldsymbol{A}_{11}^{-1}\bar{\rho}_1}\boldsymbol{A}_{11}^{-1}\bar{\rho}_1\bar{\rho}_1)\boldsymbol{A}_{11}^{-1}\bar{\rho}_1\bar{\rho}_2^\top \\
&= \boldsymbol{A}_{11}^{-1}(I + \frac{1}{1-\bar{\rho}_1^\top \boldsymbol{A}_{11}^{-1}\bar{\rho}_1}\bar{\rho}_1\bar{\rho}_1\boldsymbol{A}_{11}^{-1})\bar{\rho}_1\bar{\rho}_2^\top \\
&= \boldsymbol{A}_{11}^{-1}(\bar{\rho}_1\bar{\rho}_2^\top + \frac{\bar{\rho}_1\boldsymbol{A}_{11}^{-1}\bar{\rho}_1}{1-\bar{\rho}_1^\top \boldsymbol{A}_{11}^{-1}\bar{\rho}_1}\bar{\rho}_1\bar{\rho}_2^\top) \\
&= \boldsymbol{A}_{11}^{-1}\bar{\rho}_1\bar{\rho}_2^\top(1 + \frac{\bar{\rho}_1\boldsymbol{A}_{11}^{-1}\bar{\rho}_1}{1-\bar{\rho}_1^\top \boldsymbol{A}_{11}^{-1}\bar{\rho}_1}) \\
&= \frac{1}{1-\bar{\rho}_1^\top \boldsymbol{A}_{11}^{-1}\bar{\rho}_1}A_{11}^{-1}\bar{\rho}_1\bar{\rho}_2^\top
\end{aligned}$$

The other statement can be proved similarly. □

Claim 4.16.4.

$$\mathbb{E}[X_2|X_1] = (\boldsymbol{A}_{22} - \bar{\rho}_2\bar{\rho}_2^\top)^{-1}\bar{\rho}_2\bar{\rho}_1^\top X_1 . \mathbb{E}[Y|X_1] = -\boldsymbol{B}^{-1/2}\bar{\rho}_1^\top X_1 - \boldsymbol{B}^{-1/2}\bar{\rho}_2^\top \mathbb{E}[X_2|X_1]$$

Therefore $\mathbb{E}[Y|X_1]$ *is in the same direction as* $\mathbb{E}[X_2|X_1]$.

4.16.2 Closed form of linear conditional expectation

Refer to Claim Claim 4.10.3 and proof of Lemma 4.10.4. As this is the simplest proof we used in our paper.

4.16.3 From law of iterated expectation

$$\begin{aligned}
\mathbb{E}^L[X_2|X_1] =& \mathbb{E}^L[\mathbb{E}^L[X_2|X_1, Y]|X_1] \\
=& \mathbb{E}\left[[\boldsymbol{\Sigma}_{X_2X_1}, \boldsymbol{\Sigma}_{X_2Y}]\begin{bmatrix}\boldsymbol{\Sigma}_{X_1X_1} & \boldsymbol{\Sigma}_{X_1Y} \\ \boldsymbol{\Sigma}_{YX_1} & \boldsymbol{\Sigma}_{YY}\end{bmatrix}^{-1}\begin{bmatrix}X_1 \\ Y\end{bmatrix} \mid X_1\right]
\end{aligned}$$

$$= \boldsymbol{A}X_1 + \boldsymbol{B}\mathbb{E}^L[Y|X_1].$$

Using block matrix inverse,

$$\begin{aligned}
\boldsymbol{A} &= (\boldsymbol{\Sigma}_{X_2X_1} - \boldsymbol{\Sigma}_{X_2Y}\boldsymbol{\Sigma}_{YY}^{-1}\boldsymbol{\Sigma}_{YX_1})(\boldsymbol{\Sigma}_{X_1X_1} - \boldsymbol{\Sigma}_{X_1Y}\boldsymbol{\Sigma}_{YY}^{-1}\boldsymbol{\Sigma}_{YX_1})^{-1} \in \mathbb{R}^{d_2 \times d_1} \\
&= \boldsymbol{\Sigma}_{X_1X_2|Y}(\boldsymbol{\Sigma}_{X_1X_1|Y})^{-1} \\
\boldsymbol{B} &= \boldsymbol{\Sigma}_{X_2Y|X_1}(\boldsymbol{\Sigma}_{YY|X_1})^{-1} \in \mathbb{R}^{d_2 \times \mathcal{Y}}.
\end{aligned}$$

Therefore in general (without conditional independence assumption) our learned representation will be $\psi(x_1) = \boldsymbol{A}x_1 + \boldsymbol{B}f^*(x_1)$, where $f^*(\cdot) := \mathbb{E}^L[Y|X_1]$.

It's easy to see that to learn f^* from representation ψ, we need A to have some good property, such as light tail in eigenspace, and B needs to be full rank in its column space.

Notice in the case of conditional independence, $\boldsymbol{\Sigma}_{X_1X_2|Y} = 0$, and $A = 0$. Therefore we could easily learn f^* from ψ if X_2 has enough information of Y such that $\boldsymbol{\Sigma}_{X_2Y|X_1}$ is of the same rank as dimension of Y.

4.16.4 From $\mathbb{E}[X_2|X_1, Y] = \mathbb{E}[X_2|Y]$

Proof. Let the representation function ψ be defined as follows, and let we use law of iterated expectation:

$$\begin{aligned}
\psi(\cdot) := \mathbb{E}[X_2|X_1] &= \mathbb{E}[\mathbb{E}[X_2|X_1, Y]|X_1] \\
&= \mathbb{E}[\mathbb{E}[X_2|Y]|X_1] && \text{(uses CI)} \\
&= \sum_y P(Y = y|X_1)\mathbb{E}[X_2|Y = y] \\
&=: f(X_1)^\top A,
\end{aligned}$$

where $f : \mathbb{R}^{d_1} \to \Delta_{\mathcal{Y}}$ satisfies $f(x_1)_y = P(Y = y|X_1 = x_1)$, and $\boldsymbol{A} \in \mathbb{R}^{\mathcal{Y} \times d_2}$ satisfies $\boldsymbol{A}_{y,:} = \mathbb{E}[X_2|Y = y]$. Here Δ_d denotes simplex of dimension d, which represents the discrete probability density over support of size d.

Let $\boldsymbol{B} = \boldsymbol{A}^\dagger \in \mathbb{R}^{\mathcal{Y} \times d_2}$ be the pseudoinverse of matrix $\boldsymbol{A}$, and we get $\boldsymbol{B}\boldsymbol{A} = \boldsymbol{I}$ from our assumption that A is

of rank $|\mathcal{Y}|$. Therefore $f(\boldsymbol{x}_1) = \boldsymbol{B}\psi(\boldsymbol{x}_1), \forall x_1$. Next we have:

$$\begin{aligned}\mathbb{E}[Y|X_1 = \boldsymbol{x}_1] =& \sum_y P(Y = y|X_1 = \boldsymbol{x}_1) \times y \\ =& \hat{\boldsymbol{Y}} f(\boldsymbol{x}_1) \\ =& (\hat{\boldsymbol{Y}}\boldsymbol{B}) \cdot \psi(X_1).\end{aligned}$$

Here we denote by $\hat{\boldsymbol{Y}} \in \mathbb{R}^{k \times \mathcal{Y}}, \hat{\boldsymbol{Y}}_{:,y} = y$ that spans the whole support $\mathcal{Y}$. Therefore let $\boldsymbol{W}^* = \hat{\boldsymbol{Y}}\boldsymbol{B}$ will finish the proof.

□

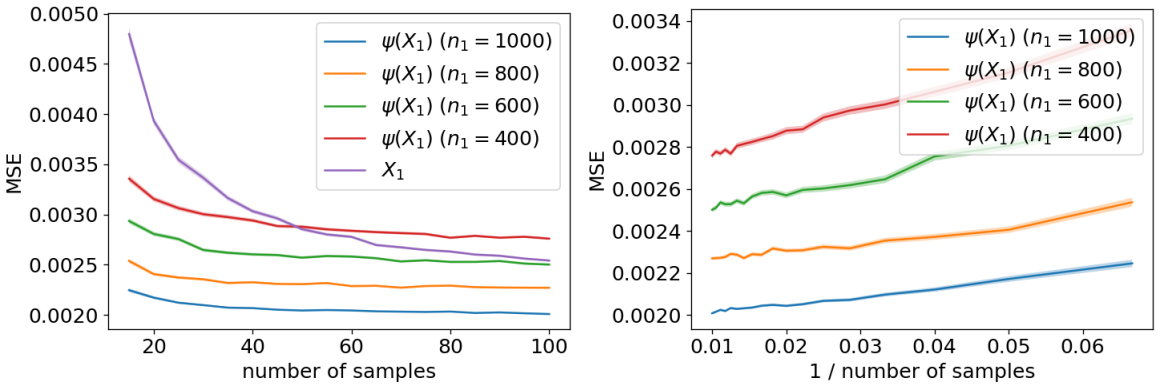


Figure 4.3: **Left**: MSE of using ψ to predict Y versus using X_1 directly to predict Y. Using ψ consistently outperforms using X_1. **Right**: MSE of ψ learned with different n_1. The MSE scale with $1/n_2$ as indicated by our analysis. Simulations are repeated 100 times, with the mean shown in solid line and one standard error shown in shadow.

4.17 Experiment details

In this section, we include more experiment setup and results.

Simulations. All the experiments are performed on a desktop computer with Intel i7-8700K, 16GB RAM. Following Theorem 4.4.4, we know that the Excessive Risk (ER) is also controlled by (1) the number of samples for the pretext task (n_1), and (2) the number of samples for the downstream task (n_2), besides k and ϵ_{CI} as discussed in the main text. In this simulation, we enforce strict conditional independence, and explore how ER varies with n_1 and n_2. We generate the data the same way as in the main text, and keep $\alpha = 0, k = 2$, $d_1 = 50$ and $d_2 = 40$ We restrict the function class to linear model. Hence ψ is the linear model to predict X_2 from X_1 given the pretext dataset. We use Mean Squared Error (MSE) as the metric, since it is the empirical version of the ER. As shown in Figure Figure 4.3, ψ consistently outperforms X_1 in predicting Y using a linear model learnt from the given downstream dataset, and ER does scale linearly with $1/n_2$, as indicated by our analysis.

Computer Vision Task. For the context encoder part, we use all the recommended hyperparameter as in the provided source codes. For the downstream resnet18 regression, we perform grid search over the hyperparameters to achieve best performance. Specifically, we set the batch size to be 24, and traing the resnet18 for 50 epoches. One pass of training (loops over all the settings with different number of labeled data) is finished within 6 hours. All the experiments are performed on a desktop computer with Intel i7-8700K, 16GB RAM, and NVIDIA Geforce 1080. Training of the context encoder is finished within 12 hours. The

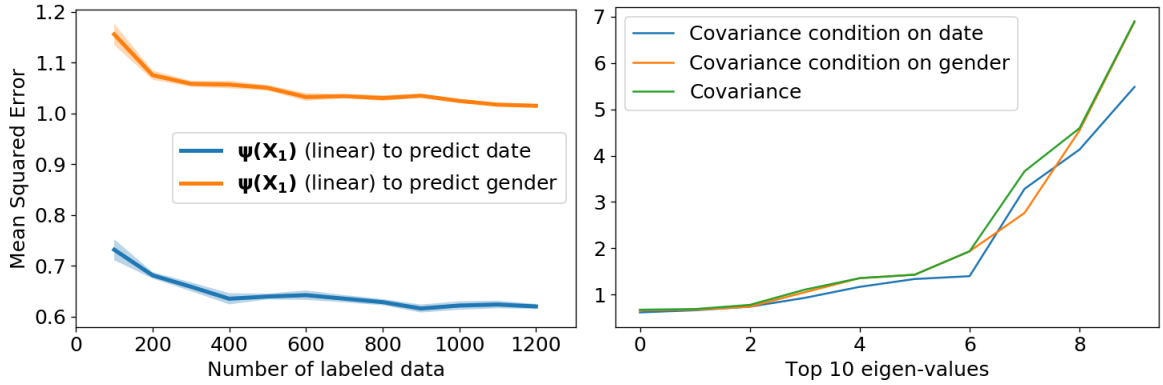


Figure 4.4: **Left**: Mean Squared Error comparison of predicting gender and predicting date. **Right**: the spectrum comparison of covariance condition on gender and condition on date.

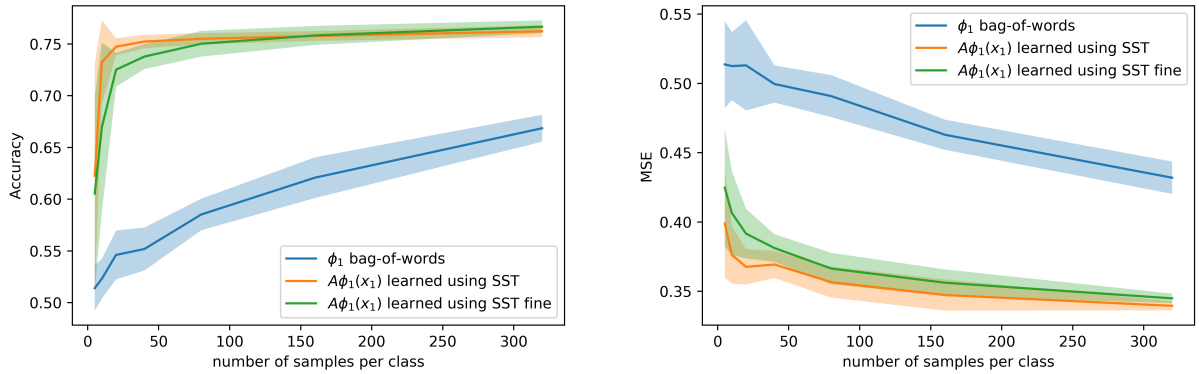


Figure 4.5: Performance on SST of baseline $\phi_1(\boldsymbol{x}_1)$, i.e. bag-of-words, and learned $\psi(\boldsymbol{x}_1)$ for the two settings. **Left:** Classification accuracy, **Right:** Regression MSE.

yearbook dataset is distributed under BSD license.

Following the same procedure, we try to predict the gender Y_G. We normalize the label (Y_G, Y_D) to unit variance, and confine ourself to linear function class. That is, instead of using a context encoder to impaint X_2 from X_1, we confine ψ to be a linear function. As shown on the left of Figure Figure 4.4, the MSE of predicting gender is higher than predicting dates. We find that $\|\boldsymbol{\Sigma}_{\boldsymbol{X}_1\boldsymbol{X}_1}^{-1/2}\boldsymbol{\Sigma}_{\boldsymbol{X}_1 X_2|Y_G}\|_F = 9.32$, while $\|\boldsymbol{\Sigma}_{\boldsymbol{X}_1\boldsymbol{X}_1}^{-1/2}\boldsymbol{\Sigma}_{\boldsymbol{X}_1 X_2|Y_D}\|_F = 8.15$. Moreover, as shown on the right of Figure Figure 4.4, conditioning on Y_D cancels out more spectrum than conditioning on Y_G. In this case, we conjecture that, unlike Y_D, Y_G does not capture much dependence between X_1 and X_2. And as a result, ϵ_{CI} is larger, and the downstream performance is worse, as we expected.

NLP Task. We look at the setting where both $\mathcal{X}_1$ and $\mathcal{X}_2$ are the set of sentences and perform experiments by enforcing CI with and without latent variables. The downstream task is sentiment classification with the Stanford Sentiment Treebank (SST) dataset [Socher et al., 2013], where inputs are movie reviews and the label set $\mathcal{Y}$ is $\{\pm 1\}$. We learn a linear representation $\psi(X_1) = \boldsymbol{B}\phi(X_1)$ in the SSL phase as defined in Section 4.4. Here we X_1, we pick $\phi(X_1)$ to be the bag-of-words representations of the movie review X_1, which has a vocabulary size of 13848 For X_2 we use a $d_2 = 300$ dimensional embedding of the sentence, that is the mean of word vectors (random Gaussians) for the words in the review X_2. For SSL data we consider 2 settings, (a) enforce CI with the labels $\mathcal{Y}$, (b) enforce CI with extra latent variables, for which we use fine-grained version of SST with label set $\bar{\mathcal{Y}} = \{1, 2, 3, 4, 5\}$[10].. In this setting, for every label $y \in \mathcal{Y}$ (or $\bar{y} \in \bar{\mathcal{Y}}$), we independently sample movie reviews X_1 and X_2 from the class y (or $\bar{y}$), thus simulating the CI (or approximate CI) condition. We test the learned ψ on SST binary task with linear regression and linear classification; results are presented in Figure Figure 4.5. We observe that in both settings ψ outperforms ϕ_1, especially in the small-sample-size regime. Exact CI is better than CI with latent variables, as suggested by theory.

The function ψ (or equivalently matrix $\boldsymbol{B} \in \mathbb{R}^{300\times 13848}$) is learnt by minimizing $\|X_2 - \boldsymbol{B}\phi(X_1)\|^2$ averaged over the SSL train data with an $\|\cdot\|_F^2$ penalty on the matrix $\boldsymbol{B}$. We use the scikit-learn RidgeRegressionCV[11] solver for this with regularizer parameters in the list $[0.001, 0.1, 10, 1000]$. Plotting Figure Figure 4.5 took less than an hour when using 8 Intel(R) Xeon(R) Silver 4214 CPUs on a cluster.

[10] Ratings $\{1, 2\}$ correspond to $y = -1$ and $\{4, 5\}$ correspond to $y = 1$

[11] `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeCV.html`

Part III

Language Modeling

Chapter 5

A Mathematical Exploration of Why Language Models Help Solve Downstream Tasks

This chapter focuses on the study of language modeling, based on previously published work [Saunshi et al., 2021]. Autoregressive language models, pretrained using large text corpora to do well on next word prediction, have been successful at solving many downstream tasks, even with zero-shot usage. However, there is little theoretical understanding of this success. This work initiates a mathematical study of this phenomenon for the downstream task of text classification by considering the following questions: (1) What is the intuitive connection between the pretraining task of next word prediction and text classification? (2) How can we mathematically formalize this connection and quantify the benefit of language modeling? For (1), we hypothesize, and verify empirically, that classification tasks of interest can be reformulated as sentence completion tasks, thus making language modeling a meaningful pretraining task. With a mathematical formalization of this hypothesis, we make progress towards (2) and show that language models that are ϵ-optimal in cross-entropy (log-perplexity) learn features that can *linearly solve* such classification tasks with $\mathcal{O}(\sqrt{\epsilon})$ error, thus demonstrating that doing well on language modeling can be beneficial for downstream tasks. We experimentally verify various assumptions and theoretical findings, and also use insights from the

analysis to design a new objective function that performs well on some classification tasks.

5.1 Introduction

The construction of increasingly powerful language models has revolutionized natural language processing (NLP). Using gigantic text corpora and a cross-entropy objective, language models are trained to predict a distribution over the *next word* to follow a given context (piece of text). Pretrained language models are useful for many downstream NLP tasks, either as initializations [Ramachandran et al., 2017, Howard and Ruder, 2018] or as a source of contextual word embeddings [McCann et al., 2017, Peters et al., 2018]. Recent models [Radford et al., 2019, Brown et al., 2020] have even bypassed the need for careful fine-tuning and have demonstrated strong performance on downstream tasks without fine-tuning. This work aims to understand this incredible success of language models.

Since next word prediction is a powerful test of language understanding, at an intuitive level it is believable that doing well on language modeling can help with many diverse NLP tasks. At the same time, it is quite intriguing how improvements in the test perplexity of language models translate to better downstream performance. Attempting to understand this phenomenon naturally raises the following questions: *(a) why should training on the next-word prediction task, with the cross-entropy objective, result in useful features for downstream tasks? (b) what role do inductive biases of the model architecture and training algorithms play in this empirical success?* Given the nascency of deep learning theory, it is very challenging to say anything mathematically precise about (b) for deep networks. Given these difficulties, this work focusses on the mathematical study of (a) by exploring if and how quantitative improvements on downstream NLP tasks can be *mathematically guaranteed* for language models that do well on the cross-entropy objective. As a first cut analysis, we restrict attention to *text classification tasks* and the striking observation that they can be solved fairly well with *linear classifiers* on top of fixed language models features, i.e. without finetuning (Table 5.1). Although we treat models as black boxes, just first-order optimality conditions of the cross-entropy objective reveal interesting properties of learned features, leading to an understanding of their success on classification tasks. Insights from the analysis help us construct a simple objective (Quad), that provably learns useful features for classification tasks, as also verified empirically. We summarize our contributions along with an overview of the chapter below.

In Section 5.2, we set up notation and formally describe language modeling and the ubiquitous low-dimensional

softmax parametrization, along with a description of the cross-entropy objective and properties of its optimal solutions. We then describe the observation, in Section 5.3.1, that text classification tasks of interest can be reformulated as sentence completion tasks. Amenability to such a reformulation is mathematically formalized (Section 5.3.2) as the classification task being a *natural task*: tasks that can be solved *linearly* using conditional distribution over words following an input text. Section 5.4 presents our main results, Theorems 5.4.1 and 5.4.3, that use the above formalization to mathematically quantify the utility of language model features on natural tasks: ϵ-optimal language model (in cross-entropy) will do $\mathcal{O}(\sqrt{\epsilon})$-well on such tasks. Theorem 5.4.3 shows a stronger result for low-dimensional softmax models by leveraging a new tool, *conditional mean features* (Definition 5.4.2), which we show (Section 5.6) to be effective in practice. The usefulness of the language model features themselves is demonstrated by arguing a weak linear relationship between them and conditional mean features. In Section 5.5.2, we present a new mathematically motivated objective (*Quad*) that has formal guarantees. Experiments in Section 5.6 verify the sentence completion reformulation idea and the good performance of conditional mean features on standard benchmarks.

5.1.1 Related work

Text embedding methods: Prior to language models, large text corpora like Wikipedia [Merity et al., 2016] were used to learn low-dimensional embeddings for words [Mikolov et al., 2013b,a, Pennington et al., 2014] and subsequently for sentences [Kiros et al., 2015, Arora et al., 2017, Pagliardini et al., 2018, Logeswaran and Lee, 2018] for downstream task usage. These methods were inspired by the distributional hypothesis [Firth, 1957, Harris, 1954], which posits that meaning of text is determined in part by the surrounding context. Recent methods like BERT [Devlin et al., 2019] and variants [Lan et al., 2020, Yang et al., 2019, Liu et al., 2019] learn models from auxiliary tasks, such as sentence completion, and are among the top performers on downstream tasks. In this work we consider autoregressive models and make a distinction from masked language models like BERT; Table 5.2 shows that language model and BERT features have comparable performances.

Language models for downstream tasks: We are interested in language models [Chen and Goodman, 1999], especially those that use neural networks to compute low-dimensional features for contexts and parametrize the next word distribution using softmax [Xu and Rudnicky, 2000, Bengio et al., 2003]. Language models have shown to be useful for downstream tasks as initializations [Ramachandran et al., 2017, Howard and Ruder, 2018] or as learned feature maps [Radford et al., 2017, McCann et al., 2017, Peters et al.,

2018]. The idea of phrasing classification tasks as sentence completion problems to use language models is motivated by recent works [Radford et al., 2019, Puri and Catanzaro, 2019, Schick and Schütze, 2021] that show that many downstream tasks can be solved by next word prediction for an appropriately conditioned language model. This idea also shares similarities with work that phrase a suite of downstream tasks as question-answering tasks [McCann et al., 2018] or text-to-text tasks [Raffel et al., 2019] and symbolic reasoning as fill-in-the-blank tasks [Talmor et al., 2020]. Our work exploits this prevalent idea of task rephrasing to theoretically analyze why language models succeed on downstream tasks.

Relevant theory: Since the success of early word embedding algorithms like word2vec [Mikolov et al., 2013a] and GloVe [Pennington et al., 2014], there have been attempts to understand them theoretically. Levy and Goldberg [2014] argue that word2vec algorithm implicitly factorizes the PMI matrix. Noise Contrastive Estimation (NCE) theory is used to understand word embeddings [Dyer, 2014] and to show parameter recovery for negative sampling based conditional models [Ma and Collins, 2018]. A latent variable model [Arora et al., 2016] is used to explain and unify various word embedding algorithms. Theoretical justification is provided for sentence embedding methods either by using a latent variable model [Arora et al., 2017] or through the lens of compressed sensing [Arora et al., 2018]. Also relevant is recent work on theory for contrastive learning [Arora et al., 2019, Tosh et al., 2021b,a, Wang and Isola, 2020] and reconstruction-based methods [Lee et al., 2021], which analyze the utility of self-supervised representations learned for downstream tasks. Our work is the first to analyze the efficacy of language model features on downstream tasks.

5.2 Language modeling and optimal solutions

We use $\mathcal{S}$ to denote the discrete set of all contexts, i.e. complete or partial sentences (prefixes), $\mathcal{W}$ to denote the vocabulary of words, with $V = |\mathcal{W}|$ being the vocabulary size. For a discrete set A, let Δ_A denote the set of distributions on A. We use $p, p_L \in \Delta_{\mathcal{S}}$ to denote probability distributions over $\mathcal{S}$, and $p_{\cdot|s}, p^*_{\cdot|s} \in \Delta_{\mathcal{W}}$ to denote conditional distributions, where $p_{\cdot|s}(w)$ is the predicted probability of word w following context s and $p^*_{\cdot|s}(w)$ denotes the true conditional probability. Boldface $\boldsymbol{p}_{\cdot|s}, \boldsymbol{p}^*_{\cdot|s} \in \mathbb{R}^V$ denote vectors of probabilities for $p_{\cdot|s}, p^*_{\cdot|s} \in \Delta_{\mathcal{W}}$. For $\boldsymbol{v} \in \mathbb{R}^V$, $\boldsymbol{v}(w)$ indexes the coordinate for $w \in \mathcal{W}$; $\boldsymbol{p}_{\cdot|s}(w)$ is the probability of w according to $p_{\cdot|s}$. We use $\phi_w \in \mathbb{R}^d$ to denote a d-dimensional embedding for word w; word embeddings are stacked into the columns $\Phi \in \mathbb{R}^{d \times V}$. We use $f : \mathcal{S} \to \mathbb{R}^d$ for a feature map from contexts to d-dimensional embeddings, e.g. $f(s)$ can be the output of a Transformer model for input context $s \in \mathcal{S}$. For embeddings $\{\theta_s\}_{s \in \mathcal{S}}$ with $\theta_s \in \mathbb{R}^D$ (any D), we use $\{\theta_s\}$ to denote $g : \mathcal{S} \to \mathbb{R}^D$ such that $g(s) = \theta_s$.

5.2.1 Language modeling using cross-entropy

Language model aims to learn the true distribution of a text corpus and a popular approach to do so is through next word prediction. Given a context (e.g., a sentence $s \in \mathcal{S}$), it predicts a distribution $p_{\cdot|s}$ over the word to follow, e.g. for the context "The food was ", the model could place high probabilities on words "delicious", "expensive", "bland", etc. We use p_L to denote the true distribution over the context set $\mathcal{S}$ in the language modeling corpus. A standard approach is to minimize the expected cross-entropy loss between the true distribution $p^*_{\cdot|s}$ and the model prediction $p_{\cdot|s}$. We define the cross-entropy loss for a language model with output vector of probabilities $\{\boldsymbol{p}_{\cdot|s}\}_{s\in\mathcal{S}}$ as

$$\ell_{\text{xent}}(\{\boldsymbol{p}_{\cdot|s}\}) = \mathop{\mathbb{E}}_{s\sim p_L} \mathop{\mathbb{E}}_{w\sim p^*_{\cdot|s}} \left[-\log(\boldsymbol{p}_{\cdot|s}(w))\right] = \mathop{\mathbb{E}}_{s\sim p_L} \left[\ell_{\text{xent},s}(\boldsymbol{p}_{\cdot|s})\right] \tag{5.1}$$

To understand what language models learn, we look at the optimal solution of the cross-entropy objective. While one cannot practically hope to learn the optimal solution due to optimization, statistical and expressivity limitations, the optimal solution at least tells us the best that language modeling can hope to do. A well-known property of cross-entropy objective is that its optimal solution is $\boldsymbol{p}^*_{\cdot|s}$, which can be proved by noting that $\ell_{\text{xent},s}(\boldsymbol{p}_{\cdot|s}) = D_{\text{KL}}(p^*_{\cdot|s}, p_{\cdot|s}) + C$.

Proposition 5.2.1 (Cross-entropy recovers $\boldsymbol{p}^*_{\cdot|s}$)**.** *The unique minimizer of $\ell_{xent}(\{\boldsymbol{p}_{\cdot|s}\})$ is $\boldsymbol{p}_{\cdot|s} = \boldsymbol{p}^*_{\cdot|s}$ for every $s \in$ support(p_L).*

5.2.2 Softmax parametrized language modeling

Unlike traditional language models like n-gram models, neural language models parametrize the conditional distribution $p_{\cdot|s}$ as a softmax computed using *low dimensional* embeddings. For an embedding $\theta \in \mathbb{R}^d$, the softmax distribution over $\mathcal{W}$ using word embeddings $\Phi \in \mathbb{R}^{d\times V}$ is $p_{\theta,\Phi}(w) = e^{\theta^\top \phi_w}/Z_\theta$, where $Z_\theta = \sum_{w'\in\mathcal{W}} e^{\theta^\top \phi_{w'}}$ is the partition function. While $p_{\theta,\Phi}$ depends on Φ, we will use p_θ instead whenever Φ is clear from context. Just like $\boldsymbol{p}^*_{\cdot|s}$, we can interpret $\boldsymbol{p}_\theta \in \mathbb{R}^V$ as a vector of probabilities for the distribution p_θ.

We now describe the abstraction for softmax models that is applicable to most neural models. A language model first embeds a context s into $f(s) \in \mathbb{R}^d$ using a feature map $f : \mathcal{S} \to \mathbb{R}^d$ that is parametrized by an architecture of choice (e.g. Transformer [Vaswani et al., 2017]). The output conditional distribution is set to be the softmax distribution induced by the context embedding $f(s)$ and word embeddings Φ, i.e. $p_{\cdot|s} = p_{f(s)}$.

The cross-entropy in its familiar form is presented below

$$\ell_{\text{xent}}(f, \Phi) = \mathop{\mathbb{E}}_{s\sim p_L} \mathop{\mathbb{E}}_{w\sim p^*_{\cdot|s}} \left[-\log(\boldsymbol{p}_{f(s)}(w))\right] = \mathop{\mathbb{E}}_{s\sim p_L} \left[\mathop{\mathbb{E}}_{w\sim p^*_{\cdot|s}} [-f(s)^\top \phi_w] + \log(Z_{f(s)})\right] \tag{5.2}$$

We rewrite it as $\ell_{\text{xent}}(f, \Phi) = \mathop{\mathbb{E}}_{s\sim p_L} [\ell_{\text{xent},s}(f(s), \Phi)]$, where $\ell_{\text{xent},s}(\theta, \Phi) = \ell_{\text{xent},s}(\boldsymbol{p}_{\theta,\Phi})$ is the cross-entropy loss for a context s that uses embedding θ. Analogous to Proposition 5.2.1, we would like to know the optimal d-dimensional feature map f^* and the induced conditional distribution $\boldsymbol{p}_{f^*(s)}$[1].

Proposition 5.2.2 (Softmax models recover $\boldsymbol{p}^*_{\cdot|s}$ on a subspace)**.** *Fix a fixed* Φ*, if* $f^* \in \arg\min_{f:\mathcal{S}\to\mathbb{R}^d} \ell_{xent}(f, \Phi)$ *exists, then* $\Phi\boldsymbol{p}_{f^*(s)} = \Phi\boldsymbol{p}^*_{\cdot|s}$ *for every* $s \in$ *support*(p_L)*.*

Unlike Proposition 5.2.1, $\boldsymbol{p}_{f^*(s)} \in \mathbb{R}^V$ is only guaranteed to be equal to $\boldsymbol{p}^*_{\cdot|s} \in \mathbb{R}^V$ on the d-dimensional subspace spanned by rows of $\Phi \in \mathbb{R}^{d\times V}$. We may not learn $\boldsymbol{p}^*_{\cdot|s}$ exactly when $d < V$, but this result at least guarantees learning $\boldsymbol{p}^*_{\cdot|s}$ on a *linear subspace* determined by word embeddings Φ. This forms the basis for our main results later and is proved by using the first-order optimality condition, i.e. $\nabla_\theta \ell_{\text{xent},s}(f^*(s)) = 0,\ \forall s \in \mathcal{S}$. The gradient of cross-entropy is $\nabla_\theta \ell_{\text{xent},s}(\theta) = -\Phi\boldsymbol{p}^*_{\cdot|s} + {}^{\nabla_\theta Z_\theta}/_{Z_\theta} = -\Phi\boldsymbol{p}^*_{\cdot|s} + \Phi\boldsymbol{p}_\theta$. Setting it to 0 completes the proof. We use the properties of optimal solutions to understand why language models help with classification tasks.

5.3 Using language models for classification tasks

Sections 5.2.1 and 5.2.2 suggest that language models aim to learn $\boldsymbol{p}^*_{\cdot|s}$, or a low-dimensional projection $\Phi\boldsymbol{p}^*_{\cdot|s}$. Thus to understand why language models help with downstream tasks, a natural starting point is to understand how access to $\boldsymbol{p}^*_{\cdot|s}$ can help with downstream tasks. In a thought experiment, we use oracle access to $\boldsymbol{p}^*_{\cdot|s}$ for any s and demonstrate that sentence classification task can be solved by reformulating it as a sentence completion problem and using $\boldsymbol{p}^*_{\cdot|s}$ to get completions to predict the label. This sentence completion reformulation is mathematically formalized as *natural tasks*.

5.3.1 Sentence completion reformulation

For exposition, we consider the sentence classification task of sentiment analysis, where the inputs are movie reviews (subset of $\mathcal{S}$) and labels belongs to $\{\pm 1\}$, denoting positive and negative reviews.

Classification task as sentence completion: Can we predict the label for a movie review s by using

[1] A finite minimizer may not always exist. This is handled in Section 5.4 that deals with ϵ-optimal solutions.

$\boldsymbol{p}^*_{\cdot|s}$? One way is to use $\boldsymbol{p}^*_{\cdot|s}$ to compare probabilities of ":)" and ":(" following a movie review and to predict sentiment based on which is higher. This seems like a reasonable strategy, since ":)" is likelier than ":(" to follow a positive movie review. One issue, however, is that $\boldsymbol{p}^*_{\cdot|s}$ will place much higher probability on words that start sentences, like "The", rather than discriminative words useful for the task. To allow a larger set of grammatically correct completions, we can append a prompt like "This movie is " at the end of all movie reviews and query probabilities of indicative adjectives like good, bad, interesting, boring etc. that are better indicators of sentiment. This approach of adding a prompt can also work for other classification tasks. For the AG news dataset [Zhang et al., 2015] containing news articles from 4 categories (world, science/tech., sports, business), a prompt like "This article is about " can help solve the task. The theoretical and practical relevance of prompts is discussed in Theorem 5.4.1, and Section 5.6 respectively. We note that the choice of prompts and completion words is less important than the underlying idea of sentence completion reformulation and its formalization.

Solving tasks using a linear function of $\boldsymbol{p}^*_{\cdot|s}$: The above process is actually a sub-case of using a linear classifier on top of $\boldsymbol{p}^*_{\cdot|s} \in \mathbb{R}^V$. For sentiment analysis, if $w_+ =$ ":)" and $w_- =$ ":(", then the sign of $\boldsymbol{p}^*_{\cdot|s}(w_+) - \boldsymbol{p}^*_{\cdot|s}(w_-)$ can predict the sentiment. This strategy can be expressed as $\boldsymbol{v}^\top \boldsymbol{p}^*_{\cdot|s}$, where the linear classifier $\boldsymbol{v} \in \mathbb{R}^V$ has $\boldsymbol{v}(w_+) = 1$, $\boldsymbol{v}(w_-) = -1$ and $\boldsymbol{v}(w') = 0$ for $w' \in \mathcal{W} \backslash \{w_+, w_-\}$. Similarly with the prompt, we can assign positive weights in $\boldsymbol{v}$ to adjectives like "good" and negative weights to adjectives like "boring". Strength of sentiment in different adjectives (e.g., "good" vs "amazing") can be captured through different weights. This equivalence between sentence completion reformulation and linear classifier on $\boldsymbol{p}^*_{\cdot|s}$ is further explored in Section 5.11.1. Other tasks can be similarly solved with a different set of words for each class. We verify experimentally that SST and AG news tasks can be solved by a linear function of probabilities of just a small subset of words in Section 5.6 and for many other classification tasks in Section 5.13.1, thus lending credibility to the sentence completion view.

5.3.2 Natural classification tasks

We now translate the above sentence completion reformulation into a reasonable mathematical characterization for classification tasks of interest. Firstly we formally define text classification tasks and the standard metric for performance of linear classification on fixed features. A binary classification task[2] $\mathcal{T}$ is characterized by a distribution $p_{\mathcal{T}}$ over $\mathcal{S} \times \{\pm 1\}$, where the input s is a piece of text from $\mathcal{S}$ and the label y is in $\{\pm 1\}$. Given a feature map $g : \mathcal{S} \to \mathbb{R}^D$ (arbitrary D), $\mathcal{T}$ is solved by fitting a linear classifier $\boldsymbol{v} \in \mathbb{R}^D$ on top of $g(s)$ and

[2]Extending to k-way tasks is straightforward.

the metric of classification loss is

$$\ell_{\mathcal{T}}(g, \boldsymbol{v}) = \mathbb{E}_{(s,y)\sim p_{\mathcal{T}}}\left[\ell(\boldsymbol{v}^\top g(s), y)\right]; \quad \ell_{\mathcal{T}}(g) = \inf_{\boldsymbol{v}\in\mathbb{R}^D} \ell_{\mathcal{T}}(g, \boldsymbol{v}) \tag{5.3}$$

where ℓ is a 1-Lipschitz surrogate to the 0-1 loss, like the hinge loss $\ell(\hat{y}, y) = (1 - y\hat{y})_+$ or the logistic loss $\ell(\hat{y}, y) = \log(1 + e^{-y\hat{y}})$. For given embeddings $\{\theta_s\}_{s\in\mathcal{S}}$, the classification loss is written as $\ell_{\mathcal{T}}(\{\theta_s\}, \boldsymbol{v}) = \mathbb{E}_{(s,y)\sim p_{\mathcal{T}}}[\ell(\boldsymbol{v}^\top \theta_s, y)]$.

We now formalize classification tasks amenable to sentence completion reformulation, from Section 5.3.1), as (τ, B)-natural tasks, i.e. tasks that achieve a small classification loss of τ by using a linear classifier with ℓ_∞-norm bounded[3] by B on top of features $\boldsymbol{p}^*_{\cdot|s} \in \mathbb{R}^V$.

Definition 5.3.1. *A classification task $\mathcal{T}$ is (τ, B)-natural if* $\min_{\boldsymbol{v}\in\mathbb{R}^V, \|\boldsymbol{v}\|_\infty \le B} \ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\}, \boldsymbol{v}) \le \tau$.

While we motivated this formalization of linear classification over $\boldsymbol{p}^*_{\cdot|s}$ in Section 5.3.1, we provide a mathematical justification in Section 5.11.1, along with interpretations for τ and B that relate them to the Bayes optimal predictor and probability mass of indicative words respectively. Low dimensional softmax models, however, only learn $\boldsymbol{p}^*_{\cdot|s}$ in the subspace of Φ, per Proposition 5.2.2. Thus we are also interested in subset of tasks that this subspace can solve.

Definition 5.3.2. *Task $\mathcal{T}$ is (τ, B)-natural w.r.t. $\Phi \in \mathbb{R}^{d\times V}$ if* $\min_{\boldsymbol{v}\in \textit{row-span}(\Phi), \|\boldsymbol{v}\|_\infty \le B} \ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\}, \boldsymbol{v}) \le \tau$.

Note that every (τ, B)-natural task w.r.t. Φ is trivially (τ, B)-natural, though the converse may not hold. However it can be argued that if Φ has some "nice properties", then (τ, B)-natural tasks of interest will roughly also be (τ, B)-natural w.r.t. Φ. Capturing the synonym structure of words can be such a nice property, as discussed in Section 5.11.2. A better understanding of these properties of word embeddings Φ can potentially enable better performance of language models on downstream tasks. In fact, Section 5.5.2 describes a carefully designed objective that can learn word embeddings with desirable properties like synonyms having similar embeddings. In the subsequent sections, we use the above formalization to show guarantees for language models on natural tasks.

[3] ℓ_∞ makes sense since $\|\boldsymbol{p}^*_{\cdot|s}\|_1 = 1$ & $\|\cdot\|_\infty$ is dual norm of $\|\cdot\|_1$.

5.4 Guarantees for language models on natural tasks

We now show guarantees for features from language models on natural tasks in two cases: 1) for an arbitrary language model $\{p_{\cdot|s}\}$ where we use V-dimensional features $\boldsymbol{p}_{\cdot|s} \in \mathbb{R}^V$ for downstream tasks and 2) for softmax language model (f, Φ) where we use new d-dimensional features $\Phi \boldsymbol{p}_{f(s)} \in \mathbb{R}^d$. Since we cannot practically hope to learn the optimal solutions described in Propositions 5.2.1 and 5.2.2, we only assume that the language models are ϵ-optimal in cross-entropy. We first define ℓ^*_{xent} to be the minimum achievable cross-entropy and $\ell^*_{\text{xent}}(\Phi)$ to be the minimum achievable cross-entropy by a d-dimensional softmax language model using Φ; clearly $\ell^*_{\text{xent}} \le \ell^*_{\text{xent}}(\Phi)$.

$$\ell^*_{\text{xent}} = \ell_{\text{xent}}(\{\boldsymbol{p}^*_{\cdot|s}\}), \quad \ell^*_{\text{xent}}(\Phi) = \mathop{\mathbb{E}}_{s \sim p_L}\left[\inf_{\theta \in \mathbb{R}^d} \ell_{\text{xent},s}(\theta, \Phi)\right] \tag{5.4}$$

We first present the results for arbitrary language models with a proof sketch that describes the main ideas, following which we present our main results for softmax language models.

5.4.1 Arbitary language models

We show guarantees for a language model that is ϵ-optimal, i.e. $\ell_{\text{xent}}(\{\boldsymbol{p}_{\cdot|s}\}) - \ell^*_{\text{xent}} \le \epsilon$, on (τ, B)-natural tasks. An important consideration is that the language model distribution p_L of contexts is often a diverse superset of the downstream distribution $p_{\mathcal{T}}$ (defined in Section 5.2.2) over sentences, thus requiring us to show how guarantees of $\boldsymbol{p}_{\cdot|s} \approx \boldsymbol{p}^*_{\cdot|s}$ *on average* over the distribution $s \sim p_L$ transfer to guarantees on a subset $p_{\mathcal{T}}$. In the worst case, all of the ϵ error in cross-entropy by $\{\boldsymbol{p}_{\cdot|s}\}$ is incurred on sentences from the subset $p_{\mathcal{T}}$, leading to pessimistic bounds[4]. In practice, however, the errors might be more evenly distributed across p_L, thus bypassing this worst case bound. As a first step, we present the worst case bound here; stronger guarantees are in Section 5.5.1. The worst-case coefficient $\gamma(p_{\mathcal{T}})$, defined below, captures that $p_{\mathcal{T}}$ is a $\gamma(p_{\mathcal{T}})$-fraction of p_L.

$$\gamma(p_{\mathcal{T}}) = \sup\{\gamma \in (0, 1] : p_L(s) \ge \gamma p_{\mathcal{T}}(s) \ \forall s \in \mathcal{S}\} \tag{5.5}$$

We now present our results that applies to any language model, regardless of the parametrization (e.g., n-gram models, softmax models). The result suggests that small test cross-entropy (hence test perplexity) is desirable to guarantee good classification performance, thus formalizing the intuition that better language

[4]For instance if $p_{\mathcal{T}}$ is 0.001 fraction of p_L, $\{p_{\cdot|s}\}$ could have 1000ϵ error on $p_{\mathcal{T}}$ and 0 error on rest of p_L.

models will be more useful for downstream tasks.

Theorem 5.4.1. *Let $\{\boldsymbol{p}_{\cdot|s}\}$ be a language model that is ϵ-optimal, i.e. $\ell_{xent}(\{\boldsymbol{p}_{\cdot|s}\}) - \ell^*_{xent} \leq \epsilon$, for some $\epsilon > 0$. For a classification task $\mathcal{T}$ that is (τ, B)-natural, we have*

$$\ell_{\mathcal{T}}\left(\{\boldsymbol{p}_{\cdot|s}\}\right) \leq \tau + \sqrt{2B^2\epsilon\left(\gamma(p_{\mathcal{T}})\right)^{-1}}$$

This upper bounds classification loss on task $\mathcal{T}$ for V-dimensional features $\{\boldsymbol{p}_{\cdot|s}\}$ from an ϵ-optimal language model. We discuss factors that lead to small upper bound and corresponding intuitions.

• ϵ is small: learned language model has smaller cross-entropy (log-perplexity)

• τ is small: task can be solved well through a sentence completion reformulation with a set of indicative words as completions, as in Section 5.3.1, and has small Bayes error (cf. Section 5.11.1)

• B is small: set of indicative words has high probability mass in $\boldsymbol{p}^*_{\cdot|s}$ (cf. Section 5.11.1). This could potentially explain the superior performance when prompts are added (Section 5.6).

• $\gamma(p_{\mathcal{T}})$ is large: $p_{\mathcal{T}}$ is closer to p_L; note that $\gamma(p_{\mathcal{T}}) \leq 1$ with equality if and only if $p_{\mathcal{T}} = p_L$

Thus the bound captures meaningful intuitions about good performance of language models on downstream tasks. We provide a detailed proof sketch in Section 5.12.1 and a strengthened version of this (Theorem 5.9.2) is presented in Section 5.12.6. Proving this result requires connecting the classification loss with language modeling cross-entropy loss and dealing with distribution mismatch; we present a rough outline to do so below. Since $\mathcal{T}$ is (τ, B)-natural, let $\boldsymbol{v}^*$ be the classifier with $\|\boldsymbol{v}^*\|_\infty \leq B$ and $\ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\}, \boldsymbol{v}^*) \leq \tau$. The result follows from the following 3 inequalities:

$$\ell_{\mathcal{T}}\left(\{\boldsymbol{p}_{\cdot|s}\}, \boldsymbol{v}^*\right) - \ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\}, \boldsymbol{v}^*) \leq \sqrt{\mathop{\mathbb{E}}_{s\sim p_{\mathcal{T}}}[(\boldsymbol{v}^{*\top}(\boldsymbol{p}_{\cdot|s} - \boldsymbol{p}^*_{\cdot|s}))^2]} \qquad \text{... Lipschitzness + Jensen's}$$

$$\mathop{\mathbb{E}}_{s\sim p_{\mathcal{T}}}[(\boldsymbol{v}^{*\top}(\boldsymbol{p}_{\cdot|s} - \boldsymbol{p}^*_{\cdot|s}))^2] \leq \gamma(p_{\mathcal{T}})^{-1} \mathop{\mathbb{E}}_{s\sim p_L}[(\boldsymbol{v}^{*\top}(\boldsymbol{p}_{\cdot|s} - \boldsymbol{p}^*_{\cdot|s}))^2] \qquad \text{... Transfer } p_{\mathcal{T}} \text{ to } p_L$$

$$\forall \boldsymbol{v} \in \mathbb{R}^V,\ (\boldsymbol{v}^\top(\boldsymbol{p}_{\cdot|s} - \boldsymbol{p}^*_{\cdot|s}))^2 \leq 2\|\boldsymbol{v}\|^2_\infty(\ell_{\text{xent},s}(\boldsymbol{p}_{\cdot|s}) - \ell_{\text{xent},s}(\boldsymbol{p}^*_{\cdot|s})) \qquad \text{... Pinsker's inequality}$$

The first and third inequalities (Lemma 5.12.8 and Lemma 5.12.3) connect the classification loss to the cross-entropy loss in language modeling, while the second inequality deals with distribution mismatch between p_L and $p_{\mathcal{T}}$. We now present a stronger result for softmax models.

5.4.2 Softmax language model with conditional mean features

We now consider a softmax language model with feature map f that satisfies $\ell_{\text{xent}}(f, \Phi) - \ell^*_{\text{xent}}(\Phi) \leq \epsilon$; suboptimality is measured w.r.t. the best d-dimensional model, unlike Theorem 5.4.1,. Note that Theorem 5.4.1 can be invoked here to give a bound of $\ell_{\mathcal{T}}(\{\boldsymbol{p}_{f(s)}\}) \leq \tau + \mathcal{O}(B\sqrt{\epsilon + \epsilon^*_\Phi})$ on (τ, B)-natural tasks, where $\epsilon^*_\Phi = \ell^*_{\text{xent}}(\Phi) - \ell^*_{\text{xent}}$ is the suboptimality of the best d-dimensional model. The fixed error of $\mathcal{O}(B\sqrt{\epsilon^*_\Phi})$ (even when $\epsilon = 0$), however, is undesirable. We improve on this by proving a stronger result specifically for softmax models. Inspired by Proposition 5.2.2, our guarantees are for features $\Phi \boldsymbol{p}_{f(s)} \in \mathbb{R}^d$ called conditional mean features.

Definition 5.4.2 (Conditional Mean Features)**.** *For a feature map $f : \mathcal{S} \to \mathbb{R}^d$ and $\Phi \in \mathbb{R}^{d \times V}$, we define conditional mean features $\Phi p_f : \mathcal{S} \to \mathbb{R}^d$, where $\Phi p_f(s) = \Phi \boldsymbol{p}_{f(s)}$, where $\boldsymbol{p}_{f(s)} \in \mathbb{R}^V$.*

We now present the result for softmax language models that has similar implications as Theorem 5.4.1, but with above-mentioned subtle differences.

Theorem 5.4.3. *For a fixed Φ, let f be features from an ϵ-optimal d-dimensional softmax language model, i.e. $\ell_{xent}(f, \Phi) - \ell^*_{xent}(\Phi) \leq \epsilon$. For a classification task $\mathcal{T}$ that is (τ, B)-natural w.r.t. Φ,*

$$\ell_{\mathcal{T}}(\Phi p_f) \leq \tau + \sqrt{2B^2 \epsilon \left(\gamma(p_{\mathcal{T}})\right)^{-1}}$$

This result guarantees good performance of conditional mean features Φp_f on some natural tasks, thereby suggesting a novel way to extract features for downstream tasks. We empirically verify the good performance of $\Phi p_f(s)$ on classifications tasks (Section 5.6) and also find a $\mathcal{O}(\sqrt{\epsilon})$-like behavior (Section 5.13.5). The proof (Section 5.12.3) is similar to that of Theorem 5.4.1, the main difference being the use of the following inequality, proved using a *softmax variant of Pinsker's inequality* (Lemma 5.12.4).

$$\forall \boldsymbol{v} \in \text{row-span}(\Phi),\ (\boldsymbol{v}^\top(\boldsymbol{p}_{f(s)} - \boldsymbol{p}^*_{\cdot|s}))^2 \leq 2\|\boldsymbol{v}\|^2_\infty (\ell_{\text{xent},s}(\boldsymbol{p}_{f(s)}) - \inf_{f^*(s) \in \mathbb{R}^d} \ell_{\text{xent},s}(\boldsymbol{p}_{f^*(s)}))$$

The more general result (Theorem 5.5.1) replaces $\gamma(p_{\mathcal{T}})$ with a more refined coefficient (Section 5.5.1). While guarantees are only for natural tasks w.r.t. Φ, Section 5.11.2 discusses why this might be enough for *tasks of interest* if word embeddings Φ satisfy *nice properties*.

5.4.3 $\Phi p_f(s)$ is a linear function of $f(s)$

Theorem 5.4.3 shows that Φp_f is useful for linear classification. However, using feature map f directly is more standard and performs better in practice (Section 5.6). Here we argue that there is a linear relation between f and Φp_f if word embeddings Φ satisfy a certain Gaussian-like property, which we show implies that tasks solvable linearly with Φp_f are also solvable linearly using f.

Assumption 5.4.4. *There exists a symmetric positive semidefinite matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$, a vector $\boldsymbol{b} \in \mathbb{R}^d$ and a constant $c \in \mathbb{R}$ such that $\log(Z_\theta) = \frac{1}{2}\theta^\top \boldsymbol{A}\theta + \theta^\top \boldsymbol{b} + c$ for any $\theta \in \mathbb{R}^d$.*

If word embeddings were distributed as Gaussians, i.e. V columns of Φ are sampled from $\mathcal{N}(\mu, \Sigma)$ independently, it is not hard to show (Lemma 5.12.1) that $\log(Z_\theta) \approx \frac{1}{2}\theta^\top \Sigma\theta + \theta^\top \mu + \log(V)$. While some papers [Arora et al., 2016, Mu and Viswanath, 2018] have noted that word embeddings are fairly random-like in the bulk to argue that the log partition function is constant for $\|\theta\|_2 = 1$, our quadratic assumption is a bit stronger. However, empirically we find the fit to be very good, as evident in Figure 5.1. Under the above assumption, we can show a linear relation between f and Φp_f.

Lemma 5.4.5. *Under Assumption 5.4.4, feature map f satisfies $\Phi p_f(s) = \boldsymbol{A} f(s) + \boldsymbol{b}, \forall s \in \mathcal{S}$.*

Corollary 5.4.6. *Under same setting as Lemma 5.4.5 and Theorem 5.4.3, $\ell_{\mathcal{T}}(f) \leq \tau + \mathcal{O}(B\sqrt{\epsilon})$.*

This shows that f itself is good for natural classification tasks. However, in practice, the linearity between f and Φp_f only weakly holds on features from pretrained GPT-2 [Radford et al., 2018]. The fractional residual norm of the best linear fit, i.e. $r = \frac{\mathop{\mathbb{E}}_{s\sim p} \|\Phi p_f(s) - \boldsymbol{A} f(s) - \boldsymbol{b}\|^2}{\mathop{\mathbb{E}}_{s\sim p} \|\Phi p_f(s)\|^2}$, measured for different distributions ($r = 0$ is perfect fit) are 0.28 for SST, 0.39 for AG News, and 0.18 for IMDb contexts. This non-trivial linear relationship, although surprising, might not completely explain the success of f, which usually performs better than Φp_f; we leave exploring this to future work.

5.5 Extensions

5.5.1 Better handling of distributional shift

The bounds in the previous section use the coefficient $\gamma(p_{\mathcal{T}})$ to transfer guarantees from p_L to $p_{\mathcal{T}}$ and we define a more refined notion of transferability here. The coefficient $\gamma(p_{\mathcal{T}})$ is independent of the learned model and assumes a worst case distribution of errors. For the refined coefficient, we first define the error made in

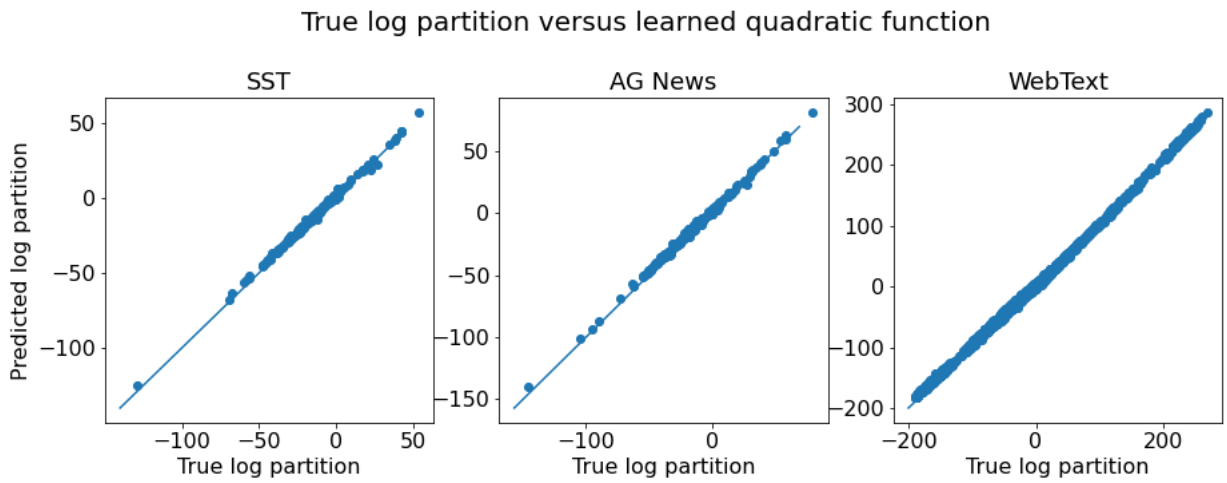


Figure 5.1: Learned quadratic function v/s log partition function on various datasets for features computed from pre-trained GPT-2 to verify Assumption 5.4.4. We also plot the $y = x$ line for reference.

predicted probabilities by a softmax language model f as $\Delta_{\{\boldsymbol{p}_{f(s)}\}}(s) = \boldsymbol{p}_{f(s)} - \boldsymbol{p}^*_{\cdot|s}$. For any distribution $p \in \Delta_S$, we define uncentered covariance of a function $g : \mathcal{S} \to \mathbb{R}^D$ as $\Sigma_p(g) = \mathbb{E}_{s\sim p}\left[g(s)g(s)^\top\right]$. The refined transferability coefficient is then defined as

$$\gamma(p; \Phi p_f) \coloneqq \left(\left\|\Sigma_{p_L}(\Phi\Delta_{\{\boldsymbol{p}_{f(s)}\}})^{-\frac{1}{2}}\Sigma_p(\Phi\Delta_{\{\boldsymbol{p}_{f(s)}\}})\Sigma_{p_L}(\Phi\Delta_{\{\boldsymbol{p}_{f(s)}\}})^{-\frac{1}{2}}\right\|_2\right)^{-1}$$

We state the refined result for softmax language models; detailed results are deferred to Section 5.9.

Theorem 5.5.1 (Simplified)**.** *In the same setting as Theorem 5.4.3,* $\ell_{\mathcal{T}}(\Phi p_f) \le \tau + \sqrt{\frac{2B^2\epsilon}{\gamma(p_{\mathcal{T}};\Phi p_f)}}$

It is easy show that $\gamma(p_{\mathcal{T}}; \Phi p_f) \ge \gamma(p_{\mathcal{T}})$, so this is indeed a stronger bound. The coefficient $\gamma(p_{\mathcal{T}}; \Phi p_f)$ measures how average error on f on p_L can propagate to $p_{\mathcal{T}}$. This can potentially be much smaller than $\gamma(p_{\mathcal{T}})$ due to some inductive biases of f. For instance, if errors made by the model are random-like, i.e. $\Delta_{\{\boldsymbol{p}_{f(s)}\}}(s) \sim \rho$, *independently of* s, then $\Sigma_{p_L}(\Phi\Delta_{\{\boldsymbol{p}_{f(s)}\}}) \approx \Sigma_p(\Phi\Delta_{\{\boldsymbol{p}_{f(s)}\}}) \approx \mathbb{E}_{\eta\sim\rho}[\eta\eta^\top]$, making $\gamma(p; \Phi p_f) \approx 1$. Independence prevents accumulation of language modeling error on contexts from $p_{\mathcal{T}}$, bypassing the worst case transfer of $\gamma(p_{\mathcal{T}})$.

5.5.2 Quad: A new objective function

In Definition 5.3.2 we discuss how low dimensional softmax language models learn a linear projection of $\boldsymbol{p}^*_{\cdot|s}$, only solving tasks that lie in the row span of word embeddings Φ. Although Φ defines tasks that language model features can solve, the standard cross-entropy objective does not lend a simple closed form expression

for optimal Φ. This motivates the construction of our Quad objective, that has two nice properties: (1) the optimal feature map f^* is a linear function of $\boldsymbol{p}^*_{\cdot|s}$ and thus can solve some natural tasks, and (2) the optimal Φ^* has an intuitively meaningful closed-form solution.

$$\ell_{quad}(f, \Phi) = \mathop{\mathbb{E}}_{s \sim p_L} \left[\mathop{\mathbb{E}}_{w \sim p^*_{\cdot|s}} [-f(s)^\top \phi_w] + \frac{1}{2} \|\Phi^\top f(s)\|^2 \right] \tag{5.6}$$

The Quad objective is very similar to the cross-entropy objective from Equation (5.2), with the log partition function replaced by a quadratic function, inspired in part by Assumption 5.4.4. We can derive the optimal solution Φ^* that depends on the eigen-decomposition of a *substitutability matrix*.

Definition 5.5.2. *The substitutability matrix is defined to be* $\Omega^* \coloneqq \mathop{\mathbb{E}}_{s \sim p_L} \left[\boldsymbol{p}^*_{\cdot|s} \, \boldsymbol{p}^{*\top}_{\cdot|s} \right] \in \mathbb{R}^{V \times V}$. *If* $\Omega^* = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^\top$ *is the eigendecomposition, then* $\boldsymbol{U}_d \in \mathbb{R}^{V \times d}$ *is matrix of top* d *eigenvectors of* Ω^*.

The matrix Ω^* captures substitutability between pairs of words. Words w and w' are substitutable if they have identical conditional probabilities for every context $s \in \mathcal{S}$ and thus can replace occurrences of each other while still providing meaningful completions. By definition, these words satisfy $\Omega^*[w] = \Omega^*[w']$. Such pairs of words were called "free variants" in the work on distributional semantics [Harris, 1954], and capture the notion of synonyms; more in Section 5.11.2.

Theorem 5.5.3. *Let* $f^*, \Phi^* = \arg\min_{f,\Phi} \ell_{quad}(f, \Phi)$. *Then* $\Phi^* = \boldsymbol{B}\boldsymbol{U}_d^\top$, *for full rank* $\boldsymbol{B} \in \mathbb{R}^{d \times d}$. *Also, for a classification task* $\mathcal{T}$ *that is* (τ, B)*-natural w.r.t.* Φ^*, *we have* $\ell_{\mathcal{T}}(f^*) \leq \tau$.

Thus f^* excels on natural tasks w.r.t. Φ^*, which in turn, is the best d-dimensional projection of Ω^*. Thus words $w, w' \in \mathcal{W}$ that are synonyms (hence substitutable) will satisfy $\phi^*_w = \phi^*_{w'}$, fulfilling the desired property for word embeddings discussed in Definition 5.3.2.

We train using the Quad objective and compare its performance to a similarly trained GPT-2 language model. The results in Table 5.3 suggest that Quad performs comparably to Φp_f from the cross-entropy objective, which fits our theory since both are linear functions of $\boldsymbol{p}^*_{\cdot|s}$. Section 5.13.3 has more details and experiments. The goal of testing Quad is to demonstrate that theoretical insights can aid the design of provably effective algorithms. Refer to Section 5.10 for more details on Quad.

Table 5.1: Accuracy (%) on k-way *linear classification* using fixed GPT-2 features. Good performance of features $f(s)$, conditional mean features $\Phi p_f(s)$ and meaningful subset of ≤ 30 (and $\leq 2k$) coordinates of $\boldsymbol{p}_{f(s)}$ verify the sentence completion reformulation and main results. The numbers right below the features denote dimensionality of the features. An asterisk indicates that we added a task-specific prompt. Other baselines are fine-tuning (FT, Section 5.13.2) and random projection of $\boldsymbol{p}_{f(s)}$ (rand. proj.). Sentence version of SST (train/test: 6.9K/1.8K) is used.

Task	k	$f(s)$ 768	$\Phi p_f(s)$ 768	$\boldsymbol{p}_{f(s)}$ (subset) ≤ 30	$\boldsymbol{p}_{f(s)}$ (class words) $\leq 2k$	$\boldsymbol{p}_{f(s)}$ (rand. proj.) 768	FT
SST	2	87.5	83.3	82.6	78.7	67.5	91.4
SST*	2	89.4	87.3	85.4	79.1	76.4	92.3
SST fine	5	49.2	43.5	44.0	39.2	23.1	50.2
SST fine*	5	49.4	48.6	47.6	40.3	28.8	53.5
AG	4	90.7	84.6	83.8	75.4	58.5	94.5
AG*	4	91.1	88.2	86.1	75.1	63.7	94.4

5.6 Experiments

We use experiments to verify (1) linear classification on fixed language model features does comparably to fine-tuning the features, (2) sentence completion reformulation (Section 5.3.1), i.e. tasks can be solved using probabilities for indicative words, (3) conditional mean features are effective.

Tasks using linear function of $\boldsymbol{p}^*_{\cdot|s}$: We validate our claims from Section 5.3 that classification tasks can be solved by linear functions of $\boldsymbol{p}^*_{\cdot|s}$. Since $\boldsymbol{p}^*_{\cdot|s}$ is never available, we instead use the output features $f(s)$ and probabilities $\boldsymbol{p}_{\cdot|s} \coloneqq \boldsymbol{p}_{f(s)}$ from a small pretrained GPT-2 model [Radford et al., 2019]. Table 5.1 demonstrates that on binary and fine-grained Stanford Sentiment Treebank (SST) [Socher et al., 2013] and AG News [Zhang et al., 2015] tasks, probabilities $\boldsymbol{p}_{f(s)}$ of just 30 or so task-relevant tokens (see Section 5.13.1) can solve the tasks. Even just one/two token per class ("class words") yields non-trivial performance. Furthermore, we validate the sentence completion reformulation in Section 5.3.1 by using the probabilities $\boldsymbol{p}_{f(s)}$ after adding a task specific prompt and consistently observing improved performance, including for fine-tuning (FT) with small datasets.

Φp_f and f are good features: We first note that linear classification over fixed features $f(s)$ from the pretrained model performs comparably to the FT baseline. We further validate Theorem 5.4.3 by verifying that the conditional mean features $\Phi p_f(s)$ also linearly solve downstream tasks fairly well. This performance is comparable to, but always worse than $f(s)$, as seen in columns 3 and 4 of Table Table 5.1. We again find that adding a prompt improves performance. Note that a random projection of $\boldsymbol{p}_{f(s)}$ to same dimensions as $\Phi p_f(s)$ has very poor performance. Section 5.12.5 has results for a wider range of classification tasks.

Evidence for Assumption 5.4.4 is provided by learning a quadratic function to fit the log partition function of features from pretrained GPT-2 model (see Section 5.13.4). Figure 5.1 demonstrates that the fit holds for its training and unseen data (e.g., WebText [Radford et al., 2019]).

5.7 Conclusions and future work

We provide intuitive and mathematical explanations for the success of language model features on classification tasks by reformulating them as sentence completion problems. This reformulation is formalized as *natural tasks*: those that can be solved linearly using the conditional probability distribution $\boldsymbol{p}^*_{\cdot|s}$. Insights from our analysis help design the Quad objective that provably learns good features for these natural tasks. We hope our analysis will inspire other mathematical insights into language models. While Section 5.4.3 argues linearity between conditional mean features Φp_f and f, it is insufficient to explain the observed superiority of f over Φp_f. We leave exploring this limitation of our analysis to future work. Guarantees for softmax models are for natural tasks w.r.t. Φ, thus knowing the optimal d-dimensional word embeddings Φ^* for $\ell_{\text{xent}}(f, \Phi)$ is also important. Other meaningful directions include providing guarantees for other successful models like BERT [Devlin et al., 2019] and more diverse downstream tasks. Although we would like to show stronger guarantees by exploiting model and algorithmic inductive biases, as well as study the setting of fine-tuning language model features, lack of a good theory of deep learning is the current bottleneck.

Acknowledgments: Sanjeev Arora, Sadhika Malladi and Nikunj Saunshi are supported by NSF, ONR, Simons Foundation, Amazon Research, DARPA and SRC.

5.8 Overview

Section 5.9 is a more detailed version of Section 5.5.1 and Section 5.10 is a detailed version of Section 5.5.2. Section 5.11.1 has a discussion about why *natural tasks* are a reasonable formalization for the sentence completion reformulation and also interpretations for τ and B in the definition of natural tasks. Section 5.11.2 discusses desirable properties of word embeddings Φ like capturing synonym structure in words. Section 5.12 contains proofs for all results, including proof sketches for the main results in Section 5.12.1. Lemma 5.12.4 is the softmax variant of Pinsker's inequality that we prove and use for our main results.

Section 5.13 contains many more experimental findings that consolidate many of our theoretical results. Section 5.13.1 provides the information about subsets of words used for results in Table 5.1 and also additional experiments to test the performance of pretrained language model embeddings f on more downstream tasks and also verifying that conditional mean embeddings Φp_f do well on these tasks. In Section 5.13.3, we present additional results for Quad objective trained on a larger corpus and tested on SST. Section 5.13.4 provides additional details on how $\boldsymbol{A}$, $\boldsymbol{b}$ and c from Assumption 5.4.4 are learned and also further verification of the assumption on more datasets. Finally, Section 5.13.5 experimentally verifies the $\mathcal{O}(\sqrt{\epsilon})$ dependence from Theorem 5.4.3.

5.9 More on better handling of distributional shift

While the bounds above used $\gamma(p_{\mathcal{T}})$ to transfer from the distribution p_L to $p_{\mathcal{T}}$, we define a more refined notion of transferability here. While $\gamma(p_{\mathcal{T}})$ only depends on p_L and $p_{\mathcal{T}}$, the more refined notions depend also on the learned language model, thus potentially exploiting some inductive biases. We first define the notion of error made in the predicted probabilities by any predictor $\boldsymbol{p}_{\cdot|s}$ as $\Delta_{\{\boldsymbol{p}_{\cdot|s}\}}(s) = \boldsymbol{p}_{\cdot|s} - \boldsymbol{p}^*_{\cdot|s}$. Thus for any softmax language model f we have $\Delta_{\{\boldsymbol{p}_{f(s)}\}}(s) = \boldsymbol{p}_{f(s)} - \boldsymbol{p}^*_{\cdot|s}$. For any distribution $p \in \Delta_S$, we define the covariance[5] of a function $g : \mathcal{S} \to \mathbb{R}^D$ as $\Sigma_p(g) = \mathop{\mathbb{E}}_{s\sim p}\left[g(s)g(s)^\top\right]$. We define 3 coefficients for the results to follow

Definition 5.9.1. *For any distribution $p \in \Delta_S$, we define the following*

$$\gamma(p; \{\boldsymbol{p}_{\cdot|s}\}) \coloneqq \left(\left\|\Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})^{-\frac{1}{2}}\Sigma_p(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})^{-\frac{1}{2}}\right\|_2\right)^{-1} \tag{5.7}$$

$$\gamma_\Phi(p; \{\boldsymbol{p}_{\cdot|s}\}) \coloneqq \left(\left\|\Sigma_{p_L}(\Phi\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})^{-\frac{1}{2}}\Sigma_p(\Phi\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\Sigma_{p_L}(\Phi\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})^{-\frac{1}{2}}\right\|_2\right)^{-1} \tag{5.8}$$

[5]This is not exactly the covariance since the mean is not subtracted, all results hold even for the usual covariance.

$$\gamma(p; \Phi p_f) \coloneqq \gamma_\Phi(p; \{\boldsymbol{p}_{f(s)}\}) \tag{5.9}$$

We notice that $\Sigma_p(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}}) = \mathop{\mathbb{E}}_{s\sim p}\left[(\boldsymbol{p}_{\cdot|s} - \boldsymbol{p}^*_{\cdot|s})(\boldsymbol{p}_{\cdot|s} - \boldsymbol{p}^*_{\cdot|s})^\top\right]$, $\Sigma_p(\Phi\Delta_{\{\boldsymbol{p}_{\cdot|s}\}}) = \Phi\Sigma_p(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\Phi^\top$. We are now ready to state the most general results.

Theorem 5.9.2 (Strengthened Theorem 5.4.1)**.** *Let $\{\boldsymbol{p}_{\cdot|s}\}$ be a language model that is ϵ-optimal, i.e. $\ell_{xent}(\{\boldsymbol{p}_{\cdot|s}\}) - \ell^*_{xent} \leq \epsilon$ for some $\epsilon > 0$. For a classification task $\mathcal{T}$ that is (τ, B)-natural, we have*

$$\ell_{\mathcal{T}}\left(\{\boldsymbol{p}_{\cdot|s}\}\right) \leq \tau + \sqrt{\frac{2B^2\epsilon}{\gamma(p_{\mathcal{T}}; \{\boldsymbol{p}_{\cdot|s}\})}}$$

For a classification task $\mathcal{T}$ that is (τ, B)-natural w.r.t. Φ, we have

$$\ell_{\mathcal{T}}\left(\{\boldsymbol{p}_{\cdot|s}\}\right) \leq \ell_{\mathcal{T}}\left(\{\Phi\boldsymbol{p}_{\cdot|s}\}\right) \leq \tau + \sqrt{\frac{2B^2\epsilon}{\gamma_\Phi(p_{\mathcal{T}}; \{\boldsymbol{p}_{\cdot|s}\})}}$$

Theorem 5.5.1 (Strengthened Theorem 5.4.3)**.** *For a fixed Φ, let f be features from an ϵ-optimal d-dimensional softmax language model, i.e. $\ell_{xent}(f, \Phi) - \ell^*_{xent}(\Phi) \leq \epsilon$, where $\ell^*_{xent}(\Phi)$ is defined in Equation* (5.4)*. For a classification task $\mathcal{T}$ that is (τ, B)-natural w.r.t. Φ, we have*

$$\ell_{\mathcal{T}}\left(\{\boldsymbol{p}_{f(s)}\}\right) \leq \ell_{\mathcal{T}}(\Phi p_f) \leq \tau + \sqrt{\frac{2B^2\epsilon}{\gamma(p_{\mathcal{T}}; \Phi p_f)}}$$

Discussions: It is not hard to show that the coefficients satisfy $\gamma_\Phi(p_{\mathcal{T}}; \{\boldsymbol{p}_{\cdot|s}\}) \geq \gamma(p_{\mathcal{T}}; \{\boldsymbol{p}_{\cdot|s}\}) \geq \gamma(p_{\mathcal{T}})$ and $\gamma(p_{\mathcal{T}}; \Phi p_f) \geq \gamma(p_{\mathcal{T}})$, thus showing that these results are strictly stronger than the ones from the previous section. The transferability coefficient is a measure of how guarantees on p_L using a language model can be transferred to another distribution of contexts and it only depends on the distribution of contexts and not the labels. Unlike $\gamma(p_{\mathcal{T}})$, the coefficients in Definition 5.9.1 depend on the learned models, either $\{\boldsymbol{p}_{\cdot|s}\}$ or $\{\boldsymbol{p}_{f(s)}\}$, and can be potentially much smaller due to the inductive bias of the learned models. For instance, if errors made by the model are random-like, i.e. $\Delta_{\{\boldsymbol{p}_{\cdot|s}\}}(s) \sim \rho$, independently of s, then $\Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}}) \approx \Sigma_p(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}}) \approx \mathbb{E}_{\eta\sim\rho}[\eta\eta^\top]$, making $\gamma(p; \{\boldsymbol{p}_{\cdot|s}\}) \approx 1$. Independence prevents language modeling error from accumulating on contexts from $p_{\mathcal{T}}$, bypassing the worst case transfer of $\gamma(p_{\mathcal{T}})$.

5.10 More on Quad

In Definition 5.3.2 we discuss how low dimensional softmax language models learn a linear projection of $\boldsymbol{p}^*_{\cdot|s}$, only solving tasks that lie in the row span of word embeddings Φ. Although Φ defines tasks that language model features can solve, the standard cross-entropy objective does not lend a simple closed form expression for optimal Φ. This motivates the construction of our Quad objective, that has two nice properties: (1) the optimal feature map f^* is a linear function of $\boldsymbol{p}^*_{\cdot|s}$ and thus can solve some natural tasks, and (2) the optimal Φ^* has an intuitively meaningful closed-form solution.

$$\ell_{quad,s}(\theta, \Phi) = \mathop{\mathbb{E}}_{w \sim p^*_{\cdot|s}} [-\theta^\top \phi_w] + \frac{1}{2}\|\Phi^\top \theta\|^2 = -\theta^\top \Phi \boldsymbol{p}^*_{\cdot|s} + \frac{1}{2}\|\Phi^\top \theta\|^2 \tag{5.10}$$

$$\ell_{quad}(f, \Phi) = \mathop{\mathbb{E}}_{s \sim p_L} [\ell_{quad,s}(f(s), \Phi)] \tag{5.11}$$

The Quad objective is very similar to the cross-entropy objective from Equation (5.2), with the log partition function replaced by a quadratic function, inspired in part by Assumption 5.4.4. We can derive the optimal solution Φ^* that depends on the eigen-decomposition of a *substitutability matrix*.

Definition 5.5.2. *The substitutability matrix is defined to be* $\Omega^* \coloneqq \mathop{\mathbb{E}}_{s \sim p_L} \left[\boldsymbol{p}^*_{\cdot|s}\ \boldsymbol{p}^{*\top}_{\cdot|s}\right] \in \mathbb{R}^{V \times V}$. *If* $\Omega^* = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^\top$ *is the eigendecomposition, then* $\boldsymbol{U}_d \in \mathbb{R}^{V \times d}$ *is matrix of top* d *eigenvectors of* Ω^*.

The matrix Ω^* captures substitutability between pairs of words. Words w and w' are substitutable if they have identical conditional probabilities for every context $s \in \mathcal{S}$ and thus can replace occurrences of each other while still providing meaningful completions. By definition, these words satisfy $\Omega^*[w] = \Omega^*[w']$. Such pairs of words were called "free variants" in the work on distributional semantics [Harris, 1954], and capture the notion of synonyms in the distributional hypothesis. We now derive expressions for the optimal solution of the Quad objective described in Equation (5.11). The proof of all results from this section are in Section 5.12.5.

Theorem 5.10.1. *The optimal solution* $f^*, \Phi^* = \arg\min_{f,\Phi} \ell_{quad}(f, \Phi)$ *satisfies*

$$\Phi^* = \boldsymbol{B}\boldsymbol{U}_d^\top, \text{ for full rank } \boldsymbol{B} \in \mathbb{R}^{d \times d}$$
$$f^*(s) = (\Phi^* \Phi^{*\top})^{-1/2} \Phi^* \boldsymbol{p}^*_{\cdot|s} = \boldsymbol{C}\boldsymbol{U}_d^\top \boldsymbol{p}^*_{\cdot|s}, \text{ for full rank } \boldsymbol{C} \in \mathbb{R}^{d \times d}$$

If Φ *is fixed, then the optimal solution is* $f^*(s) = (\Phi\Phi^\top)^{-1/2}\Phi \boldsymbol{p}^*_{\cdot|s}$.

Theorem 5.5.3. *Let* $f^*, \Phi^* = \arg\min_{f,\Phi} \ell_{quad}(f, \Phi)$. *Then* $\Phi^* = \boldsymbol{B}\boldsymbol{U}_d^\top$, *for full rank* $\boldsymbol{B} \in \mathbb{R}^{d \times d}$. *Also, for*

a classification task $\mathcal{T}$ that is (τ, B)-natural w.r.t. Φ^*, we have $\ell_{\mathcal{T}}(f^*) \leq \tau$.

Thus f^* excels on natural tasks w.r.t. Φ^*, which in turn, is the best d-dimensional projection of Ω^*. Thus words $w, w' \in \mathcal{W}$ that are synonyms (hence substitutable) will satisfy $\phi^*_w = \phi^*_{w'}$, fulfilling the desired property for word embeddings discussed in Definition 5.3.2. We train using the Quad objective and compare its performance to a similarly trained language model, finding Quad to be reasonably effective. The goal of testing Quad is not to obtain state-of-the-art results, but to demonstrate that theoretical insights can aid the design of provably effective algorithms.

5.11 More on natural tasks

The discussions in this section may not be formal and precise in places, they are meant to provide more intuition for some of the definitions and results.

5.11.1 Sentence completion reformulation ≡ natural task

We provide informal justification for why the sentence completion reformulation can be formalized as being able to solve using a linear classifier over $\boldsymbol{p}^*_{\cdot|s} \in \mathbb{R}^V$. The analysis will also end up providing some intuitions for τ and B in Definition 5.3.1 and Theorem 5.4.1. In particular, we will show that a task that is amenable to the sentence completion reformulation will be (τ, B)-natural, with $\tau = \mathcal{O}(\text{Bayes-error}(\mathcal{T}))$, i.e. τ is small if the Bayes error for the task error, and $B = \mathcal{O}(\alpha(\mathcal{W}_{\text{indicative}})^{-1})$ is inversely proportional to the probability mass of the set of indicative words for the task. This is formalized in Proposition 5.11.2.

Linear classifier over $\boldsymbol{p}^*_{\cdot|s}$

Consider a binary classification task $\mathcal{T}$ and that can be solved with a sentence completion reformulation after adding a prompt as in Section 5.3.1, for e.g. sentiment classification can be solved by adding a prompt "This movie is" at the end of every movie review and use the completions to solve the task. Recall that $p_{\mathcal{T}}$ is the distribution over $\mathcal{S} \times \{\pm 1\}$ for the task $\mathcal{T}$. We abuse notation and use $p_{\mathcal{T}}$ to denote the distribution over inputs where a prompt is added to each to input, for e.g. "I loved the movie." is transformed to "I loved the movie. This movie is". For any $s \sim p_{\mathcal{T}}$, let $p_{\mathcal{T}}(y = 1|s)$ and $p_{\mathcal{T}}(y = -1|s)$ denote the conditional probabilities of the sentiment of review s (with an added prompt) being positive and negative respectively.

By law of total probability we can write this conditional probability as

$$p_{\mathcal{T}}(y=1|s) = \sum_{w\in\mathcal{W}} \Pr(y=1|(s,w)) \Pr(w|s) = \sum_{w\in\mathcal{W}} \Pr(y=1|(s,w))\ p^*_{\cdot|s}(w) \tag{5.12}$$

For any task $\mathcal{T}$ we can roughly partition the vocabulary set $\mathcal{W}$ into the following

Indicative words $\mathcal{W}_{\text{indicative}}$: w can be an *indicative completion* for the task, like "good", "boring", "trash" etc, after a movie review like $s =$"I loved the movie. This movie is". In this case the sentence completion reformulation can be interpreted as the following: the completion w after a review s is sufficient to determine the sentiment of the review, i.e. we do not need to know the content of the review s to predict the label if we know the completion w. This can be formalized as $\Pr(y=1|(s,w)) \approx P(y=1|w)$ for some fixed distribution P for indicative completions w.

Irrelevant words $\mathcal{W}_{\text{irrelevant}}$: w can be an *irrelevant completion* for the task, like "a", "very", "not". In this case the completions, on the other hand, do not reveal anything more about the sentiment for the review than s itself, i.e. $\Pr(y=1|(s,w)) \approx p_{\mathcal{T}}(y=1|s)$ for irrelevant completions w.

Thus from Equation (5.12) we get

$$\begin{aligned}
p_{\mathcal{T}}(y=1|s) &= \sum_{w\in\mathcal{W}_{\text{indicative}}} \Pr(y=1|(s,w))\ p^*_{\cdot|s}(w) + \sum_{w\in\mathcal{W}_{\text{irrelevant}}} \Pr(y=1|(s,w))\ p^*_{\cdot|s}(w) \\
&\approx \sum_{w\in\mathcal{W}_{\text{indicative}}} P(y=1|w)\ p^*_{\cdot|s}(w) + \sum_{w\in\mathcal{W}_{\text{irrelevant}}} p_{\mathcal{T}}(y=1|s)\ p^*_{\cdot|s}(w) \\
&= \sum_{w\in\mathcal{W}_{\text{indicative}}} \boldsymbol{v}_1(w)\ p^*_{\cdot|s}(w) + p_{\mathcal{T}}(y=1|s) \sum_{w\in\mathcal{W}_{\text{irrelevant}}} p^*_{\cdot|s}(w) \\
&= \boldsymbol{v}_1^\top \boldsymbol{p}^*_{\cdot|s} + p_{\mathcal{T}}(y=1|s) p^*_{\cdot|s}(\mathcal{W}_{\text{irrelevant}})
\end{aligned}$$

where $\boldsymbol{v}_1 \in \mathbb{R}^V$ is defined as $\boldsymbol{v}_1(w) = P(y=1|w)$ for $w \in \mathcal{W}_{\text{indicative}}$ and $\boldsymbol{v}_1(w) = 0$ for $w \in \mathcal{W}_{\text{irrelevant}}$. Similarly we can define $\boldsymbol{v}_{-1} \in \mathbb{R}^V$ with $\boldsymbol{v}_{-1}(w) = P(y=-1|w)$ for $w \in \mathcal{W}_{\text{indicative}}$, $\boldsymbol{v}_{-1}(w) = 0$ for $w \in \mathcal{W}_{\text{irrelevant}}$. From the earlier calculation, and a similar one for $y=-1$, we get

$$p_{\mathcal{T}}(y=b|s) \approx \frac{1}{1-p^*_{\cdot|s}(\mathcal{W}_{\text{irrelevant}})} \boldsymbol{v}_b^\top \boldsymbol{p}^*_{\cdot|s} = \frac{1}{p^*_{\cdot|s}(\mathcal{W}_{\text{indicative}})} \boldsymbol{v}_b^\top \boldsymbol{p}^*_{\cdot|s}, \text{ for } b \in \{\pm 1\}$$

If we assume $p^*_{\cdot|s}(\mathcal{W}_{\text{indicative}}) \approx \alpha(\mathcal{W}_{\text{indicative}})$ is roughly the same for all s, i.e. probability mass of indicative

words following a modified review is approximately the same, then we get

$$p_{\mathcal{T}}(y=1|s) - p_{\mathcal{T}}(y=-1|s) \approx \boldsymbol{v}_{\mathcal{T}}^{\top} \boldsymbol{p}^*_{\cdot|s} \quad \text{, where } \boldsymbol{v}_{\mathcal{T}} = \frac{1}{\alpha(\mathcal{W}_{\text{indicative}})}(\boldsymbol{v}_1 - \boldsymbol{v}_{-1}) \tag{5.13}$$

Thus we can approximately express the difference in conditional probabilities of the 2 classes as a linear function of $\boldsymbol{p}^*_{\cdot|s}$. While it is intuitively clear why knowing $p_{\mathcal{T}}(y=1|s) - p_{\mathcal{T}}(y=-1|s)$ is useful for solving the task, we show precisely why in the next part.

Interpretation for τ and B

Based on the above discussed, we will show that the task $\mathcal{T}$ from earlier is (τ, B)-natural according to the Definition 5.3.1 and will also give us an interpretation for τ and B. First we show that the following predictor from Equation (5.13) is effective for task $\mathcal{T}$

$$g_{\mathcal{T}}(s) = p_{\mathcal{T}}(y=1|s) - p_{\mathcal{T}}(y=-1|s) \approx \boldsymbol{v}_{\mathcal{T}}^{\top} \boldsymbol{p}^*_{\cdot|s} \tag{5.14}$$

We reuse the notation from Equation (5.3) and define the task loss for any predictor $g : \mathcal{S} \to \mathbb{R}$ as

$$\ell_{\mathcal{T}}(g) = \mathbb{E}_{(s,y)\sim p_{\mathcal{T}}} \left[\ell(g(s), y)\right] \tag{5.15}$$

Furthermore let Bayes-error$(\mathcal{T}) \coloneqq \inf_{g:\mathcal{S}\to\mathbb{R}} \mathbb{E}_{(s,y)\sim p_{\mathcal{T}}}[\mathbf{1}\{g(s) \neq y\}]$ denote the Bayes error of the task $\mathcal{T}$, i.e. the optimal $0-1$ error achievable on the task.

Proposition 5.11.1. *For any task $\mathcal{T}$ and for the hinge loss ℓ, $\ell_{\mathcal{T}}(g_{\mathcal{T}}) \leq 4$ Bayes-error$(\mathcal{T})$, where $g_{\mathcal{T}}(s) = p_{\mathcal{T}}(y=1|s) - p_{\mathcal{T}}(y=-1|s)$.*

Thus if a task is easily solvable, i.e. has small Bayes error, then it will be solvable by the predictor $g_{\mathcal{T}}(s)$. Since we argued above that sentence reformulation implies that $g_{\mathcal{T}}(s)$ is a linear function of $\boldsymbol{p}^*_{\cdot|s}$, we can now show that $\mathcal{T}$ is a *natural task* as formalized in Definition 5.3.1.

Proposition 5.11.2 (Informal)**.** *Task $\mathcal{T}$ that can be reformulated as a sentence completion task (described above) is a (τ, B)-natural task w.r.t. the hinge loss, with the follow parameters*

$$\boldsymbol{\tau \leq 4} \textit{ Bayes-error}\boldsymbol{(\mathcal{T})} \textit{ and } \boldsymbol{B = \alpha(\mathcal{W}_{\text{indicative}})^{-1}}$$

Here Bayes-error($\mathcal{T}$) is the Bayes error of task $\mathcal{T}$ and $\alpha(\mathcal{W}_{indicative})$ is the total mass of the indicative words for the task.

If the task $\mathcal{T}$ can be reformulated as sentence completion, then $\mathcal{T}$ is (τ, B)-natural where

- τ is small if the task is unambiguous, i.e. it has small Bayes error
- B is small if the probability mass of the set of indicative words $\mathcal{W}_{\text{indicative}}$ is large, i.e. the task depends on a large set of frequent words

Thus the upper bound in Theorem 5.4.1 is smaller if the task can be reformulated as sentence completion task with a large and frequent set of completions, and we can ever hope to solve it well (Bayes error is small). The proofs for the above propositions are in Section 5.11.1.

5.11.2 Nice properties of word embeddings Φ

We argue here that if the word embeddings Φ satisfy certain *nice properties*, then (τ, B)-natural *tasks of interest* will be (τ', B')-natural w.r.t. Φ, where we will provide informal quantifications for the *nice properties* and *tasks of interest* that lead to a small value for τ' and B'. The *nice property* will be related to Φ capturing the semantic meaning (synonym structure) of words and *tasks of interest* will be those that try to distinguish word completion (in the sentence completion reformulation) with very different meanings, i.e. tries to distinguish more coarse-grained semantic notions rather than very fine-grained ones. Note that the results here are informal and qualitative, rather than quantitative.

Consider a task $\mathcal{T}$ that is (τ, B)-natural task and let $\boldsymbol{v}^* \in \mathbb{R}^V$ be the classifier such that $\ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\}, \boldsymbol{v}^*) \leq \tau$ and $\|\boldsymbol{v}^*\|_\infty \leq B$. We want to find properties of Φ and $\boldsymbol{v}^*$ that will make $\mathcal{T}$ to be (τ', B')-natural w.r.t. Φ such that τ' and B' are not too large.[6]

We will show that $\mathcal{T}$ is (τ', B')-natural w.r.t. Φ by finding a classifier $\boldsymbol{v}$ such that $\boldsymbol{v} = \Phi^\top \lambda \in \mathbb{R}^V$, $\|\boldsymbol{v}\|_\infty \leq B'$ and $\ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\}, \boldsymbol{v}) \leq \tau'$. First we define $P_\Phi \coloneqq \Phi^\dagger \Phi \in \mathbb{R}^{V \times V}$ to be the projection matrix for the row-span of Φ and $P_\Phi^\perp \coloneqq I_V - P_\Phi$ to be orthogonal projection matrix. We will show that the classifier $\boldsymbol{v} = P_\Phi \boldsymbol{v}^*$ suffices for our case, under some intuitive conditions on $\boldsymbol{v}^*$ and Φ.

[6] Note that the converse is trivially true, i.e. a (τ, B)-natural task w.r.t. Φ is also (τ, B)-natural.

To compute B', we first look at the ℓ_∞ norm of $\boldsymbol{v} = P_\Phi \boldsymbol{v}^*$

$$B' = \|\boldsymbol{v}\|_\infty = \|P_\Phi \boldsymbol{v}^*\|_\infty = \|\boldsymbol{v}^* - P_\Phi^\perp \boldsymbol{v}^*\|_\infty \leq \|\boldsymbol{v}^*\|_\infty + \|P_\Phi^\perp \boldsymbol{v}^*\|_\infty \leq B + \|P_\Phi^\perp \boldsymbol{v}^*\|_2$$

To find the upper bound τ', we upper bound the classification loss of $\boldsymbol{v} = P_\Phi \boldsymbol{v}^*$. We first define the substitutability matrix $\Omega_p^* = \mathop{\mathbb{E}}_{s\sim p}\left[\boldsymbol{p}^*_{\cdot|s}{\boldsymbol{p}^*_{\cdot|s}}^\top\right]$, similar to the one in Definition 5.5.2. Then

$$\begin{aligned}
\ell_\mathcal{T}(\{\boldsymbol{p}^*_{\cdot|s}\}, \boldsymbol{v}) &= \mathop{\mathbb{E}}_{(s,y)\sim p_\mathcal{T}}\left[\ell(\boldsymbol{v}^\top \boldsymbol{p}^*_{\cdot|s}, y)\right] = \mathop{\mathbb{E}}_{(s,y)\sim p_\mathcal{T}}\left[\ell((P_\Phi \boldsymbol{v}^*)^\top \boldsymbol{p}^*_{\cdot|s}, y)\right] \\
&\leq^{(a)} \mathop{\mathbb{E}}_{(s,y)\sim p_\mathcal{T}}\left[\ell({\boldsymbol{v}^*}^\top \boldsymbol{p}^*_{\cdot|s}, y)\right] + \mathop{\mathbb{E}}_{s\sim p_\mathcal{T}}[|(\boldsymbol{v}^* - P_\Phi \boldsymbol{v}^*)^\top \boldsymbol{p}^*_{\cdot|s}|] \\
&= \ell_\mathcal{T}(\{\boldsymbol{p}^*_{\cdot|s}\}, \boldsymbol{v}^*) + \mathop{\mathbb{E}}_{s\sim p_\mathcal{T}}\left[|{\boldsymbol{v}^*}^\top P_\Phi^\perp \boldsymbol{p}^*_{\cdot|s}|\right] \\
&\leq^{(b)} \tau + \sqrt{\mathop{\mathbb{E}}_{s\sim p_\mathcal{T}}\left[({\boldsymbol{v}^*}^\top P_\Phi^\perp \boldsymbol{p}^*_{\cdot|s})^2\right]} = \tau + \sqrt{\mathop{\mathbb{E}}_{s\sim p_\mathcal{T}}\left[{\boldsymbol{v}^*}^\top P_\Phi^\perp \boldsymbol{p}^*_{\cdot|s} {\boldsymbol{p}^*_{\cdot|s}}^\top P_\Phi^\perp \boldsymbol{v}^*\right]} \\
&=^{(c)} \tau + \sqrt{{\boldsymbol{v}^*}^\top P_\Phi^\perp \Omega^*_{p_\mathcal{T}} P_\Phi^\perp \boldsymbol{v}^*} \leq^{(d)} \tau + \|P_\Phi^\perp \boldsymbol{v}^*\|_2 \sqrt{\left\|P_\Phi^\perp \Omega^*_{p_\mathcal{T}} P_\Phi^\perp\right\|_2}
\end{aligned}$$

where (a) follows from 1-Lipschitz property of ℓ, (b) from Jensen's inequality and that $\ell_\mathcal{T}(\{\boldsymbol{p}^*_{\cdot|s}\}, \boldsymbol{v}^*) \leq \tau$, (c) from the definition of substitutability matrix $\Omega^*_{p_\mathcal{T}}$ and (d) by definition of spectral norm of a symmetric PSD matrix.

Thus we have shown that $\mathcal{T}$ is (τ', B')-natural w.r.t. Φ, where

$$\tau' = \tau + \|P_\Phi^\perp \boldsymbol{v}^*\|_2 \sqrt{\left\|P_\Phi^\perp \Omega^*_{p_\mathcal{T}} P_\Phi^\perp\right\|_2},\ B' = B + \|P_\Phi^\perp \boldsymbol{v}^*\|_2 \tag{5.16}$$

We will now show that if Φ captures the notion of synonyms, then $\left\|P_\Phi^\perp \Omega^*_{p_\mathcal{T}} P_\Phi^\perp\right\|_2$ will be small leading to τ' being small. Furthermore we also shed some light on what it means for $\|P_\Phi^\perp \boldsymbol{v}^*\|_2$ to be small, which will in turn make B' small and τ' smaller. We do so with the following arguments, 1) $\Omega^*_{p_\mathcal{T}}$ captures semantic meaning of words and thus its top eigen-directions will capture more dominant semantic concepts, 2) if Φ captures the "top-d" directions of meaning, i.e. the top-d eigen-directions of $\Omega^*_{p_\mathcal{T}}$, then $\left\|P_\Phi^\perp \Omega^*_{p_\mathcal{T}} P_\Phi^\perp\right\|_2 = \mathcal{O}(1/d)$, 3) if additionally $\boldsymbol{v}^*$ cares about the "top-d" directions of meaning, i.e. top-d eigen-directions of $\Omega^*_{p_\mathcal{T}}$ then $\|P_\Phi^\perp \boldsymbol{v}^*\|_2$ will be small. We expand on these points below

1. **Substitutability matrix ($\Omega^*_{p_\mathcal{T}}$)** captures semantic meaning: We use a similar argument to the one in Section 5.5.2 right after Definition 5.5.2 that is based on distributional semantics [Harris, 1954]. Harris

[1954] posits that meaning for elements (words) can be derived from the environments (contexts) in which they occur. Thus Harris [1954] argues that words that occur in almost identical set of contexts have the same meaning, i.e. are synonyms. On the other hand, if two words share some contexts but not all, then they have different meanings and the amount of difference in meaning roughly corresponds to amount of difference in contexts. In our setting, the similarity of words w and w' can then be determined by the probabilities assigned to them by different contexts s. In particular, if $\boldsymbol{p}^*_{\cdot|s}(w) = \boldsymbol{p}^*_{\cdot|s}(w')$ for all or most $s \in \text{supp}(p_{\mathcal{T}})$, then w and w' have essentially the same meaning w.r.t. the distribution of contexts $p_{\mathcal{T}}$ and the closer $[\boldsymbol{p}^*_{\cdot|s}(w)]_{s\in\text{supp}(p_{\mathcal{T}})}$ and $[\boldsymbol{p}^*_{\cdot|s}(w')]_{s\in\text{supp}(p_{\mathcal{T}})}$ are, the closer the meaning of w and w' are. For the substitutability matrix $\Omega^*_{p_{\mathcal{T}}} = \mathop{\mathbb{E}}_{s\sim p_{\mathcal{T}}}[\boldsymbol{p}^*_{\cdot|s}\boldsymbol{p}^*_{\cdot|s}{}^\top] \in \mathbb{R}^{V\times V}$, it is not hard to show that $\Omega^*_{p_{\mathcal{T}}}(w) = \Omega^*_{p_{\mathcal{T}}}(w')$ is equivalent to $\boldsymbol{p}^*_{\cdot|s}(w) = \boldsymbol{p}^*_{\cdot|s}(w')\ \forall s \sim p_{\mathcal{T}}$, where $\Omega^*_{p_{\mathcal{T}}}(w)$ is the row of $\Omega^*_{p_{\mathcal{T}}}$ corresponding to word w. To show this, we can define $\boldsymbol{\beta}_w \in \mathbb{R}^{|\text{supp}(p_{\mathcal{T}})|}$ to be an embedding of w that looks like $\boldsymbol{\beta}_w = [\boldsymbol{p}^*_{\cdot|s}(w)\sqrt{p_{\mathcal{T}}(s)}]_{s\in\text{supp}(p_{\mathcal{T}})}$. It is easy to see that $\boldsymbol{\beta}_{w_1}^\top\boldsymbol{\beta}_{w_2} = \mathop{\mathbb{E}}_{s\sim p_{\mathcal{T}}}\left[\boldsymbol{p}^*_{\cdot|s}(w_1)\boldsymbol{p}^*_{\cdot|s}(w_2)\right] = \Omega^*_{p_{\mathcal{T}}}(w_1, w_2)$. Thus $\boldsymbol{\beta}_w = \boldsymbol{\beta}_{w'} \implies \Omega^*_{p_{\mathcal{T}}}(w) = \Omega^*_{p_{\mathcal{T}}}(w')$ is straightforward to see. For the converse,

$$\Omega^*_{p_{\mathcal{T}}}(w) = \Omega^*_{p_{\mathcal{T}}}(w') \implies \Omega^*_{p_{\mathcal{T}}}(w,w) = \Omega^*_{p_{\mathcal{T}}}(w',w) = \Omega^*_{p_{\mathcal{T}}}(w,w') = \Omega^*_{p_{\mathcal{T}}}(w',w') \tag{5.17}$$

$$\implies \boldsymbol{\beta}_w^\top\boldsymbol{\beta}_w = \boldsymbol{\beta}_w^\top\boldsymbol{\beta}_{w'} = \boldsymbol{\beta}_{w'}^\top\boldsymbol{\beta}_{w'} \implies \boldsymbol{\beta}_w = \boldsymbol{\beta}_{w'} \tag{5.18}$$

Thus $\Omega^*_{p_{\mathcal{T}}}$ indeed does capture the synonyms structure between words, and the top eigen-directions of it capture the most significant "semantic meaning" directions.

2. Φ **has nice properties**: if Φ roughly respects this synonym structure by aligning with the top-d eigen-directions of $\Omega^*_{p_{\mathcal{T}}}$, we have

$$\left\|P_\Phi^\perp\Omega^*_{p_{\mathcal{T}}}P_\Phi^\perp\right\|_2 \le \lambda_{d+1}(\Omega^*_{p_{\mathcal{T}}}) \le \frac{1}{d+1}\sum_{i=1}^{d+1}\lambda_i(\Omega^*_{p_{\mathcal{T}}}) \le \frac{1}{d+1}\text{tr}(\Omega^*_{p_{\mathcal{T}}}) \tag{5.19}$$

$$\le \frac{1}{d+1}\mathop{\mathbb{E}}_{s\sim p_{\mathcal{T}}}\text{tr}(\boldsymbol{p}^*_{\cdot|s}\boldsymbol{p}^*_{\cdot|s}{}^\top) \le \frac{1}{d+1} \tag{5.20}$$

From Equation (5.16), we then have $\tau' \le \tau + \frac{\|P_\Phi^\perp\boldsymbol{v}^*\|_2}{\sqrt{d}}$

3. **Tasks of interest**: It is more likely for a classifier $\boldsymbol{v}^*$ to separate words with big differences in meaning rather than small differences. For e.g., it is more likely for a task to separate word completions "good" and "bad" rather than "good" and "nice". Since top eigen-directions of $\Omega^*_{p_{\mathcal{T}}}$ capture more dominant semantic

meanings, this could correspond to $\boldsymbol{v}^*$ aligning with the top eigen-directions of $\Omega^*_{p_\mathcal{T}}$. In combination with the above property about Φ, this could suggest that $\|P^{\perp}_{\Phi}\boldsymbol{v}^*\|_2$ is small, thus leading to τ' and B' being small.

Note that they above arguments are informal and qualitative, and we leave exploring desirable properties of Φ more formally to future work.

5.11.3 Proofs for Section 5.11.1

Proposition 5.11.1. Let $p_b(s) = p_\mathcal{T}(y = b|s)$ for $b \in \{\pm 1\}$, $p_{min}(s) = \min_{b\in\{\pm1\}} p_b(s)$, $p_{max}(s) = \max_{b\in\{\pm1\}} p_b(s)$ and $g^*(s) = \arg\max_{b\in\{\pm1\}} p_b(s)$ denote the Bayes optimal predictor. We first notice that there is a simple well-known closed form expression for the Bayes risk

$$\begin{aligned}\text{Bayes-error}(\mathcal{T}) &= \mathop{\mathbb{E}}_{(s,y)\sim p_\mathcal{T}}\left[\mathbf{1}\left\{g^*(s) \neq y\right\}\right] \\ &= \mathop{\mathbb{E}}_{(s,y)\sim p_\mathcal{T}}\left[\mathbf{1}\left\{\arg\max_{b\in\{\pm1\}} p_b(s) \neq y\right\}\right] = \mathop{\mathbb{E}}_{s\sim p_\mathcal{T}}\left[p_{min}(s)\right]\end{aligned}$$

□

We now analyze the hinge loss of the predictor $g_{p_\mathcal{T}}$ defined in Equation (5.14). Note that since $g_{p_\mathcal{T}}(s) \leq 1$, the hinge loss $\ell(g_{p_\mathcal{T}}(s), y) = (1 - yg_{p_\mathcal{T}}(s))_+ = 1 - yg_{p_\mathcal{T}}(s)$ for every s, y. Thus the total loss is

$$\begin{aligned}g_{p_\mathcal{T}}(s) &= \mathop{\mathbb{E}}_{(s,y)\sim p_\mathcal{T}}\left[(1 - yg_{p_\mathcal{T}}(s))_+\right] = \mathop{\mathbb{E}}_{(s,y)\sim p_\mathcal{T}}\left[(1 - yg_{p_\mathcal{T}}(s))\right] \\ &=^{(a)} \mathop{\mathbb{E}}_{s\sim p_\mathcal{T}}\left[p_1(s)\left(1 - g_{p_\mathcal{T}}(s)\right) + p_{-1}(s)\left(1 + g_{p_\mathcal{T}}(s)\right)\right] = \mathop{\mathbb{E}}_{s\sim p_\mathcal{T}}\left[1 - (p_1(s) - p_{-1}(s))g_{p_\mathcal{T}}(s)\right] \\ &=^{(b)} \mathop{\mathbb{E}}_{s\sim p_\mathcal{T}}\left[1 - (p_1(s) - p_{-1}(s))^2\right] = \mathop{\mathbb{E}}_{s\sim p_\mathcal{T}}\left[(p_1(s) + p_{-1}(s))^2 - (p_1(s) - p_{-1}(s))^2\right] \\ &= \mathop{\mathbb{E}}_{s\sim p_\mathcal{T}}\left[4p_1(s)p_{-1}(s)\right] = 4\mathop{\mathbb{E}}_{s\sim p_\mathcal{T}}\left[p_{min}(s)p_{max}(s)\right] \\ &\leq^{(c)} 4\mathop{\mathbb{E}}_{s\sim p_\mathcal{T}}\left[p_{min}(s)\right] = 4\ \text{Bayes-error}(\mathcal{T})\end{aligned}$$

where (a) follows by splitting the expectation over $y|s$, (b) follows from the definition of $g_{p_\mathcal{T}}(s)$ in Equation (5.14) and (c) follows from $p_{max}(s) \leq 1$. This completes the proof.

Proposition 5.11.2. Let $B = \alpha(\mathcal{W}_{\text{indicative}})^{-1}$. We first note the following using the definition of $\boldsymbol{v}$ from

Equation (5.13).

$$\|\boldsymbol{v}_{\mathcal{T}}\|_\infty = \alpha(\mathcal{W}_{\text{indicative}})^{-1} \max_{w \in \mathcal{W}} |\boldsymbol{v}_1(w) - \boldsymbol{v}_{-1}(w)| = B \max_{w \in \mathcal{W}} |P(y=1|w) - P(y=-1|w)| \leq B \tag{5.21}$$

To find the value of τ that makes the task (τ, B)-natural (Definition 5.3.1), we observe the following

$$\begin{aligned}
\min_{\boldsymbol{v} \in \mathbb{R}^V, \|\boldsymbol{v}\| \leq B} \ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\}, \boldsymbol{v}) &=^{(a)} \ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\}, \boldsymbol{v}_{\mathcal{T}}) = \mathop{\mathbb{E}}_{(s,y)\sim p_{\mathcal{T}}} [\ell(\boldsymbol{v}_{\mathcal{T}}^\top \boldsymbol{p}^*_{\cdot|s}, y)] \\
&=^{(b)} \mathop{\mathbb{E}}_{(s,y)\sim p_{\mathcal{T}}} [\ell(g_{\mathcal{T}}(s), y)] = \ell_{\mathcal{T}}(g_{\mathcal{T}}) \\
&\leq^{(c)} 4 \text{ Bayes-error}(\mathcal{T})
\end{aligned}$$

where (a) follows from the calculation in Equation (5.21), (b) follows from Equation (5.13) and (c) follows from Proposition 5.11.1. □

5.12 Omitted proofs

5.12.1 Proof sketch

We first present a sketch of the arguments that help us show our main results, Theorems 5.4.1 and 5.4.3. The subsections after the next one contain the full proofs for strengthened versions of these results.

Proof sketch for arbitrary language models: Theorem 5.4.1

Here we want to show guarantees for features $\{\boldsymbol{p}_{\cdot|s}\}$ on a (τ, B)-natural task $\mathcal{T}$. From the definition of natural tasks, we know

$$\exists \boldsymbol{v}^* \in \mathbb{R}^V, \|\boldsymbol{v}^*\|_\infty \leq B \text{ s.t. } \ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\}, \boldsymbol{v}^*) \leq \tau \tag{5.22}$$

We wish to upper bound the classification error $\ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\})$ and do so using the following sequence of inequalities.

$$\ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\}) - \tau = \inf_{\boldsymbol{v} \in \mathbb{R}^V} \ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\}, \boldsymbol{v}) - \tau \leq \ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\}, \boldsymbol{v}^*) - \ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\}, \boldsymbol{v}^*)$$

$$= \frac{\ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\}, \boldsymbol{v}^*) - \ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\}, \boldsymbol{v}^*)}{\sqrt{\mathop{\mathbb{E}}_{s\sim p_{\mathcal{T}}}[(\boldsymbol{v}^{*\top}(\boldsymbol{p}_{\cdot|s} - \boldsymbol{p}^*_{\cdot|s}))^2]}} \cdot \sqrt{\frac{\mathop{\mathbb{E}}_{s\sim p_{\mathcal{T}}}[(\boldsymbol{v}^{*\top}(\boldsymbol{p}_{\cdot|s} - \boldsymbol{p}^*_{\cdot|s}))^2]}{\mathop{\mathbb{E}}_{s\sim p_L}[(\boldsymbol{v}^{*\top}(\boldsymbol{p}_{\cdot|s} - \boldsymbol{p}^*_{\cdot|s}))^2]}} \cdot \sqrt{\mathop{\mathbb{E}}_{s\sim p_L}[(\boldsymbol{v}^{*\top}(\boldsymbol{p}_{\cdot|s} - \boldsymbol{p}^*_{\cdot|s}))^2]}$$

$$= \underbrace{\frac{\ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\}, \boldsymbol{v}^*) - \ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\}, \boldsymbol{v}^*)}{\sqrt{\boldsymbol{v}^{*\top}\Sigma_{p_{\mathcal{T}}}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v}^*}}}_{\boldsymbol{\alpha_1(v^*)}} \cdot \underbrace{\sqrt{\frac{\boldsymbol{v}^{*\top}\Sigma_{p_{\mathcal{T}}}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v}^*}{\boldsymbol{v}^{*\top}\Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v}^*}}}_{\boldsymbol{\alpha_2(v^*)}} \cdot \underbrace{\sqrt{\mathop{\mathbb{E}}_{s\sim p_L}[(\boldsymbol{v}^{*\top}(\boldsymbol{p}_{\cdot|s} - \boldsymbol{p}^*_{\cdot|s}))^2]}}_{\boldsymbol{\alpha_3(v^*)}} \tag{5.23}$$

Classification loss → error covariance on $p_{\mathcal{T}}$
Use Lipschitzness of ℓ and Jensen's inequality

Error covariance from $p_{\mathcal{T}} \to p_L$
Use transferability coefficient

Error covariance → cross-entropy loss
Use (modified) Pinsker's inequality

where $\Sigma_p(g) \coloneqq \mathop{\mathbb{E}}_{s\sim p}[g(s)g(s)^\top]$ is the uncentered covariance of g w.r.t. distribution $p \in \Delta_{\mathcal{S}}$, as defined in Section 5.5.1. We upper bound $\ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\}) - \tau$ by upper bounding each of $\boldsymbol{\alpha_1}(\boldsymbol{v}^*), \boldsymbol{\alpha_2}(\boldsymbol{v}^*), \boldsymbol{\alpha_3}(\boldsymbol{v}^*)$ as follows

- **Classification loss → prediction error covariance:** $\boldsymbol{\alpha_1}(\boldsymbol{v}^*)$ is upper bounded by using Lipschitzness of the loss ℓ used in the definition of $\ell_{\mathcal{T}}$, e.g. hinge loss or logistic loss, and then followed by an application of Jensen's inequality

$$\text{Lemma 5.12.8} \implies \boldsymbol{\alpha_1}(\boldsymbol{v}) \le 1 \text{ for all } \boldsymbol{v} \in \mathbb{R}^V$$

- **Error covariance from $p_{\mathcal{T}} \to p_L$:** $\boldsymbol{\alpha_2}(\boldsymbol{v}^*)$ handles the mismatch in distributions $p_{\mathcal{T}}$ and p_L over which the classification loss and cross-entropy losses are measured respectively. It is upper bounded by the transferability coefficient

$$\text{Lemma 5.12.10 and Lemma 5.12.9} \implies \boldsymbol{\alpha_2}(\boldsymbol{v}) \le \sqrt{\gamma(p_{\mathcal{T}})^{-1}} \text{ for all } \boldsymbol{v} \in \mathbb{R}^V$$

- **Error covariance → cross-entropy loss (arbitrary language models):** This is arguably the most important step that connects the error in prediction to the cross-entropy loss. For the arbitrary language model case, this is proved using Pinsker's inequality and taking expectation over the distribution p_L.

$$\text{Lemma 5.12.3} \implies \boldsymbol{\alpha_3}(\boldsymbol{v}) \le \sqrt{2\|\boldsymbol{v}\|_\infty^2(\ell_{\text{xent}}(\{\boldsymbol{p}_{\cdot|s}\}) - \ell_{\text{xent}}(\boldsymbol{p}^*_{\cdot|s}))} \text{ for all } \boldsymbol{v} \in \mathbb{R}^V$$

Proof sketch for softmax language models: Theorem 5.4.3

Here we want to show guarantees for features $\Phi p_f = \{\Phi \boldsymbol{p}_{f(s)}\}$ on a (τ, B)-natural task $\mathcal{T}$ w.r.t Φ. From the definition of natural tasks w.r.t. Φ, we know

$$\exists \boldsymbol{v}^* = \Phi^\top \lambda \in \mathbb{R}^V, \|\boldsymbol{v}^*\|_\infty \le B \text{ s.t. } \ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\}, \boldsymbol{v}^*) \le \tau \tag{5.24}$$

Note that the difference here is that $\boldsymbol{v}^*$ is in the span of Φ rather than an arbitrary vector in $\mathbb{R}^V$. We wish to upper bound the classification error $\ell_{\mathcal{T}}(\{\Phi \boldsymbol{p}_{f(s)}\})$ and do so using the following sequence of inequalities.

$$\begin{aligned}
\ell_{\mathcal{T}}(\{\Phi \boldsymbol{p}_{f(s)}\}) - \tau &= \inf_{\lambda \in \mathbb{R}^d} \ell_{\mathcal{T}}(\{\Phi \boldsymbol{p}_{f(s)}\}, \lambda) - \tau \\
&= \inf_{\boldsymbol{v} = \Phi^\top \lambda \in \mathbb{R}^V} \ell_{\mathcal{T}}(\{\boldsymbol{p}_{f(s)}\}, \boldsymbol{v}) - \tau \\
&\leq \ell_{\mathcal{T}}(\{\boldsymbol{p}_{f(s)}\}, \boldsymbol{v}^*) - \ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\}, \boldsymbol{v}^*) \\
&\leq \boldsymbol{\alpha_1}(\boldsymbol{v}^*) \cdot \boldsymbol{\alpha_2}(\boldsymbol{v}^*) \cdot \boldsymbol{\alpha_3}(\boldsymbol{v}^*) \qquad (5.25)
\end{aligned}$$

where the first inequality follows because $\boldsymbol{v}^*$ is in the span of Φ and second inequality follows from Equation (5.23). The bounds for $\boldsymbol{\alpha_1}(\boldsymbol{v}^*)$ and $\boldsymbol{\alpha_2}(\boldsymbol{v}^*)$ are the same as arbitrary language models. The main difference is the bound on $\boldsymbol{\alpha_3}(\boldsymbol{v}^*)$ which will be a stronger bound for softmax models.

- **Error covariance $\rightarrow$ cross-entropy loss (softmax language models):** For softmax language models, we need to prove a modified version of Pinsker's inequality specifically for softmax models. This version will show a bound that only works when $\boldsymbol{v}^*$ is in the span of Φ and if the evaluated model $\boldsymbol{p}_{f(s)}$ computes softmax using Φ as well.

$$\text{Lemma 5.12.4} \implies \boldsymbol{\alpha_3}(\boldsymbol{v}) \leq \sqrt{2\|\boldsymbol{v}\|_\infty^2 (\ell_{\text{xent}}(\{\boldsymbol{p}_{f(s)}\}) - \inf_{f^*}(\{\boldsymbol{p}_{f^*(s)}\}))} \quad \forall \boldsymbol{v} = \Phi^\top \lambda \in \mathbb{R}^V$$

Thus we suffer the suboptimality of the language model $\{\boldsymbol{p}_{f(s)}\}$ w.r.t. the best softmax model $\{\boldsymbol{p}_{f^*(s)}\}$ rather than the absolute best language model $\{\boldsymbol{p}^*_{\cdot|s}\}$. This is done using the softmax variant of Pinsker's inequality in Lemma 5.12.4. We now present the detailed proofs for all results.

5.12.2 Proofs for arbitrary language models

Theorem 5.9.2 (Strengthened Theorem 5.4.1)**.** *Let $\{\boldsymbol{p}_{\cdot|s}\}$ be a language model that is ϵ-optimal, i.e. $\ell_{xent}(\{\boldsymbol{p}_{\cdot|s}\}) - \ell^*_{xent} \leq \epsilon$ for some $\epsilon > 0$. For a classification task $\mathcal{T}$ that is (τ, B)-natural, we have*

$$\ell_{\mathcal{T}}\left(\{\boldsymbol{p}_{\cdot|s}\}\right) \leq \tau + \sqrt{\frac{2B^2\epsilon}{\gamma(p_{\mathcal{T}}; \{\boldsymbol{p}_{\cdot|s}\})}}$$

For a classification task $\mathcal{T}$ that is (τ, B)-natural w.r.t. Φ, we have

$$\ell_{\mathcal{T}}\left(\{\boldsymbol{p}_{\cdot|s}\}\right) \leq \ell_{\mathcal{T}}\left(\{\Phi \boldsymbol{p}_{\cdot|s}\}\right) \leq \tau + \sqrt{\frac{2B^2\epsilon}{\gamma_\Phi(p_{\mathcal{T}}; \{\boldsymbol{p}_{\cdot|s}\})}}$$

Proof. The proof has two main steps that we summarize by the following two lemmas. The first one upper bounds the downstream performance on natural tasks with the covariance of errors.

Lemma 5.12.2. *For a language model $\{\boldsymbol{p}_{\cdot|s}\}$, if $\mathcal{T}$ is (τ, B)-natural,*

$$\ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\}) \leq \tau + \sup_{\boldsymbol{v}\in\mathbb{R}^V, \|\boldsymbol{v}\|_\infty \leq B} \sqrt{\frac{\boldsymbol{v}^\top \Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v}}{\gamma(p_{\mathcal{T}}; \{\boldsymbol{p}_{\cdot|s}\})}}$$

If $\mathcal{T}$ is (τ, B)-natural w.r.t. $\Phi \in \mathbb{R}^{d\times V}$,

$$\ell_{\mathcal{T}}(\{\Phi\boldsymbol{p}_{\cdot|s}\}) \leq \tau + \sup_{\substack{\boldsymbol{v}=\Phi^\top\lambda\in\mathbb{R}^V, \\ \|\boldsymbol{v}\|_\infty \leq B}} \sqrt{\frac{\boldsymbol{v}^\top \Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v}}{\gamma_\Phi(p_{\mathcal{T}}; \{\boldsymbol{p}_{\cdot|s}\})}}$$

where $\gamma(\cdot)$ and $\gamma_\Phi(\cdot)$ are from Definition 5.9.1.

The second lemma upper bounds the covariance of error with the suboptimality of the language model.

Lemma 5.12.6. *For a language model $\{\boldsymbol{p}_{\cdot|s}\}$ and classifier $\boldsymbol{v} \in \mathbb{R}^V$,*

$$\boldsymbol{v}^\top \Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v} \leq 2\|\boldsymbol{v}\|_\infty^2 \left(\ell_{xent}(\{\boldsymbol{p}_{\cdot|s}\}) - \ell^*_{xent}\right)$$

where $\Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}}) = \mathop{\mathbb{E}}_{s\sim p_L}\left[(\boldsymbol{p}_{\cdot|s} - \boldsymbol{p}^*_{\cdot|s})(\boldsymbol{p}_{\cdot|s} - \boldsymbol{p}^*_{\cdot|s})^\top\right]$ as defined in Section 5.9.

We prove both the above lemmas in Section 5.12.6. We first use these to prove the main result.

Combining the two lemmas, we get the following inequality

$$\begin{aligned}
\ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\}) &\leq^{(a)} \tau + \sup_{\boldsymbol{v}\in\mathbb{R}^V, \|\boldsymbol{v}\|_\infty \leq B} \sqrt{\frac{\boldsymbol{v}^\top \Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v}}{\gamma(p_{\mathcal{T}}; \{\boldsymbol{p}_{\cdot|s}\})}} \\
&\leq^{(b)} \tau + \sup_{\boldsymbol{v}\in\mathbb{R}^V, \|\boldsymbol{v}\|_\infty \leq B} \sqrt{\frac{2\|\boldsymbol{v}\|_\infty^2 \left(\ell_{\text{xent}}(\{\boldsymbol{p}_{\cdot|s}\}) - \ell^*_{\text{xent}}\right)}{\gamma(p_{\mathcal{T}}; \{\boldsymbol{p}_{\cdot|s}\})}} \\
&\leq^{(c)} \tau + \sqrt{\frac{2B^2\epsilon}{\gamma(p_{\mathcal{T}}; \{\boldsymbol{p}_{\cdot|s}\})}}
\end{aligned}$$

where (a) uses first part of Lemma 5.12.2, (b) uses Lemma 5.12.6 and (c) uses the ϵ-optimality of $\{\boldsymbol{p}_{\cdot|s}\}$. This

proves the first part of the result. The second part can also be proved similarly.

$$\begin{aligned}
\ell_{\mathcal{T}}(\{\Phi \boldsymbol{p}_{\cdot|s}\}) &\leq^{(a)} \tau + \sup_{\substack{\boldsymbol{v}=\Phi^\top \lambda \in \mathbb{R}^V, \\ \|\boldsymbol{v}\|_\infty \leq B}} \sqrt{\frac{\boldsymbol{v}^\top \Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v}}{\gamma_\Phi(p_{\mathcal{T}}; \{\boldsymbol{p}_{\cdot|s}\})}} \\
&\leq^{(b)} \tau + \sup_{\substack{\boldsymbol{v}=\Phi^\top \lambda \in \mathbb{R}^V, \\ \|\boldsymbol{v}\|_\infty \leq B}} \sqrt{\frac{2\|\boldsymbol{v}\|_\infty^2 \left(\ell_{\text{xent}}(\{\boldsymbol{p}_{\cdot|s}\}) - \ell^*_{\text{xent}}\right)}{\gamma_\Phi(p_{\mathcal{T}}; \{\boldsymbol{p}_{\cdot|s}\})}} \\
&\leq \tau + \sup_{\boldsymbol{v} \in \mathbb{R}^V, \|\boldsymbol{v}\|_\infty \leq B} \sqrt{\frac{2\|\boldsymbol{v}\|_\infty^2 \left(\ell_{\text{xent}}(\{\boldsymbol{p}_{\cdot|s}\}) - \ell^*_{\text{xent}}\right)}{\gamma_\Phi(p_{\mathcal{T}}; \{\boldsymbol{p}_{\cdot|s}\})}} \leq^{(c)} \tau + \sqrt{\frac{2B^2\epsilon}{\gamma_\Phi(p_{\mathcal{T}}; \{\boldsymbol{p}_{\cdot|s}\})}}
\end{aligned}$$

where (a) uses second part of Lemma 5.12.2, (b) uses Lemma 5.12.6 and (c) uses the ϵ-optimality of $\{\boldsymbol{p}_{\cdot|s}\}$. The proof of the lemmas can be found in Section 5.12.6. □

Theorem 5.4.1. *Let $\{\boldsymbol{p}_{\cdot|s}\}$ be a language model that is ϵ-optimal, i.e. $\ell_{xent}(\{\boldsymbol{p}_{\cdot|s}\}) - \ell^*_{xent} \leq \epsilon$, for some $\epsilon > 0$. For a classification task $\mathcal{T}$ that is (τ, B)-natural, we have*

$$\ell_{\mathcal{T}}\left(\{\boldsymbol{p}_{\cdot|s}\}\right) \leq \tau + \sqrt{\frac{2B^2\epsilon}{\gamma(p_{\mathcal{T}})}}$$

Proof. This follows from the first part of Theorem 5.9.2 if we can also show that $\gamma(p_{\mathcal{T}}; \{\boldsymbol{p}_{\cdot|s}\})^{-1} \leq \gamma(p_{\mathcal{T}})^{-1}$. For that we use the following lemma that we prove in Section 5.12.6.

Lemma 5.12.9. *For any $g : \mathcal{S} \to \mathbb{R}^D$ and $p_{\mathcal{T}} \in \Delta_{\mathcal{S}}$, we have $\|\Sigma_{p_L}(g)^{-\frac{1}{2}} \Sigma_{p_{\mathcal{T}}}(g) \Sigma_{p_L}(g)^{-\frac{1}{2}}\|_2 \leq \gamma(p_{\mathcal{T}})^{-1}$*

Instantiating this for $g = \Delta_{\{\boldsymbol{p}_{\cdot|s}\}}$ and using Equation (5.7), we get $\gamma(p_{\mathcal{T}}; \{\boldsymbol{p}_{\cdot|s}\})^{-1} \leq \gamma(p_{\mathcal{T}})^{-1}$, which completes the proof. □

5.12.3 Proofs for softmax language models

Theorem 5.5.1 (Strengthened Theorem 5.4.3)**.** *For a fixed Φ, let f be features from an ϵ-optimal d-dimensional softmax language model, i.e. $\ell_{xent}(f, \Phi) - \ell^*_{xent}(\Phi) \leq \epsilon$, where $\ell^*_{xent}(\Phi)$ is defined in Equation* (5.4)*. For a classification task $\mathcal{T}$ that is (τ, B)-natural w.r.t. Φ, we have*

$$\ell_{\mathcal{T}}\left(\{\boldsymbol{p}_{f(s)}\}\right) \leq \ell_{\mathcal{T}}(\Phi p_f) \leq \tau + \sqrt{\frac{2B^2\epsilon}{\gamma(p_{\mathcal{T}}; \Phi p_f)}}$$

Proof. Instantiating Lemma 5.12.2 for $\boldsymbol{p}_{\cdot|s} = \boldsymbol{p}_{f(s)}$, we get

$$\begin{aligned}
\ell_{\mathcal{T}}(\{\Phi \boldsymbol{p}_{f(s)}\}) &\leq \tau + \sup_{\substack{\boldsymbol{v}=\Phi^\top \lambda \in \mathbb{R}^V, \\ \|\boldsymbol{v}\|_\infty \leq B}} \sqrt{\frac{\boldsymbol{v}^\top \Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{f(s)}\}})\boldsymbol{v}}{\gamma_\Phi(p_{\mathcal{T}}; \{\boldsymbol{p}_{f(s)}\})}} \\
&=^{(a)} \tau + \sqrt{\frac{\sup_{\|\Phi^\top \lambda\|_\infty \leq B} \lambda^\top \Phi \Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{f(s)}\}})\Phi^\top \lambda}{\gamma(p_{\mathcal{T}}; \Phi p_f)}} \\
&= \tau + \sqrt{\frac{\sup_{\|\Phi^\top \lambda\|_\infty \leq B} \lambda^\top \Sigma_{p_L}(\Phi \Delta_{\{\boldsymbol{p}_{f(s)}\}})\lambda}{\gamma(p_{\mathcal{T}}; \Phi p_f)}}
\end{aligned}$$

where (a) follows from Equation (5.9) that says $\gamma(p_{\mathcal{T}}; \Phi p_f) = \gamma_\Phi(p_{\mathcal{T}}; \{\boldsymbol{p}_{f(s)}\})$. We now prove a similar result for the second term in the following lemma that we prove in Section 5.12.6.

Lemma 5.12.7. *For a fixed Φ and a softmax language model with features f and $\lambda \in \mathbb{R}^d$,*

$$\lambda^\top \Sigma_{p_L}(\Phi \Delta_{\{\boldsymbol{p}_{f(s)}\}})\lambda \leq 2\|\Phi^\top \lambda\|_\infty^2 \left(\ell_{xent}(f, \Phi) - \ell^*_{xent}(\Phi)\right)$$

where $\Sigma_{p_L}(\Phi \Delta_{\{\boldsymbol{p}_{f(s)}\}}) = \mathop{\mathbb{E}}_{s \sim p_L}\left[(\Phi \boldsymbol{p}_{f(s)} - \Phi \boldsymbol{p}^*_{\cdot|s})(\Phi \boldsymbol{p}_{f(s)} - \Phi \boldsymbol{p}^*_{\cdot|s})^\top\right]$ as defined in Section 5.9.

Using Lemma 5.12.7 directly gives us $\ell_{\mathcal{T}}(\Phi p_f) = \ell_{\mathcal{T}}(\{\Phi \boldsymbol{p}_{f(s)}\}) \leq \tau + \sqrt{\frac{B^2(\ell_{\text{xent}}(f,\Phi) - \ell^*_{\text{xent}}(\Phi))}{\gamma_\Phi(p_{\mathcal{T}}; \Phi p_f)}}$, and the ϵ-optimality almost completes the proof. The only thing remaining to show is that $\ell_{\mathcal{T}}(\{\boldsymbol{p}_{f(s)}\}) \leq \ell_{\mathcal{T}}(\Phi p_f)$ which follows from the following sequence.

$$\begin{aligned}
\ell_{\mathcal{T}}(\{\boldsymbol{p}_{f(s)}\}) &= \inf_{\boldsymbol{v} \in \mathbb{R}^V, b \in \mathbb{R}} \ell_{\mathcal{T}}(\{\boldsymbol{p}_{f(s)}\}, \boldsymbol{v}) \leq \inf_{\Phi^\top \lambda \in \mathbb{R}^V, b \in \mathbb{R}} \ell_{\mathcal{T}}(\{\boldsymbol{p}_{f(s)}\}, (\Phi^\top \lambda, b)) \\
&= \inf_{\lambda \in \mathbb{R}^d, b \in \mathbb{R}} \ell_{\mathcal{T}}(\{\Phi \boldsymbol{p}_{f(s)}\}, (\lambda, b)) = \ell_{\mathcal{T}}(\Phi p_f)
\end{aligned}$$

□

Theorem 5.4.3. *For a fixed Φ, let f be features from an ϵ-optimal d-dimensional softmax language model, i.e. $\ell_{xent}(f, \Phi) - \ell^*_{xent}(\Phi) \leq \epsilon$, where $\ell^*_{xent}(\Phi)$ is defined in Equation (5.4). For a classification task $\mathcal{T}$ that is (τ, B)-natural w.r.t. Φ, we have*

$$\ell_{\mathcal{T}}\left(\{\boldsymbol{p}_{f(s)}\}\right) \leq \ell_{\mathcal{T}}(\Phi p_f) \leq \tau + \sqrt{\frac{2B^2\epsilon}{\gamma(p_{\mathcal{T}})}}$$

Proof. This result follows directly from Theorem 5.5.1, if we can also show that $\gamma(p_{\mathcal{T}}; \Phi p_f)^{-1} \leq \gamma(p_{\mathcal{T}})^{-1}$ just like in the proof of Theorem 5.4.1. For that we again use Lemma 5.12.9 with $g = \Phi \Delta_{\{\boldsymbol{p}_{f(s)}\}}$ and Equation (5.9) and this completes the proof. □

5.12.4 Proofs for Section 5.4.3

We first show why Assumption 5.4.4 is approximately true when word embeddings are gaussian like.

Lemma 5.12.1. *Suppose word embeddings ϕ_w are independent samples from the distribution $\mathcal{N}(\mu, \Sigma)$. Then for any $\theta \in \mathbb{R}^d$ such that $\lambda^2 = \theta^\top \Sigma \theta = O(1)$ we have that $|\log(Z_\theta) - \frac{1}{2}\theta^\top \Sigma \theta - \theta^\top \mu - \log(V)| \leq \epsilon$ with probability $1 - \delta$ for $\epsilon = \tilde{O}\left(\frac{e^{\lambda^2}}{\sqrt{V}}\right)$ and $\delta = 1 - \exp(-\Omega(\log^2(V)))$.*

Proof. We first note that $\log(Z_\theta) = \log\left(\sum_w e^{\theta^\top \phi_w}\right) = \theta^\top \mu + \log\left(\sum_w e^{\theta^\top(\phi_w - \mu)}\right)$, thus we can simply deal with the case where ϕ_w are sampled from $\mathcal{N}(0, \Sigma)$. Furthermore the only random variable of interest is $X_w = \theta^\top \phi_w$ which is a gaussian variable $\mathcal{N}(0, \theta^\top \Sigma \theta) = \mathcal{N}(0, \lambda^2)$. Thus the problem reduces to showing that for V samples of $X_w \sim \mathcal{N}(0, \lambda^2)$, $\log(Z)$ is concentrated around $\lambda^2 + \log(V)$ where $Z = \sum_w \exp(X_w)$. This can be proved similarly to the proof of Lemma 2.1 in Arora et al. [2016]. It is easy to see that $\mathop{\mathbb{E}}_{X_w \sim \mathcal{N}(0,\lambda^2)}[\exp(X_w)] = e^{\lambda^2}$. However the variable $\exp(X_w)$ is neither sub-gaussian nor sub-exponential and thus standard inequalities cannot be used directly. We use the same technique as Arora et al. [2016] to first observe that $\mathbb{E}[Z] = Ve^{\frac{1}{2}\lambda^2}$ and $\mathrm{Var}[Z] \leq \mathbb{E}[\exp(2X_w)] = Ve^{2\lambda^2}$. After conditioning on the event that $X_w \leq \frac{1}{2}\lambda \log(V)$ and applying Berstein's inequality just like in Arora et al. [2016] completes the proof. □

We next prove Lemma 5.4.5 that establishes a linear relationship between Φp_f and f (under Assumption 5.4.4) and also the guarantees for f on natural tasks.

Lemma 5.4.5. *Under Assumption 5.4.4, any feature map $f : \mathcal{S} \to \mathbb{R}^d$ satisfies $\Phi p_f(s) = \boldsymbol{A} f(s) + \boldsymbol{b}$, for all $s \in \mathcal{S}$.*

Proof. Assumption 5.4.4 gives us that $\log(Z_\theta) = \frac{1}{2}\theta^\top \boldsymbol{A}\theta + \theta^\top \boldsymbol{b} + c$. We prove this lemma by matching the gradients of $\log(Z_\theta)$ and the quadratic function on the R.H.S.

$$\nabla_\theta \log(Z_\theta) = \frac{\nabla_\theta Z_\theta}{Z_\theta} = \frac{\sum_{w \in \mathcal{W}} e^{\phi_w^\top \theta} \phi_w}{Z_\theta} = \sum_{w \in \mathcal{W}} p_\theta(w) \phi_w = \Phi p_\theta$$

Whereas the gradient of the quadratic part is $\nabla_\theta[\frac{1}{2}\theta^\top \boldsymbol{A}\theta + \theta^\top \boldsymbol{b} + c] = \boldsymbol{A}\theta + \boldsymbol{b}$. Matching the two for $\theta = f(s)$ gives us $\Phi p_f(s) = \Phi \boldsymbol{p}_{f(s)} = \boldsymbol{A}f(s) + \boldsymbol{b}$. □

Corollary 5.4.6. *Using Lemma 5.4.5, for any ϵ-optimal f, as defined in Theorem 5.4.3, for classification tasks that are (τ, B)-natural w.r.t. Φ we have $\ell_{\mathcal{T}}(f) \leq \tau + \mathcal{O}(\sqrt{\epsilon})$.*

Proof. The main idea is that Lemma 5.4.5 gives us that $\Phi p_f(s) = \boldsymbol{A}f(s) + \boldsymbol{b}$ and thus any linear function of Φp_f will also be a linear function of $f(s)$. From Theorem 5.5.1 (or Theorem 5.4.3), we also know that Φp_f will do well on $\mathcal{T}$, i.e. $\ell_{\mathcal{T}}(\Phi p_f) \leq \tau + \mathcal{O}(B\sqrt{\epsilon})$. We formalize[7] the intuition as

$$\begin{aligned}\ell_{\mathcal{T}}(\Phi p_f) &= \inf_{\lambda \in \mathbb{R}^d, b} \ell_{\mathcal{T}}(\Phi p_f, (\lambda, b)) = \inf_{\lambda \in \mathbb{R}^d, b} \ell_{\mathcal{T}}(\boldsymbol{A}f + \boldsymbol{b}, (\lambda, b)) = \inf_{\lambda \in \mathbb{R}^d, b} \ell_{\mathcal{T}}(f, (\boldsymbol{A}^\top \lambda, b + \lambda^\top \boldsymbol{b})) \\ &\geq \inf_{\boldsymbol{v} \in \mathbb{R}^d, b'} \ell_{\mathcal{T}}(f, (\boldsymbol{v}, b')) = \ell_{\mathcal{T}}(f)\end{aligned}$$

This shows that $\ell_{\mathcal{T}}(f) \leq \ell_{\mathcal{T}}(\Phi p_f) \leq \tau + \mathcal{O}(B\sqrt{\epsilon})$ and completes the proof. □

5.12.5 Proofs for Section 5.10

Theorem 5.10.1. *The optimal solution $f^*, \Phi^* = \arg\min_{f,\Phi} \ell_{quad}(f, \Phi)$ satisfies*

$$\begin{aligned}\Phi^* &= \boldsymbol{B}\boldsymbol{U}_d^\top, \text{ for full rank } \boldsymbol{B} \in \mathbb{R}^{d \times d} \\ f^*(s) &= (\Phi^* {\Phi^*}^\top)^{-1/2} \Phi^* \boldsymbol{p}^*_{\cdot|s} = \boldsymbol{C}\boldsymbol{U}_d^\top \boldsymbol{p}^*_{\cdot|s}, \text{ for full rank } \boldsymbol{C} \in \mathbb{R}^{d \times d}\end{aligned}$$

If Φ is fixed, then the optimal solution is $f^*(s) = (\Phi\Phi^\top)^{-1/2}\Phi \boldsymbol{p}^*_{\cdot|s}$.

Proof. From Equations (5.10) and (5.11) we know that, $\ell_{quad,s}(\theta, \Phi) = -\theta^\top \Phi \boldsymbol{p}^*_{\cdot|s} + \frac{1}{2}\|\Phi^\top \theta\|^2$ and $\ell_{quad}(f, \Phi) = \mathop{\mathbb{E}}_{s \sim p_L}[\ell_{quad,s}(f(s), \Phi)]$. For a fixed Φ, we define $f^*_\Phi(s) = \arg\min_{\theta \in \mathbb{R}^d} \ell_{quad,s}(\theta, \Phi)$.

We use the first-order optimality condition to get $f^*_\Phi(s)$, by using the fact that $\nabla_\theta \ell_{quad,s}(\theta, \Phi) = -\Phi \boldsymbol{p}^*_{\cdot|s} + \Phi\Phi^\top \theta$. Setting the gradient to zero, we get $f^*_\Phi(s) = (\Phi\Phi^\top)^{-1}\Phi \boldsymbol{p}^*_{\cdot|s}$[8]. To get the optimal Φ^* for this objective, we

[7] Note that here we assume that we learn both a linear classifier and an intercept for a downstream classification task. All results in the paper essentially remain the same with an intercept in the definition of classification loss.

[8] It will be clear later that the optimal solution will have as high a rank as possible Φ. All inverses can be replaced by pseudo-inverses for low-rank matrices.

plug in this expression for f^*_Φ in ℓ_{quad} and find $\Phi^* = \arg\min_\Phi \ell_{quad}(f^*_\Phi, \Phi)$.

$$\begin{aligned}
\ell_{quad}(f^*_\Phi, \Phi) &= \mathop{\mathbb{E}}_{s\sim p^*}\left[\ell_{quad,s}(f^*_\Phi(s), \Phi)\right] = \mathop{\mathbb{E}}_{s\sim p^*}\left[-f^*_\Phi(s)^\top \Phi \boldsymbol{p}^*_{\cdot|s} + \frac{1}{2}\|\Phi^\top f^*_\Phi(s)\|^2\right] \\
&= \mathop{\mathbb{E}}_{s\sim p^*}\left[-((\Phi\Phi^\top)^{-1}\Phi\boldsymbol{p}^*_{\cdot|s})^\top \Phi \boldsymbol{p}^*_{\cdot|s} + \frac{1}{2}\|\Phi^\top(\Phi\Phi^\top)^{-1}\Phi\boldsymbol{p}^*_{\cdot|s}\|^2\right] \\
&= \mathop{\mathbb{E}}_{s\sim p^*}\left[-\boldsymbol{p}^*_{\cdot|s}{}^\top \Phi^\top(\Phi\Phi^\top)^{-1}\Phi\boldsymbol{p}^*_{\cdot|s} + \frac{1}{2}\boldsymbol{p}^*_{\cdot|s}{}^\top \Phi^\top(\Phi\Phi^\top)^{-1}\Phi\Phi^\top(\Phi\Phi^\top)^{-1}\Phi\boldsymbol{p}^*_{\cdot|s}\right] \\
&= \mathop{\mathbb{E}}_{s\sim p^*}\left[-\frac{1}{2}\boldsymbol{p}^*_{\cdot|s}{}^\top \Phi^\top(\Phi\Phi^\top)^{-1}\Phi\boldsymbol{p}^*_{\cdot|s}\right] = -\frac{1}{2}\mathop{\mathbb{E}}_{s\sim p^*}\left[\mathrm{tr}\left(\boldsymbol{p}^*_{\cdot|s}{}^\top \Phi^\top(\Phi\Phi^\top)^{-1}\Phi\boldsymbol{p}^*_{\cdot|s}\right)\right] \\
&= -\frac{1}{2}\mathrm{tr}\left(\Phi^\top(\Phi\Phi^\top)^{-1}\Phi \mathop{\mathbb{E}}_{s\sim p^*}\left[\boldsymbol{p}^*_{\cdot|s}\boldsymbol{p}^*_{\cdot|s}{}^\top\right]\right) \\
&= -\frac{1}{2}\left\langle \Phi^\top(\Phi\Phi^\top)^{-1}\Phi, \mathop{\mathbb{E}}_{s\sim p^*}\left[\boldsymbol{p}^*_{\cdot|s}\boldsymbol{p}^*_{\cdot|s}{}^\top\right]\right\rangle = -\frac{1}{2}\left\langle \Phi^\top(\Phi\Phi^\top)^{-1}\Phi, \Omega^*\right\rangle
\end{aligned}$$

where Ω^* is the substitutability matrix defined in Definition 5.5.2. Let $\Phi = \boldsymbol{N}\boldsymbol{T}\boldsymbol{V}^\top$ be the SVD. Then the above objective reduces to $\ell_{quad}(f^*_\Phi, \Phi) = -\frac{1}{2}\left\langle \boldsymbol{V}\boldsymbol{V}^\top, \Omega^*\right\rangle$ And hence learning the optimal Φ^* reduces to learning an optimal $\boldsymbol{V}^*$ such that

$$\boldsymbol{V}^* = \mathop{\arg\min}_{\boldsymbol{V}\in\mathbb{R}^{V\times d}, \boldsymbol{V}^\top\boldsymbol{V}=I_d} -\langle \boldsymbol{V}\boldsymbol{V}^\top, \Omega^*\rangle$$

We will now show that the best such matrix is the matrix of top d eigenvectors of Ω^*, i.e. $\boldsymbol{V}^* = \boldsymbol{U}_d$ (cf. Definition 5.5.2). Here we will assume that the eigenvalues of Ω^* are all distinct for simplicity of presentation. First we note that $\langle \boldsymbol{V}\boldsymbol{V}^\top, \Omega^*\rangle = \|\boldsymbol{V}\boldsymbol{V}^\top \Omega^{*\frac{1}{2}}\|_F^2$, where $\Omega^{*\frac{1}{2}} = \boldsymbol{U}\boldsymbol{S}^{\frac{1}{2}}\boldsymbol{U}^\top$, with $\boldsymbol{U}$, $\boldsymbol{U}_d$ and $\boldsymbol{S}$ define in Definition 5.5.2. This can be shown by the following sequence of steps

$$\begin{aligned}
\langle \boldsymbol{V}\boldsymbol{V}^\top, \Omega^*\rangle &= \mathrm{tr}(\boldsymbol{V}\boldsymbol{V}^\top\Omega^*) = \mathrm{tr}(\boldsymbol{V}\boldsymbol{V}^\top\boldsymbol{V}\boldsymbol{V}^\top\Omega^*) = \mathrm{tr}(\boldsymbol{V}\boldsymbol{V}^\top\Omega^*\boldsymbol{V}\boldsymbol{V}^\top) \\
&= \mathrm{tr}(\boldsymbol{V}\boldsymbol{V}^\top\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^\top\boldsymbol{V}\boldsymbol{V}^\top) = \mathrm{tr}(\boldsymbol{V}\boldsymbol{V}^\top\boldsymbol{U}\boldsymbol{S}^{\frac{1}{2}}\boldsymbol{U}^\top\boldsymbol{U}\boldsymbol{S}^{\frac{1}{2}}\boldsymbol{U}^\top\boldsymbol{V}\boldsymbol{V}^\top) \\
&= \mathrm{tr}(\boldsymbol{V}\boldsymbol{V}^\top\Omega^{*\frac{1}{2}}\Omega^{*\frac{1}{2}}\boldsymbol{V}\boldsymbol{V}^\top) = \langle \boldsymbol{V}\boldsymbol{V}^\top\Omega^{*\frac{1}{2}}, \boldsymbol{V}\boldsymbol{V}^\top\Omega^{*\frac{1}{2}}\rangle \\
&= \|\boldsymbol{V}\boldsymbol{V}^\top\Omega^{*\frac{1}{2}}\|_F^2
\end{aligned}$$

Furthermore, we notice that $\|\boldsymbol{V}\boldsymbol{V}^\top\Omega^{*\frac{1}{2}}\|_F^2 = \|\Omega^{*\frac{1}{2}}\|_F^2 - \|\Omega^{*\frac{1}{2}} - \boldsymbol{V}\boldsymbol{V}^\top\Omega^{*\frac{1}{2}}\|_F^2$ as shown below

$$\|\Omega^{*\frac{1}{2}} - \boldsymbol{V}\boldsymbol{V}^\top\Omega^{*\frac{1}{2}}\|_F^2 = \|\Omega^{*\frac{1}{2}}\|_F^2 + \|\boldsymbol{V}\boldsymbol{V}^\top\Omega^{*\frac{1}{2}}\|_F^2 - 2\mathrm{tr}(\Omega^{*\frac{1}{2}}\boldsymbol{V}\boldsymbol{V}^\top\Omega^{*\frac{1}{2}})$$

$$
\begin{aligned}
&= \|\Omega^{*\frac{1}{2}}\|_F^2 + \|\boldsymbol{V}\boldsymbol{V}^\top\Omega^{*\frac{1}{2}}\|_F^2 - 2\mathrm{tr}(\Omega^{*\frac{1}{2}}\boldsymbol{V}\boldsymbol{V}^\top\boldsymbol{V}\boldsymbol{V}^\top\Omega^{*\frac{1}{2}}) \\
&= \|\Omega^{*\frac{1}{2}}\|_F^2 + \|\boldsymbol{V}\boldsymbol{V}^\top\Omega^{*\frac{1}{2}}\|_F^2 - 2\|\boldsymbol{V}\boldsymbol{V}^\top\Omega^{*\frac{1}{2}}\|_F^2 \\
&= \|\Omega^{*\frac{1}{2}}\|_F^2 - \|\boldsymbol{V}\boldsymbol{V}^\top\Omega^{*\frac{1}{2}}\|_F^2
\end{aligned}
$$

Thus we get $\underset{\boldsymbol{V}\in\mathbb{R}^{V\times d},\boldsymbol{V}^\top\boldsymbol{V}=I_d}{\arg\min} -\langle\boldsymbol{V}\boldsymbol{V}^\top,\Omega^*\rangle = \underset{\boldsymbol{V}\in\mathbb{R}^{V\times d},\boldsymbol{V}^\top\boldsymbol{V}=I_d}{\arg\min} \|\Omega^{*\frac{1}{2}} - \boldsymbol{V}\boldsymbol{V}^\top\Omega^{*\frac{1}{2}}\|_F^2$.

Note that $\boldsymbol{V}\boldsymbol{V}^\top\Omega^{*\frac{1}{2}}$ has columns that are columns of $\Omega^{*\frac{1}{2}}$ projected on the space spanned by columns $\boldsymbol{V}$. It is folklore that the best such subspace $\boldsymbol{V}^*$ is the subspace spanned by the top d eigenvectors of $\Omega^{*\frac{1}{2}}$, which is the same as top d eigenvectors of Ω^*, thus giving us $\boldsymbol{V}^*\boldsymbol{V}^{*\top} = \boldsymbol{U}_d\boldsymbol{U}_d^\top$. Thus we get $\boldsymbol{V}^* = \boldsymbol{U}_d\boldsymbol{M}$ for $\boldsymbol{M} = \boldsymbol{U}_d^\top\boldsymbol{V}^*$.

This tells us that the optimal solution Φ^* will have SVD of the form $\Phi^* = \boldsymbol{N}^*\boldsymbol{T}^*\boldsymbol{V}^{*\top}$, thus giving us $\Phi^* = \boldsymbol{B}\boldsymbol{U}_d^\top$ for matrix $\boldsymbol{B} = \boldsymbol{N}^*\boldsymbol{T}^*\boldsymbol{M}^\top \in \mathbb{R}^{d\times d}$. This directly gives $f^* = f^*_{\Phi^*} = (\Phi^*\Phi^{*\top})^{-1}\Phi^*\boldsymbol{p}^*_{\cdot|s} = \boldsymbol{N}^*\boldsymbol{T}^{-1}\boldsymbol{V}^{*\top}\boldsymbol{p}^*_{\cdot|s} = \boldsymbol{C}\boldsymbol{U}_d^\top\boldsymbol{p}^*_{\cdot|s}$ for $\boldsymbol{C} = \boldsymbol{N}^*\boldsymbol{T}^{*-1}\boldsymbol{M}^\top$.

□

5.12.6 Proofs for supporting lemmas

Lemma 5.12.2. *For a language model $\{\boldsymbol{p}_{\cdot|s}\}$, if $\mathcal{T}$ is (τ, B)-natural,*

$$
\ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\}) \le \tau + \sup_{\boldsymbol{v}\in\mathbb{R}^V, \|\boldsymbol{v}\|_\infty \le B} \sqrt{\frac{\boldsymbol{v}^\top\Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v}}{\gamma(p_{\mathcal{T}};\{\boldsymbol{p}_{\cdot|s}\})}}
$$

If $\mathcal{T}$ is (τ, B)-natural w.r.t. $\Phi \in \mathbb{R}^{d\times V}$,

$$
\ell_{\mathcal{T}}(\{\Phi\boldsymbol{p}_{\cdot|s}\}) \le \tau + \sup_{\substack{\boldsymbol{v}=\Phi^\top\lambda\in\mathbb{R}^V, \\ \|\boldsymbol{v}\|_\infty\le B}} \sqrt{\frac{\boldsymbol{v}^\top\Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v}}{\gamma_\Phi(p_{\mathcal{T}};\{\boldsymbol{p}_{\cdot|s}\})}}
$$

where $\gamma(\cdot)$ and $\gamma_\Phi(\cdot)$ are from Definition 5.9.1.

Proof. We note the following upper bounds on $\ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\})$ and $\ell_{\mathcal{T}}(\{\Phi\boldsymbol{p}_{\cdot|s}\})$.

$$\ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\}) = \inf_{\boldsymbol{v}\in\mathbb{R}^V} \left\{\ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\},\boldsymbol{v})\right\} \leq \inf_{\substack{\boldsymbol{v}\in\mathbb{R}^V,\\ \|\boldsymbol{v}\|_\infty\leq B}} \left\{\ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\},\boldsymbol{v})\right\} \tag{5.26}$$

$$\ell_{\mathcal{T}}(\{\Phi\boldsymbol{p}_{\cdot|s}\}) = \inf_{\boldsymbol{v}=\Phi^\top\lambda\in\mathbb{R}^V} \left\{\ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\},\boldsymbol{v})\right\} \leq \inf_{\substack{\boldsymbol{v}=\Phi^\top\lambda\in\mathbb{R}^V, b\in\mathbb{R},\\ \|\boldsymbol{v}\|_\infty\leq B}} \left\{\ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\},\boldsymbol{v})\right\} \tag{5.27}$$

When $\mathcal{T}$ is (τ, B)-natural, by Definition 5.3.1 we know that $\inf_{\substack{\boldsymbol{v}\in\mathbb{R}^V\\ \|\boldsymbol{v}\|_\infty\leq B}} \left[\ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\},\boldsymbol{v})\right] \leq \tau$. We now upper bound $\ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\},\boldsymbol{v})$ using Lemma 5.12.8. Taking infimum w.r.t. $\boldsymbol{v}\in\mathbb{R}^V, \|\boldsymbol{v}\|_\infty\leq B$ from the inequality in Lemma 5.12.8.

$$\ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\},\boldsymbol{v}) \leq \ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\},\boldsymbol{v}) + \sqrt{\boldsymbol{v}^\top\Sigma_{p_{\mathcal{T}}}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v}}$$

$$\inf_{\substack{\boldsymbol{v}\in\mathbb{R}^V\\ \|\boldsymbol{v}\|_\infty\leq B}} \ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\},\boldsymbol{v}) \leq \inf_{\substack{\boldsymbol{v}\in\mathbb{R}^V\\ \|\boldsymbol{v}\|_\infty\leq B}} \ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\},\boldsymbol{v}) + \sup_{\boldsymbol{v}\in\mathbb{R}^V,\|\boldsymbol{v}\|_\infty\leq B} \sqrt{\boldsymbol{v}^\top\Sigma_{p_{\mathcal{T}}}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v}}$$

This, combined with Equation (5.26), gives us

$$\ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\}) \leq \tau + \sup_{\boldsymbol{v}\in\mathbb{R}^V,\|\boldsymbol{v}\|_\infty\leq B} \sqrt{\boldsymbol{v}^\top\Sigma_{p_{\mathcal{T}}}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v}} \tag{5.28}$$

Using Lemma 5.12.10 and the definition of $\gamma(p_{\mathcal{T}};\{\boldsymbol{p}_{\cdot|s}\})$ in Equation (5.7), we get that

$$\begin{aligned}\boldsymbol{v}^\top\Sigma_{p_{\mathcal{T}}}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v} &\leq \left\|\Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})^{-\frac{1}{2}}\Sigma_{p_{\mathcal{T}}}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})^{-\frac{1}{2}}\right\|_2 \left(\boldsymbol{v}^\top\Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v}\right)\\ &= \frac{\boldsymbol{v}^\top\Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v}}{\gamma(p_{\mathcal{T}};\{\boldsymbol{p}_{\cdot|s}\})}\end{aligned} \tag{5.29}$$

We have thus successfully transferred the bound from the distribution $p_{\mathcal{T}}$ to p_L. Combining this with Equation (5.28) completes the proof of the first part of the lemma.

We now prove the second part of the lemma where we only assume that $\mathcal{T}$ is (τ, B)-natural w.r.t. Φ. Here we instead take the infimum over classifiers in the span of Φ in Lemma 5.12.8 to get

$$\inf_{\substack{\boldsymbol{v}=\Phi^\top\lambda\in\mathbb{R}^V, b\in\mathbb{R},\\ \|\boldsymbol{v}\|_\infty\leq B}} \left\{\ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\},\boldsymbol{v})\right\} \leq \inf_{\substack{\boldsymbol{v}=\Phi^\top\lambda\in\mathbb{R}^V, b\in\mathbb{R},\\ \|\boldsymbol{v}\|_\infty\leq B}} \left\{\ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\},\boldsymbol{v})\right\} +$$

$$\sup_{\substack{\boldsymbol{v}=\Phi^\top\lambda\in\mathbb{R}^V,\\ \|\boldsymbol{v}\|_\infty\le B}} \sqrt{\boldsymbol{v}^\top\Sigma_{p_\mathcal{T}}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v}} \tag{5.30}$$

This, combined with definition of (τ, B)-natural task w.r.t. Φ and Equation (5.27) gives us

$$\ell_\mathcal{T}(\{\Phi\boldsymbol{p}_{\cdot|s}\}) \le \tau + \sup_{\substack{\boldsymbol{v}=\Phi^\top\lambda\in\mathbb{R}^V,\\ \|\boldsymbol{v}\|_\infty\le B}} \sqrt{\boldsymbol{v}^\top\Sigma_{p_\mathcal{T}}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v}} \tag{5.31}$$

For the last term, for any $\boldsymbol{v} = \Phi^\top\lambda, \lambda \in \mathbb{R}^d$ we notice that

$$\begin{aligned}
\boldsymbol{v}^\top\Sigma_{p_\mathcal{T}}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v} &= \lambda^\top\Phi\Sigma_{p_\mathcal{T}}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\Phi^\top\lambda = \lambda^\top\Sigma_{p_\mathcal{T}}(\Phi\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\lambda \\
&\le^{(a)} \left\|\Sigma_{p_L}(\Phi\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})^{-\frac{1}{2}}\Sigma_{p_\mathcal{T}}(\Phi\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\Sigma_{p_L}(\Phi\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})^{-\frac{1}{2}}\right\|_2 \left(\lambda^\top\Sigma_{p_L}(\Phi\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\lambda\right) \\
&= \frac{\lambda^\top\Sigma_{p_L}(\Phi\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\lambda}{\gamma_\Phi(p_\mathcal{T};\{\boldsymbol{p}_{\cdot|s}\})} = \frac{\boldsymbol{v}^\top\Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v}}{\gamma_\Phi(p_\mathcal{T};\{\boldsymbol{p}_{\cdot|s}\})}
\end{aligned}$$

This combined with Equation (5.31), we get

$$\ell_\mathcal{T}(\{\Phi\boldsymbol{p}_{\cdot|s}\}) \le \tau + \inf_{\substack{\boldsymbol{v}=\Phi^\top\lambda\in\mathbb{R}^V,\\ \|\boldsymbol{v}\|_\infty\le B}} \sqrt{\frac{\boldsymbol{v}^\top\Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v}}{\gamma_\Phi(p_\mathcal{T};\{\boldsymbol{p}_{\cdot|s}\})}}$$

□

Lemma 5.12.3 (Pinsker's inequality)**.** *For discrete distributions $q, q^* \in \Delta_V$, let $\boldsymbol{q}, \boldsymbol{q^*} \in \mathbb{R}^V$ be the corresponding vector of probabilities. Then we have*

$$\max_{\|\boldsymbol{v}\|_\infty\le 1} |\boldsymbol{v}^\top(\boldsymbol{q} - \boldsymbol{q^*})| \le \sqrt{2D_{\mathrm{KL}}(q^*, q)}$$

Proof. This basically follows from Pinsker's inequality which upper bounds the total variation distance between distributions by their KL-divergence

$$\max_{\|\boldsymbol{v}\|_\infty\le 1} |\boldsymbol{v}^\top(\boldsymbol{q} - \boldsymbol{q^*})| = \|\boldsymbol{q} - \boldsymbol{q^*}\|_1 = 2\ \mathrm{TV}(q^*, q) \le \sqrt{2D_{\mathrm{KL}}(q^*, q)}$$

□

We remind the reader that for an embedding matrix $\Phi \in \mathbb{R}^{d\times V}$, $p_{\theta,\Phi} \coloneqq \mathrm{softmax}(\Phi^\top\theta)$

Lemma 5.12.4 (Softmax variant of Pinsker's inequality)**.** *Consider a matrix $\Phi \in \mathbb{R}^{d\times V}$ with $d \le V$. For any discrete distribution $q^* \in \Delta_V$ and softmax distribution $p_{\theta,\Phi} = \mathit{softmax}(\Phi^\top\theta) \in \Delta_V$ for $\theta \in \mathbb{R}^d$, let $\boldsymbol{q^*}, \boldsymbol{p}_{\theta,\Phi} \in \mathbb{R}^V$ be the corresponding vector of probabilities. Then we have*

$$\max_{\substack{\boldsymbol{v}=\Phi^\top\lambda,\\ \|\boldsymbol{v}\|_\infty\le 1}} |\boldsymbol{v}^\top(\boldsymbol{p}_{\theta,\Phi} - \boldsymbol{q^*})| \le \sqrt{2\left(D_{\mathrm{KL}}(p_{\theta,\Phi}, q^*) - \inf_{\theta^*\in\mathbb{R}^d} D_{\mathrm{KL}}(p_{\theta^*,\Phi}, q^*)\right)} \tag{5.32}$$

Pinsker's inequality (Lemma 5.12.3), on the other hand, gives

$$\max_{\|\boldsymbol{v}\|_\infty\le 1} |\boldsymbol{v}^\top(\boldsymbol{p}_{\theta,\Phi} - \boldsymbol{q^*})| \le \sqrt{2D_{\mathrm{KL}}(p_{\theta,\Phi}, q^*)}$$

Proof. Define the loss $\rho(\theta) \coloneqq D_{\mathrm{KL}}(p_{\theta,\Phi}, q^*)$. The statement in Equation (5.32) to prove reduces to

$$\max_{\|\Phi^\top\lambda\|_\infty\le 1} |\lambda^\top(\Phi\boldsymbol{p}_{\theta,\Phi} - \Phi\boldsymbol{q^*})| \le \sqrt{2\left(\rho(\theta) - \inf_{\theta^*\in\mathbb{R}^d}\rho(\theta^*)\right)} \tag{5.33}$$

To prove this, we compute the gradient and hessian of $\rho(\theta)$ w.r.t. θ. We can simplify $\rho(\theta)$ as follows

$$\begin{aligned}\rho(\theta) = D_{\mathrm{KL}}(p_{\theta,\Phi}, q^*) &= \mathop{\mathbb{E}}_{w\sim q^*}[-\log(p_{\theta,\Phi}(w))] = \mathop{\mathbb{E}}_{w\sim q^*}\left[-\log\left(\frac{e^{\theta^\top\phi_w}}{\sum_{w'} e^{\theta^\top\phi_{w'}}}\right)\right]\\ &= -\theta^\top\Phi\boldsymbol{q}^* + \log\left(\sum_{w'} e^{\theta^\top\phi_{w'}}\right) = -\theta^\top\Phi\boldsymbol{q}^* + \log\left(Z_\theta\right)\end{aligned}$$

The gradient is

$$\begin{aligned}\nabla\rho(\theta) &= \nabla\left[-\theta^\top\Phi q^* + \log(Z_\theta)\right] = -\Phi\boldsymbol{q}^* + \frac{\nabla Z_\theta}{Z_\theta}\\ &= -\Phi\boldsymbol{q}^* + \frac{\nabla\sum_w e^{\theta^\top\phi_w}}{Z_\theta} = -\Phi\boldsymbol{q}^* + \frac{\sum_w e^{\theta^\top\phi_w}\phi_w}{Z_\theta}\\ &= -\Phi\boldsymbol{q}^* + \Phi p_{\theta,\Phi}\end{aligned}$$

Similarly the Hessian can be computed

$$\begin{aligned}\nabla^2\rho(\theta) = \nabla(\nabla\rho(\theta)) = \nabla[-\Phi\boldsymbol{q}^* + \Phi p_{\theta,\Phi}] &= \nabla\sum_{w\in\mathcal{W}} p_{\theta,\Phi}(w)\phi_w = \sum_{w\in\mathcal{W}}\nabla\frac{e^{\theta^\top\phi_w}}{Z_\theta}\phi_w\\ = \sum_{w\in\mathcal{W}}\frac{e^{\theta^\top\phi_w}}{Z_\theta}\phi_w\phi_w^\top - \frac{e^{\theta^\top\phi_w}}{Z_\theta^2}\phi_w&\left(\sum_{w'} e^{\theta^\top\phi_{w'}}\phi_{w'}\right)^\top\end{aligned}$$

$$= \mathop{\mathbb{E}}_{w\sim p_{\theta,\Phi}}[\phi_w\phi_w^\top] - \left(\mathop{\mathbb{E}}_{w\sim p_{\theta,\Phi}}[\phi_w]\right)\left(\mathop{\mathbb{E}}_{w\sim p_{\theta,\Phi}}[\phi_w]\right)^\top = \mathrm{Cov}_{w\sim p_{\theta,\Phi}}[\phi_w]$$

Where $\mathrm{Cov}_{w\sim p_{\theta,\Phi}}[\phi_w]$ denotes the covariance of the word embeddings ϕ_w when measured w.r.t. the distribution $p_{\theta,\Phi}$. This directly gives us that $\nabla^2\rho(\theta) \succcurlyeq 0$, since the covariance is always psd, and thus ρ is convex in θ.

We return to the statement in Equation (5.33) that we need to prove. With the expression for gradient of ρ at hand, we can rewrite Equation (5.33) as trying to prove

$$|\lambda^\top\nabla\rho(\theta)| \le \|\Phi^\top\lambda\|_\infty\sqrt{2\left(\rho(\theta) - \inf_{\theta^*\in\mathbb{R}^d}\rho(\theta^*)\right)} \tag{5.34}$$

Furthermore, using the definition of the Hessian, it is not hard to see for some $\lambda, \tilde{\theta} \in \mathbb{R}^d$ that $\lambda^\top\nabla^2\rho(\tilde{\theta})\lambda = \mathrm{Cov}_{w\sim p_{\tilde{\theta},\Phi}}[\lambda^\top\phi_w] \le \mathop{\mathbb{E}}_{w\sim p_{\tilde{\theta},\Phi}}[(\lambda^\top\phi_w)^2] \le \|\Phi^\top\lambda\|_\infty^2$. Thus we can evoke Lemma 5.12.5 with $\ell = \rho$ and $L = \|\Phi^\top\lambda\|_\infty^2$ to prove Equation (5.34) and thus completing the proof. Intuitively Lemma 5.12.5 exploits the smoothness of the function to argue that small suboptimality (i.e. being close to optimal solution in function value) is sufficient to guarantee small norm of the gradient, a property that is well-known in the optimization literature. We now present this lemma □

Lemma 5.12.5. *If a function $\ell : \mathbb{R}^d \to \mathbb{R}$ and $\lambda \in \mathbb{R}^d$ satisfy $\lambda^\top\nabla^2\ell(\tilde{\theta})\lambda \le L, \forall\tilde{\theta} \in \mathbb{R}^d$ (L-smoothness in the direction of λ) and if $\ell^* = \inf_{\theta\in\mathbb{R}^d}\ell(\theta)$, then $|\lambda^\top\nabla\ell(\theta)|^2 \le 2L(\ell(\theta) - \ell^*)$*

Proof. This is a variant of a classical result used in optimization and we prove it here for completeness. For any $\eta \in \mathbb{R}$ we have

$$\begin{aligned}
\ell(\theta) - \ell^* &\ge^{(a)} \ell(\theta) - \ell(\theta - \eta\lambda) \\
&\ge^{(b)} \ell(\theta) - \left(\ell(\theta) + \langle\nabla\ell(\theta), -\eta\lambda\rangle + \frac{\eta^2}{2}\lambda^\top\nabla^2\ell(\tilde{\theta})\lambda\right) \\
&\ge^{(c)} \eta(\lambda^\top\nabla\ell(\theta)) - \frac{\eta^2 L}{2}
\end{aligned}$$

where (a) follows from the definition of infimum and (b) follows from Taylor's expansion for some $\tilde{\theta} \in [\theta - \eta\lambda, \theta]$ and (c) follows from the smoothness condition in the statement of the lemma. Picking $\eta = \frac{\lambda^\top\nabla\ell(\theta)}{L}$ gives us $\ell(\theta) - \ell^* \ge \frac{1}{2L}|\lambda^\top\nabla\ell(\theta)|^2$, thus completing the proof. □

Lemma 5.12.6. *For a language model $\{\boldsymbol{p}_{\cdot|s}\}$ and classifier $\boldsymbol{v} \in \mathbb{R}^V$,*

$$\boldsymbol{v}^\top \Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v} \leq 2\|\boldsymbol{v}\|_\infty^2 \left(\ell_{xent}(\{\boldsymbol{p}_{\cdot|s}\}) - \ell^*_{xent}\right)$$

where $\Sigma_{p_L}(g) = \mathop{\mathbb{E}}_{s\sim p_L}[g(s)g(s)^\top]$ and $\Delta_{\{\boldsymbol{p}_{\cdot|s}\}}(s) = \boldsymbol{p}_{\cdot|s} - \boldsymbol{p}^*_{\cdot|s}$ are defined in Section 5.9

Proof. We first note that

$$\ell_{\text{xent}}(\{\boldsymbol{p}_{\cdot|s}\}) - \ell_{\text{xent}}(\{\boldsymbol{p}^*_{\cdot|s}\}) = \mathop{\mathbb{E}}_{s\sim p_L}\mathop{\mathbb{E}}_{w\sim p^*_{\cdot|s}}\left[\log\left(\frac{\boldsymbol{p}^*_{\cdot|s}(w)}{\boldsymbol{p}_{\cdot|s}(w)}\right)\right] = \mathop{\mathbb{E}}_{s\sim p_L}\left[D_{\text{KL}}(\boldsymbol{p}^*_{\cdot|s}, \boldsymbol{p}_{\cdot|s})\right] \tag{5.35}$$

We bound $\boldsymbol{v}^\top \Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v}$ below

$$\begin{aligned}
\boldsymbol{v}^\top \Sigma_{p_L}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v} &= \mathop{\mathbb{E}}_{s\sim p_L}\left[\left(\boldsymbol{v}^\top(\boldsymbol{p}_{\cdot|s} - \boldsymbol{p}^*_{\cdot|s})\right)^2\right] \\
&\leq^{(a)} \|\boldsymbol{v}\|_\infty^2 \mathop{\mathbb{E}}_{s\sim p_L}\left[2D_{\text{KL}}(\boldsymbol{p}^*_{\cdot|s}, \boldsymbol{p}_{\cdot|s})\right] \\
&=^{(b)} 2\|\boldsymbol{v}\|_\infty^2 \left(\ell_{\text{xent}}(\{\boldsymbol{p}_{\cdot|s}\}) - \ell_{\text{xent}}(\{\boldsymbol{p}^*_{\cdot|s}\})\right)
\end{aligned}$$

where (a) follows from Lemma 5.12.3 (Pinsker's inequality), (b) uses Equation (5.35). □

Lemma 5.12.7. *For a fixed Φ, a softmax language model with features f and $\lambda \in \mathbb{R}^d$,*

$$\lambda^\top \Sigma_{p_L}(\Phi\Delta_{\{\boldsymbol{p}_{f(s)}\}})\lambda \leq 2\|\Phi^\top\lambda\|_\infty^2 \left(\ell_{xent}(f, \Phi) - \ell^*_{xent}(\Phi)\right)$$

where $\Sigma_{p_L}(\Phi\Delta_{\{\boldsymbol{p}_{f(s)}\}}) = \mathop{\mathbb{E}}_{s\sim p_L}\left[(\Phi\boldsymbol{p}_{f(s)} - \Phi\boldsymbol{p}^*_{\cdot|s})(\Phi\boldsymbol{p}_{f(s)} - \Phi\boldsymbol{p}^*_{\cdot|s})^\top\right]$ as defined in Section 5.9.

Proof. We start by nothing that

$$\begin{aligned}
\lambda^\top \Sigma_{p_L}(\Phi\Delta_{\{\boldsymbol{p}_{f(s)}\}})\lambda &= \lambda^\top \mathop{\mathbb{E}}_{s\sim p_L}\left[(\Phi\boldsymbol{p}_{f(s)} - \Phi\boldsymbol{p}^*_{\cdot|s})(\Phi\boldsymbol{p}_{f(s)} - \Phi\boldsymbol{p}^*_{\cdot|s})^\top\right]\lambda \\
&= \mathop{\mathbb{E}}_{s\sim p_L}[|\lambda^\top(\Phi\boldsymbol{p}_{f(s)} - \Phi\boldsymbol{p}^*_{\cdot|s})|^2] = \mathop{\mathbb{E}}_{s\sim p_L}[|(\Phi^\top\lambda)^\top(\boldsymbol{p}_{f(s)} - \boldsymbol{p}^*_{\cdot|s})|^2]
\end{aligned}$$

We will use the variant of Pinsker's inequality from Lemma 5.12.4 to bound each term on the right hand side.

Notice that $\ell_{\text{xent}}(f,\Phi) - \ell^*_{\text{xent}}(\Phi) = \mathop{\mathbb{E}}_{s\sim p_L}[\ell_{\text{xent},s}(f(s),\Phi) - \inf_{\theta\in\mathbb{R}^d} \ell_{\text{xent},s}(\theta,\Phi)]$.

$$\begin{aligned}
\lambda^\top \Sigma_{p_L}(\Phi\Delta_{\{\boldsymbol{p}_{f(s)}\}})\lambda &= \mathop{\mathbb{E}}_{s\sim p_L}[|(\Phi^\top\lambda)^\top(\boldsymbol{p}_{f(s)} - \boldsymbol{p}^*_{\cdot|s})|^2] \\
&\leq^{(a)} 2\|\Phi^\top\lambda\|^2_\infty \mathop{\mathbb{E}}_{s\sim p_L}\left[D_{\text{KL}}(\boldsymbol{p}^*_{\cdot|s}, \boldsymbol{p}_{f(s),\Phi}) - \inf_{\theta\in\mathbb{R}^d} D_{\text{KL}}(\boldsymbol{p}^*_{\cdot|s}, \boldsymbol{p}_{\theta,\Phi})\right] \\
&\leq 2\|\Phi^\top\lambda\|^2_\infty \mathop{\mathbb{E}}_{s\sim p_L}\left[\ell_{\text{xent},s}(f(s),\Phi) - \inf_{\theta\in\mathbb{R}^d} \ell_{\text{xent},s}(\theta,\Phi)\right] \\
&\leq 2\|\Phi^\top\lambda\|^2_\infty (\ell_{\text{xent}}(f,\Phi) - \ell^*_{\text{xent}}(\Phi))
\end{aligned}$$

where (a) follows from Lemma 5.12.4. This completes the proof. □

Classification loss to covariance of error

Lemma 5.12.8. *For any task $\mathcal{T}$ and classifier $\boldsymbol{v}\in\mathbb{R}^V$ and predicted probabilities $\{\boldsymbol{p}_{\cdot|s}\}$*

$$\begin{aligned}
\ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\},\boldsymbol{v}) &\leq \ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\},\boldsymbol{v}) + \sqrt{\mathop{\mathbb{E}}_{s\sim p_{\mathcal{T}}}\left[(\boldsymbol{v}^\top(\boldsymbol{p}_{\cdot|s} - \boldsymbol{p}^*_{\cdot|s}))^2\right]} \\
&= \ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\},\boldsymbol{v}) + \sqrt{\boldsymbol{v}^\top\Sigma_{p_{\mathcal{T}}}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v}}
\end{aligned}$$

where $\Sigma_{p_{\mathcal{T}}}(g) = \mathop{\mathbb{E}}_{s\sim p_{\mathcal{T}}}[g(s)g(s)^\top]$ and $\Delta_{\{\boldsymbol{p}_{\cdot|s}\}}(s) = \boldsymbol{p}_{\cdot|s} - \boldsymbol{p}^*_{\cdot|s}$ are defined in Section 5.9.

Proof. The following sequence of inequalities proves it

$$\begin{aligned}
\ell_{\mathcal{T}}(\{\boldsymbol{p}_{\cdot|s}\},\boldsymbol{v}) &= \mathop{\mathbb{E}}_{(s,y)\sim p_{\mathcal{T}}}\left[\ell(\boldsymbol{v}^\top\boldsymbol{p}_{\cdot|s}, y)\right] \leq^{(a)} \mathop{\mathbb{E}}_{(s,y)\sim p_{\mathcal{T}}}\left[\ell(\boldsymbol{v}^\top\boldsymbol{p}^*_{\cdot|s}, y) + |\boldsymbol{v}^\top(\boldsymbol{p}^*_{\cdot|s} - \boldsymbol{p}_{\cdot|s})|\right] \\
&\leq^{(b)} \mathop{\mathbb{E}}_{(s,y)\sim p_{\mathcal{T}}}\left[\ell(\boldsymbol{v}^\top\boldsymbol{p}^*_{\cdot|s}, y)\right] + \sqrt{\mathop{\mathbb{E}}_{s\sim p_{\mathcal{T}}}\left[\left|\boldsymbol{v}^\top(\boldsymbol{p}^*_{\cdot|s} - \boldsymbol{p}_{\cdot|s})\right|^2\right]} \\
&= \ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\},\boldsymbol{v}) + \sqrt{\boldsymbol{v}^\top\left(\mathop{\mathbb{E}}_{s\sim p_{\mathcal{T}}}\left[(\boldsymbol{p}^*_{\cdot|s} - \boldsymbol{p}_{\cdot|s})(\boldsymbol{p}^*_{\cdot|s} - \boldsymbol{p}_{\cdot|s})^\top\right]\right)\boldsymbol{v}} \\
&= \ell_{\mathcal{T}}(\{\boldsymbol{p}^*_{\cdot|s}\},\boldsymbol{v}) + \sqrt{\boldsymbol{v}^\top\Sigma_{p_{\mathcal{T}}}(\Delta_{\{\boldsymbol{p}_{\cdot|s}\}})\boldsymbol{v}}
\end{aligned}$$

where (a) follows from 1-lipschitzness of ℓ, (b) follows from Jensen's inequality. □

Handling distribution shift

Lemma 5.12.9. *For any $g:\mathcal{S}\to\mathbb{R}^D$ and $p_{\mathcal{T}}\in\Delta_{\mathcal{S}}$, we have $\|\Sigma_{p_L}(g)^{-\frac{1}{2}}\Sigma_{p_{\mathcal{T}}}(g)\Sigma_{p_L}(g)^{-\frac{1}{2}}\|_2 \leq \gamma(p_{\mathcal{T}})^{-1}$*

Proof. By definition of $\gamma(p_{\mathcal{T}})$, we have that

$$\Sigma_{p_L}(g) = \mathop{\mathbb{E}}_{s\sim p_L}[g(s)g(s)^\top] = \sum_{s\in\mathcal{S}} p_L(s)g(s)g(s)^\top$$
$$\succcurlyeq \gamma(p_{\mathcal{T}}) \sum_{s\in\mathcal{S}} p_{\mathcal{T}}(s)g(s)g(s)^\top = \gamma(p_{\mathcal{T}}) \mathop{\mathbb{E}}_{s\sim p_{\mathcal{T}}}[g(s)g(s)^\top] = \gamma(p_{\mathcal{T}})\Sigma_{p_{\mathcal{T}}}(g)$$

Thus $\frac{1}{\gamma(p_{\mathcal{T}})}\Sigma_{p_L}(g) \succcurlyeq \Sigma_{p_{\mathcal{T}}}(g)$ and hence $\frac{1}{\gamma(p_{\mathcal{T}})}\Sigma_{p_L}(g)^{-\frac{1}{2}}\Sigma_{p_L}(g)\Sigma_{p_L}(g)^{-\frac{1}{2}} \succcurlyeq \Sigma_{p_L}(g)^{-\frac{1}{2}}\Sigma_{p_{\mathcal{T}}}(g)\Sigma_{p_L}(g)^{-\frac{1}{2}}$, which is equivalent to $\frac{1}{\gamma(p_{\mathcal{T}})}I_D \succcurlyeq \Sigma_{p_L}(g)^{-\frac{1}{2}}\Sigma_{p_{\mathcal{T}}}(g)\Sigma_{p_L}(g)^{-\frac{1}{2}}$. This finishes the proof. □

Lemma 5.12.10. *For matrices $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{D\times D}$ s.t. $\boldsymbol{X}, \boldsymbol{Y} \succcurlyeq 0$ and $\boldsymbol{Y}$ is full rank, we have that $\max_{\mathbf{a}\in\mathbb{R}^D, 0<\|\mathbf{a}\|\leq\lambda} \frac{\mathbf{a}^\top \boldsymbol{X}\mathbf{a}}{\mathbf{a}^\top \boldsymbol{Y}\mathbf{a}} = \|\boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{Y}^{-\frac{1}{2}}\|_2$ for any norm $\|\cdot\|$.*

Proof. Note that $\frac{\mathbf{a}^\top \boldsymbol{X}\mathbf{a}}{\mathbf{a}^\top \boldsymbol{Y}\mathbf{a}}$ is independent of the scaling of $\mathbf{a}$. The following sequence of inequalities completes the proof

$$\max_{\mathbf{a}\in\mathbb{R}^D, 0<\|\mathbf{a}\|\leq\lambda} \frac{\mathbf{a}^\top \boldsymbol{X}\mathbf{a}}{\mathbf{a}^\top \boldsymbol{Y}\mathbf{a}} = \max_{\mathbf{a}\in\mathbb{R}^D} \frac{\mathbf{a}^\top \boldsymbol{X}\mathbf{a}}{\mathbf{a}^\top \boldsymbol{Y}\mathbf{a}} = \max_{\mathbf{a}\in\mathbb{R}^D} \frac{\mathbf{a}^\top \boldsymbol{X}\mathbf{a}}{(\boldsymbol{Y}^{\frac{1}{2}}\mathbf{a})^\top(\boldsymbol{Y}^{\frac{1}{2}}\mathbf{a})}$$
$$= \max_{\mathbf{a}\in\mathbb{R}^D, \|\boldsymbol{Y}^{\frac{1}{2}}\mathbf{a}\|_2=1} \mathbf{a}^\top \boldsymbol{X}\mathbf{a} = \max_{\boldsymbol{b}\in\mathbb{R}^D, \|\boldsymbol{b}\|_2=1} (\boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{b})^\top \boldsymbol{X}(\boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{b})$$
$$= \max_{\boldsymbol{b}\in\mathbb{R}^D, \|\boldsymbol{b}\|_2=1} \boldsymbol{b}^\top \boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{b} = \|\boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{Y}^{-\frac{1}{2}}\|_2$$

□

5.13 Experiment details

For all experiments[9], we use the 117M parameter "small" GPT-2 model proposed in Radford et al. [2019] and implemented in HuggingFace [Wolf et al., 2019]. Linear classification experiments (except for fine-tuning baseline in Table 5.1) are performed on *fixed* output features from GPT-2.

We note that the binary SST-2 dataset used in all experiments is comprised of complete sentences, and there are 6,920 train examples and 1,821 test examples. In particular, this dataset is smaller than the version included with the GLUE benchmark [Wang et al., 2018]. This smaller version of SST-2 better fits the sentence completion hypothesis we propose.

[9]Link to code: `https://github.com/sadhikamalladi/mathematical-exploration-downstream-tasks`.

5.13.1 Solving downstream tasks using f and Φp_f

The features f from GPT-2 for any input sequence $(w_1, \ldots, w_N)$ is the output embedding of the final token w_N at the final layer, where N is the input length and can be different for different inputs. This is also the embedding that is directly multiplied by the word embeddings to get the softmax distribution for language modeling, as in the theoretical setting. To use a prompt, the same prompt is added at the end of all inputs and the features are extracted for this modified input.

We use the `LogisticRegressionCV` class from the scikit-learn package to fit linear classifiers to all fixed features (i.e., no finetuning). We use the liblinear solver and one-vs-rest loss function unless it catastrophically fails (e.g., close to random performance) on a particular multi-class task. In that case, we use the stochastic average gradient (SAG) algorithm with multinomial loss. We use 5-fold cross validation for all experiments and test values for the regularization parameter C between 1e−6 and 1e4 for small datasets (i.e., fewer than 10K examples) and between 1e−3 and 1e3 for larger datasets.

Details about word subsets: For all of the results presented in Table Table 5.1, we use a pre-trained GPT-2 model. For SST, we use the prompt "This movie is " when indicated. For AG News, we use the prompt "This article is about " when indicated.

We compute the conditional probability of selecting a subset of words to complete the sentence. For AG News, this subset is: 'world', 'politics', 'sports', 'business', 'science', 'financial', 'market', 'foreign', 'technology', 'international', 'stock', 'company', 'tech', 'technologies'. For SST, this subset is: ':)', ':(', 'great', 'charming', 'flawed', 'classic', 'interesting', 'boring', 'sad', 'happy', 'terrible', 'fantastic', 'exciting', 'strong'. For AG News, the class words we use are: 'foreign', 'sports', 'financial', 'scientific'. For SST, the class words we use are ':)' and ':('.

We account for BPE tokenization by using the encoding of the word directly and the encoding of the word with a space prepended. We then filter to use only words that encode to a single BPE token.

Tests on additional datasets: We also test the performance of pre-trained GPT-2 embeddings f and the conditional mean embeddings Φp_f on the DBPedia [Auer et al., 2007], Yahoo Answers [Zhang et al., 2015], TREC [Li and Roth, 2002], IMDb [Maas et al., 2011], Customer Review (CR) [Hu and Liu, 2004], and MPQA polarity [Wilson and Wiebe, 2003] datasets in Table Table 5.2. We limited the training set size to 250K for larger datasets (i.e., DBPedia and Yahoo Answers). For CR and MPQA, we follow Zhang et al.

[2015] and average the performance across 10 random 90-10 train-test splits of the dataset.

We find that Φp_f consistently has comparable performance to f across non-sentiment and sentiment downstream classification tasks. We include baseline results of bag of n-grams (BonG) for most tasks and the mLSTM model [Radford et al., 2017] for sentiment tasks. BonG performs quite well on the larger datasets, but not as well on smaller datasets, due to the high dimensionality of features.

For sentiment tasks, adding a prompt almost always boosts performance. We also demonstrate that much of the performance can be recovered by only looking at "positive" and "negative" or ":)" and ":(" as class words. Using these 2-dimensional features is even more sample-efficient than the standard 768-dimensional ones.

We also include results using the pre-trained BERT base cased model [Devlin et al., 2019, Wolf et al., 2019], using the embedding at the first token as input to the downstream task. We also tried using the mean embedding and last token embedding and found that the first token embedding is often the best. Moreover, the first token embedding is what is extracted in the traditional usage of BERT on downstream tasks, though we note that it is rare to use BERT without fine-tuning.

5.13.2 Finetuning experiments

As a strong baseline, we finetune the GPT-2 features along with learning a linear classifier for the SST and AG News classification tasks and report accuracy numbers in Table 5.1. We use a maximum sequence length of 128 BPE tokens for downstream inputs of SST-2 and a maximum length of 400 BPE tokens for AG News inputs. We use the end of sentence token as the padding token. The datasets are described below.

1. AG News has 108K train examples, 12K dev examples, 7600 test examples. We split the train set for AG News into train and dev (90-10) and use the same test set as the non-finetuning experiments.

2. The sentence version of SST-2 has 6,920 train examples (same as non-finetuning), and 810 examples for dev and test each (split the original test set in half).

3. Fine-grained SST-2 has 8,544 train examples (same as non-finetuning), and 1,105 examples each for the dev and test data (split the original test set in half).

To select the best hyperparameter configuration, we run a grid search over learning rate and batch size. We

Table 5.2: GPT-2 performance without fine-tuning on downstream task test sets with k classes. We provide the performance of bag of n-grams (BonG) as an approximate baseline for these tasks. AG News, DBPedia and Yahoo performances were reported in Zhang et al. [2015], and the other tasks were reported in Khodak et al. [2018]. We also include results from mLSTM (Sentiment Neuron) [Radford et al., 2017] for the sentiment-related classification tasks (SST, IMDb, CR, and MPQA) with numbers reported from Khodak et al. [2018]. Furthermore, we include results for BERT [Devlin et al., 2019] features without fine-tuning, where we use the output features for the first position of an input for linear classification. An asterisk indicates we add a standard sentiment prompt "The sentiment is" to each input, but for AG News we used the prompt "This article is about". We also tested the performance of the conditional probability distribution over "positive" and "negative" as well as ":)" and ":(" on the sentiment-related tasks with and without the prompt.

Task	k	$f(s)$	$\Phi p_f(s)$	$\boldsymbol{p}_{\cdot\|s}$: pos,neg	$\boldsymbol{p}_{\cdot\|s}$: :),:(	BonG	mLSTM	BERT
Non-sentiment								
AG News	4	90.7	84.6	-	-	92.4 ($n = 5$)	-	88.9
AG News*	4	91.1	88.2	-	-	-	-	89.9
DBPedia	14	97.2	88.2	-	-	98.6 ($n = 5$)	-	98.7
Yahoo	10	69.2	56.7	-	-	68.5 ($n = 5$)	-	65.0
TREC	6	93.6	87.8	-	-	89.8 ($n = 3$)	-	90.6
Sentiment								
SST	2	87.5	83.3	74.9	78.7	80.9 ($n = 2$)	91.8	85.8
SST*	2	89.4	87.3	80.8	79.1	-	-	84.1
SST fine	5	49.2	43.5	37.5	39.2	42.3 ($n = 3$)	52.9	43.5
SST fine*	5	49.4	48.0	41.5	40.2	-	-	43.3
IMDb	2	88.1	82.7	73.8	76.2	89.8 ($n = 3$)	92.3	82.2
IMDb*	-	88.4	85.3	81.8	80.9	-	-	84.0
CR	2	86.8	84.6	74.9	80.0	78.3 ($n = 3$)	91.4	85.5
CR*	-	87.9	87.1	82.5	79.4	-	-	84.6
MPQA	2	86.0	79.2	75.6	70.7	85.6 ($n = 3$)	88.5	87.3
MPQA*	-	87.8	86.1	80.3	71.4	-	-	88.1

train each model for 10 epochs. For all datasets, we test learning rates 5e−5, 1e−4, and 3e−4. For both version of SST-2, we try batch sizes 8, 16, and 32, and for AG News, we try batch sizes 8, 12, and 16. We note that the longer sequence length of AG News inputs required us to use parallelization across multiple GPUs to simulate larger batch sizes, which made batch size 32 prohibitively expensive to test.

We take the hyperparameter configuration that achieves the best performance on the dev set and then perform fine-tuning using those settings with three different random seeds: 8, 33, and 42. We then report the average performance on the test set in Table 5.1.

We perform the hyperparameter grid search over the standard datasets and then perform fine-tuning using the best settings on the dataset with task-specific prompts added. For SST-2, we use the prompt "This movie is ", and for AG News we use "This article is about ".

Table 5.3: Comparing Quad features to cross-entropy features for GPT-2 trained on the IMDb unlabeled corpus [Maas et al., 2011]. In this experiment we fix Φ to be the word embeddings from prertained GPT-2 model for the cross-entropy objective. For the Quad objective, we initialize Φ to be the SVD of the pre-trained embeddings. An asterisk indicates that we added the prompt "This movie is " to each input.

Task	$f(s)$ (xent)	$\Phi p_f(s)$ (xent)	$f(s)$ (Quad)
SST	82.1%	79.9%	77.3%
SST*	83.1%	81.1%	80.7%

Table 5.4: Comparing Quad features to cross-entropy features for GPT-2 trained on the Amazon corpus. An asterisk indicates that we added the prompt "This movie is " to each input. Note that the validation loss was still decreasing at the time of measurement.

Task	$f(s)$ (xent)	$\Phi p_f(s)$ (xent)	$f(s)$ (Quad, learned Φ)
SST	89.4%	89.7%	79.2%
SST*	89.7%	89.2%	84.3%

5.13.3 Testing Quad objective

We test two models with the same parametrizations, one trained using our Quad objective and another trained with the standard cross-entropy objective using the unlabeled IMDb corpus [Maas et al., 2011] and the Amazon product review corpus [McAuley et al., 2015]. We slightly modify the standard architecture of GPT-2 to generate Tables Table 5.3 and Table 5.4. First we add a single linear layer (that is trained) on top of the output features of the standard Transformer architecture. Furthermore, instead of tying the input and output word (token) embeddings, we learn them separately so that f and Φ are independent functions; this is more in line with out theoretical setup. We fix the input embeddings and the positional embeddings to be the parameters from the pre-trained GPT-2.

For Quad, we initialize Φ, the output embeddings, using the singular vectors of the pre-trained word embeddings Φ. For the cross-entropy models, we initialize Φ to be the full pre-trained word embeddings Φ, because we found that initializing with the singular vectors harmed performance. Given our parameterization, initializing with the singular vectors is as expressive as initializing with the pretrained embeddings Φ themselves; however it potentially lends a better optimization landscape and speeds up training for our new objective Quad. As described in Section 5.5.2, we minimize the following objective

$$\ell_{quad}(f, \Phi) = \mathop{\mathbb{E}}_{(s,w)} \left[-f(s)^\top \phi_w + \frac{1}{2} \|\Phi^\top f(s)\|^2 \right] \tag{5.36}$$

where (s, w) are sampled from the text corpus. The implementation of the Quad loss is the same as the

standard cross-entropy loss, the main difference being the second term: it is $\frac{1}{2}\|\Phi^\top f(s)\|^2$ for Quad instead of the log-partition function $\log\left(\sum_{w'} e^{f(s)^\top \phi_{w'}}\right)$ in the cross-entropy objective.

Because IMDb is a smaller dataset, we fix Φ at its initialization and only train f to generate Table Table 5.3. When training on the Amazon dataset, we initialized Φ the same way as we did for the IMDb dataset, but we allowed f and Φ to both be trained, since more data was available. To train the models, we use the standard learning rate schedule as in in Radford et al. [2019]. To learn a model on IMDb, we use a context size of 512 BPE tokens, and for the Amazon reviews dataset [McAuley et al., 2015], we use the standard context length of 1,024 BPE tokens.

We observe that training using Quad, in both cases, yields comparable performance to the language model on the SST task, but always slightly worse. According to the theory, features $f(s)$ from Quad should learn $\boldsymbol{p}^*_{\cdot|s}$ on a subspace, just like Φp_f from cross-entropy models, thus making the comparison between these two important. Furthermore, adding a prompt consistently improves performance for both objectives. While Quad did not beat the cross-entropy in either case, its good performs at least demonstrates that insights from the theoretical analysis can translate to practical algorithms. We leave exploring the gap in performance between Quad and cross-entropy and a more extensive evaluation of Quad for future work.

5.13.4 Learning the quadratic approximation of the log-partition function

In Assumption 5.4.4, we assert that there is a quadratic fit for the log partition function, which allows us to show in Lemma 5.4.5 that a linear relation holds between f and Φp_f. We validate these theoretical findings by fitting a quadratic function to the log partition function for a subset of embeddings from the IMDb, SST, and AG News datasets (Figure 5.1). Here, we describe how we learned $\boldsymbol{A}$, $\boldsymbol{b}$ and c. To ensure $\boldsymbol{A}$ is symmetric and positive semi-definite as required, we parametrize $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{U}^T$. As defined earlier, the partition function $Z_\theta = \sum_{w'} e^{\theta^\top \phi_{w'}}$ and $\Phi p_\theta = \sum_{w'} \frac{e^{\theta^\top \phi_{w'}}}{Z_\theta}\phi_{w'}$ for any $\theta \in \mathbb{R}^d$. We minimize the following objective function:

$$\mathcal{L}(\boldsymbol{U}, \boldsymbol{b}, c) = \mathop{\mathbb{E}}_{\theta}\left[\lambda_1 \left(\log(Z_\theta) - \frac{1}{2}\theta^\top \boldsymbol{U}\boldsymbol{U}^\top \theta - \theta^\top \boldsymbol{b} - c\right)^2 + \lambda_2 \left\|\Phi p_\theta - \boldsymbol{U}\boldsymbol{U}^\top \theta - \boldsymbol{b}\right\|^2\right] \tag{5.37}$$

In practice, we train only on the regression loss (i.e., $\lambda_1 = 0$, $\lambda_2 = 1$) for the most promising results. Note

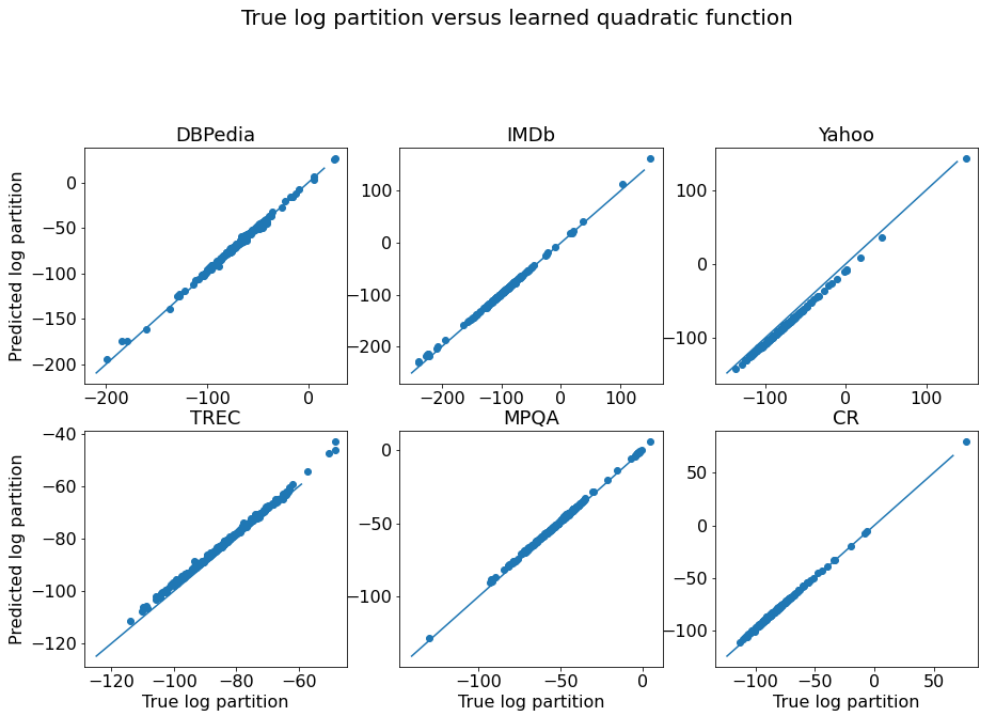
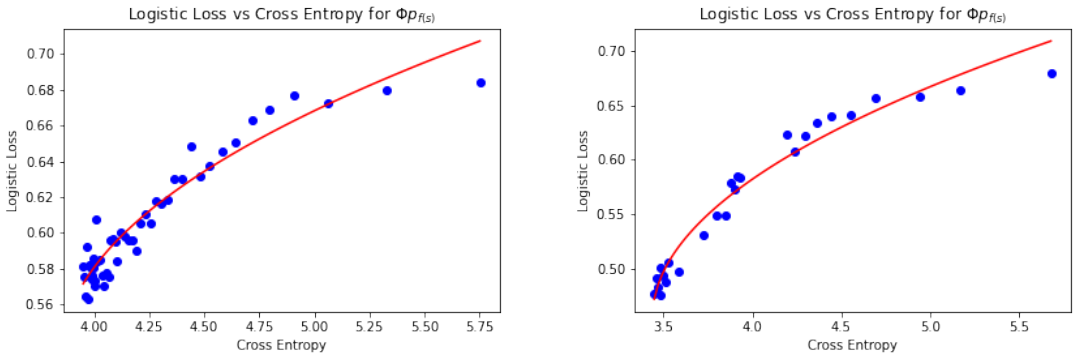


Figure 5.2: Fit of the learned quadratic function to the log partition function on various datasets for features computed by the full, pre-trained GPT-2. We also plot the $y = x$ line for reference. These plots are meant to verify Assumption 5.4.4.



(a) Trained on IMDb [Maas et al., 2011]

(b) Trained on Amazon [McAuley et al., 2015]

Figure 5.3: Logistic loss of conditional mean features on the SST-2 task for various checkpoints of a GPT-2 architecture trained on IMDb and Amazon. The reported cross-entropy is measured on the validation set. The red trend shows the fit of a square-root function, which is what the upper bound in Theorem 5.4.3 looks like.

that the regression term is trying to learn a linear relationship between between θ and Φp_θ that Lemma 5.4.5 aims to prove. This ends up learning a matrix $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{U}^\top$ and vector $\boldsymbol{b}$ that also satisfy the quadratic form

of $\log(Z_\theta)$ from Assumption 5.4.4.

We use 20,000 examples from a mix of IMDb, SST, and AG News embeddings as the training set. Thus we sample θ by sampling s from the aforementioned datasets and set $\theta = f(s)$, f being the feature map from pretrained GPT-2. We use the Adam [Kingma and Ba, 2015] optimizer with learning rate 1e$-$3 for $\boldsymbol{U}$ and learning rate 1e$-$4 for $\boldsymbol{b}$ and c. We decay the learning rate every 50 steps by a factor of 0.1. We use the $\boldsymbol{U}$ obtained after 8 epochs of training. We further demonstrate the quality of the learned fit by plotting the true log partition and estimated log partition function for embeddings from other datasets in Figure 5.2.

5.13.5 Experimentally checking Theorem 5.4.3

Theorem 5.4.3 can be informally summarized as stating that an ϵ suboptimality in the cross-entropy of a d-dimensional language model propagates to a $\sqrt{\epsilon}$ increase in the logistic loss. We note that the τ, B, and $\gamma(p_{\mathcal{T}})$ factors are fixed for a given pre-training corpus and downstream task, so we can empirically test if this square root relationship holds in practice. In particular, Theorem 5.4.3 says

$$\ell_{\mathcal{T}}(\Phi p_f) \leq \tau + \sqrt{2B^2 \left(\gamma(p_{\mathcal{T}})\right)^{-1} \left(\ell_{\text{xent}}(f, \Phi) - \ell^*_{\text{xent}}\right)} \tag{5.38}$$

Of these, $\tau, B, \gamma(p_{\mathcal{T}})^{-1}$ and ℓ^*_{xent} are independent of the language model (f, Φ) and only depend on the task $\mathcal{T}$ and language modeling distribution. Thus we can rewrite this as $\ell_{\mathcal{T}}(\Phi p_f) \leq c + a\sqrt{\ell_{\text{xent}}(f, \Phi) - b}$ for suitable constants $a, b, c \in \mathbb{R}$. The left hand side, $\ell_{\mathcal{T}}(\Phi p_f)$, is the logistic loss of conditional mean features from language model (f, Φ) on task $\mathcal{T}$ and $\ell_{\text{xent}}(f, \Phi)$ is the cross-entropy loss of the language model, both of which can be measured in practice.

We train a 117M parameter GPT-2 model from scratch on the IMDb and Amazon corpora, described in Section 5.13.3. We maintain checkpoints during training, and for each checkpoint, we measure the cross-entropy of the model on the validation set as well as the performance of the conditional mean features Φp_f on SST-2. Plotting these values together yields Figure 5.3.

We furthermore fit a square root trend, shown in red, to these points. We learn a, b, c such that $y \approx a\sqrt{x - b} + c$, where $y = \ell_{\mathcal{T}}(\Phi p_f)$ is the logistic loss and $x = \ell_{\text{xent}}(f, \Phi)$ is the cross-entropy loss. For this, we perform a grid search over 100 evenly spaced valid values of b, and for each b, we perform linear regression on $\sqrt{x - b}$ to find a and c. We choose the a, b, c that maximizes the r-value of the regression. While Theorem 5.4.3 only provides an upper bound on the logistic loss, this experiment shows that some square-root trend is observable

in practice.

Bibliography

Samira Abnar, Mostafa Dehghani, Behnam Neyshabur, and Hanie Sedghi. Exploring the limits of large scale pre-training. In *International Conference on Learning Representations*, 2022.

Woo-Kyoung Ahn and William F Brewer. Psychological studies of explanation—based learning. *Investigating explanation-based learning*, 1993.

Guillaume Alain and Yoshua Bengio. What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, 2014.

Rie Kubota Ando and Tong Zhang. Two-view feature generation model for semi-supervised learning. In *International conference on Machine learning*, 2007.

Sanjeev Arora and Andrej Risteski. Provable benefits of representation learning. arXiv, 2017.

Sanjeev Arora, Rong Ge, and Ankur Moitra. Learning topic models–going beyond svd. In *2012 IEEE 53rd annual symposium on foundations of computer science*. IEEE, 2012.

Sanjeev Arora, Rong Ge, Yonatan Halpern, David Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. A practical algorithm for topic modeling with provable guarantees. In *International conference on machine learning*. PMLR, 2013.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 2016.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*, 2017.

Sanjeev Arora, Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. A compressed sensing view of

unsupervised text embeddings, bag-of-n-grams, and LSTMs. In *International Conference on Learning Representations*, 2018.

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. In *International Conference on Machine Learning*, 2019.

Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Dipendra Misra. Investigating the role of negatives in contrastive representation learning. In *International Conference on Artificial Intelligence and Statistics*, 2022.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, 2007.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Neural Information Processing Systems*, 2019.

Charles R Baker. Joint measures and cross-covariance operators. *Transactions of the American Mathematical Society*, 1973.

Yamini Bansal, Gal Kaplun, and Boaz Barak. For self-supervised learning, rationality implies generalization, provably. In *International Conference on Learning Representations*, 2021.

Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 1993.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 2002.

Aurélien Bellet, Amaury Habrard, and Marc Sebban. Similarity learning for provably accurate sparse linear classification. In *International Conference on Machine Learning*, 2012.

Aurélien Bellet, Amaury Habrard, and Marc Sebban. A survey on metric learning for feature vectors and structured data. *CoRR*, abs/1306.6709, 2013.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 2003.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 2013.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 2003.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Conference on Computational Learning Theory*, 1998.

George E. P. Box. Science and statistics. *Journal of the American Statistical Association*, 1976. doi: 10.1080/01621459.1976.10480949.

Leo Breiman and Jerome H Friedman. Estimating optimal transformations for multiple regression and correlation. *Journal of the American statistical Association*, 1985.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Neural Information Processing Systems*, 2020.

Andreas Buja. Remarks on functional canonical variates, alternating least squares methods and ace. *The Annals of Statistics*, 1990.

Stanley F Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 1999.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 2020a.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. In *Neural Information Processing Systems*, 2020b.

Ting Chen, Calvin Luo, and Lala Li. Intriguing properties of contrastive losses. *Neural Information Processing Systems*, 2021.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Gated feedback recurrent neural networks. In *International Conference on Machine Learning*, 2015.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Two-stage learning kernel algorithms. In *International Conference on Machine Learning*, 2010.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *IEEE International Conference on Computer Vision*, 2015.

Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Neural Information Processing Systems*, 2014.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Simon S Du, Wei Hu, Sham M Kakade, Jason D Lee, and Qi Lei. Few-shot learning via learning the representation, provably. In *International Conference on Learning Representations*, 2020.

Chris Dyer. Notes on noise contrastive estimation and negative sampling. *arXiv preprint arXiv:1410.8251*, 2014.

Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *IEEE conference on computer vision and pattern recognition*, 2017.

John R Firth. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, 1957.

Kenji Fukumizu, Francis R Bach, and Michael I Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research*, 2004.

Kenji Fukumizu, Francis R Bach, Michael I Jordan, et al. Kernel dimension reduction in regression. *The Annals of Statistics*, 2009.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing*, 2021.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

Shiry Ginosar, Kate Rakelly, Sarah Sachs, Brian Yin, and Alexei A Efros. A century of portraits: A visual historical record of american high school yearbooks. In *IEEE International Conference on Computer Vision Workshops*, 2015.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. DeCLUTR: Deep contrastive learning for unsupervised textual representations. In *Association for Computational Linguistics*, 2021.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International Conference on Algorithmic Learning Theory*. Springer, 2005.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Neural Information Processing Systems*, 2020.

David Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 2011.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *International Conference on Artificial Intelligence and Statistics*, 2010.

Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. In *Neural Information Processing Systems*, 2021.

David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

Zellig Harris. Distributional structure. *Word*, 1954.

Elad Hazan and Tengyu Ma. A non-generative framework and convex relaxations for unsupervised learning. In *Neural Information Processing Systems*, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019.

Thomas Hofmann. Probabilistic latent semantic indexing. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Association for Computational Linguistics*, 2018.

Daniel Hsu, Sham M Kakade, and Tong Zhang. Random design analysis of ridge regression. In *Conference on Learning Theory*, 2012.

Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020.

Tzee-Ming Huang. Testing conditional independence using maximal nonlinear conditional correlation. *The Annals of Statistics*, 2010.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International Conference on Machine Learning*, 2017.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*, 2022.

Sham M Kakade and Dean P Foster. Multi-view regression via canonical correlation analysis. In *International Conference on Computational Learning Theory*, 2007.

Mikhail Khodak, Nikunj Saunshi, Yingyu Liang, Tengyu Ma, Brandon Stewart, and Sanjeev Arora. A la carte embedding: Cheap but effective induction of semantic feature vectors. In *Association for Computational Linguistics*, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Neural Information Processing Systems*, 2015.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Abhishek Kumar, Alexandru Niculescu-Mizil, Koray Kavukcoglu, and Hal Daumé. A binary classification framework for two-stage multiple kernel learning. In *International Conference on Machine Learning*, 2012.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020.

Yann LeCun and Ishan Misra. Self-supervised learning: The dark matter of intelligence. `https://ai.facebook.com/blog/self-supervised-learning-the-dark-matter-of-intelligence/`, 2021. Accessed: 2022-06-28.

Jason D. Lee, Qi Lei, Nikunj Saunshi, and Jiacheng Zhuo. Predicting what you already know helps: provable self-supervised learning. *Neural Information Processing Systems*, 2021.

Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Neural Information Processing Systems*, 2014.

Xin Li and Dan Roth. Learning question classifiers. In *International Conference on Computational Linguistics*, 2002.

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.

Lajanugen Logeswaran and Honglak Lee. An efficient framework for learning sentence representations. In *International Conference on Learning Representations*, 2018.

Zhuang Ma and Michael Collins. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. In *Conference on Empirical Methods in Natural Language Processing*, 2018.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Association of Computational Linguistics*, 2011.

Anuran Makur, Fabián Kozynski, Shao-Lun Huang, and Lizhong Zheng. An efficient algorithm for information decomposition and extraction. In *Conference on Communication, Control, and Computing (Allerton)*, 2015.

Andreas Maurer. A vector-contraction inequality for rademacher complexities. In *International Conference on Algorithmic Learning Theory*, 2016.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. The benefit of multitask representation learning. *J. Mach. Learn. Res.*, 2016.

Julian J. McAuley, Rahul Pandey, and Jure Leskovec. Inferring networks of substitutable and complementary products. *CoRR*, 2015.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. In *Neural Information Processing Systems*, 2017.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*, 2018.

Yu Meng, Chenyan Xiong, Payal Bajaj, Paul Bennett, Jiawei Han, Xia Song, et al. Coco-lm: Correcting and contrasting text sequences for language model pretraining. In *Neural Information Processing Systems*, 2021.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Charles A Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 2006.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Neural Information Processing Systems*, 2013b.

Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, 2016.

Jovana Mitrovic, Brian McWilliams, Jacob Walker, Lars Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms. In *International Conference on Learning Representations*, 2021.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Jiaqi Mu and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. In *International Conference on Learning Representations*, 2018.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, 2016.

Kento Nozawa and Issei Sato. Understanding negative samples in instance discriminative self-supervised representation learning. In *Neural Information Processing Systems*, 2021.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. Unsupervised learning of sentence embeddings using compositional n-gram features. North American Chapter of the ACL, 2018.

Christos H Papadimitriou, Prabhakar Raghavan, Hisao Tamaki, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. *Journal of Computer and System Sciences*, 2000.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *IEEE conference on computer vision and pattern recognition*, 2016.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*, 2014.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *Proceedings of NAACL-HLT*, 2018.

Raul Puri and Bryan Catanzaro. Zero-shot text classification with generative language models. *arXiv prepring arXiv:1912.10165*, 2019.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*, 2017.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

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. *arXiv preprint arXiv:1910.10683*, 2019.

Prajit Ramachandran, Peter Liu, and Quoc Le. Unsupervised pretraining for sequence to sequence learning. In *Conference on Empirical Methods in Natural Language Processing*, 2017.

Michael Reed. *Methods of modern mathematical physics: Functional analysis*. Elsevier, 2012.

Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. Can contrastive learning avoid shortcut solutions? In *Neural Information Processing Systems*, 2021.

Nikunj Saunshi, Sadhika Malladi, and Sanjeev Arora. A mathematical exploration of why language models help solve downstream tasks. In *International Conference on Learning Representations*, 2021.

Nikunj Saunshi, Jordan Ash, Surbhi Goel, Dipendra Misra, Cyril Zhang, Sanjeev Arora, Sham Kakade, and Akshay Krishnamurthy. Understanding contrastive learning requires incorporating inductive biases. In *International Conference on Machine Learning*, 2022.

Timo Schick and Hinrich Schütze. It's not just size that matters: Small language models are also few-shot learners. In *North American Chapter of the Association for Computational Linguistics*, 2021.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Neural Information Processing Systems 30*. 2017.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Empirical Methods in Natural Language Processing*, 2013.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. oLMpics–on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 2020.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European Conference on Computer Vision*, 2020a.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In *Neural Information Processing Systems*, 2020b.

Yuandong Tian, Lantao Yu, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning with dual deep networks. *arXiv preprint arXiv:2010.00578*, 2020c.

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. MLP-Mixer: An all-mlp architecture for vision. *Neural Information Processing Systems*, 2021.

Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive learning, multi-view redundancy, and linear models. 2021a.

Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive estimation reveals topic posterior information to linear models. *Journal of Machine Learning Research*, 2021b.

Yao-Hung Hubert Tsai, Yue Wu, Ruslan Salakhutdinov, and Louis-Philippe Morency. Demystifying self-supervised learning: An information-theoretical framework. *arXiv preprint arXiv:2006.05576*, 2020.

Yao-Hung Hubert Tsai, Yue Wu, Ruslan Salakhutdinov, and Louis-Philippe Morency. Self-supervised learning from a multi-view perspective. In *International Conference on Learning Representations*, 2021.

Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. In *International Conference on Learning Representations*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, 2017.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 2011.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning*, 2008.

Julius Von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. In *Neural Information Processing Systems*, 2021.

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. *arXiv preprint arXiv:1804.07461*, 2018.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, 2020.

Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *IEEE International Conference on Computer Vision*, 2015.

Yifei Wang, Qi Zhang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. Chaos is a ladder: A new understanding of contrastive learning. In *International Conference on Learning Representations*, 2022.

Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Zixin Wen and Yuanzhi Li. Toward understanding the feature learning process of self-supervised contrastive learning. In *International Conference on Machine Learning*, 2021.

Theresa Wilson and Janyce Wiebe. Annotating opinions in the world press. In *SIGdial Workshop of Discourse and Dialogue*, 2003.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Sen Wu, Hongyang Zhang, and Christopher Ré. Understanding and improving information transfer in multi-task learning. In *International Conference on Learning Representations*, 2020a.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *IEEE conference on computer vision and pattern recognition*, 2018.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. CLEAR: Contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466*, 2020b.

Wei Xu and Alex Rudnicky. Can artificial neural networks learn language models? In *International Conference on Spoken Language Processing*, 2000.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In *Association for Computational Linguistics*, 2021.

Han Yang, Xiao Yan, Xinyan Dai, Yongqiang Chen, and James Cheng. Self-enhanced gnn: Improving graph neural networks using model outputs. In *International Joint Conference on Neural Networks*, 2021.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. XLNet: Generalized autoregressive pretraining for language understanding. In *Neural Information Processing Systems*, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017a.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, 2016.

Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017b.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Neural Information Processing Systems*. 2015.

Zaiwei Zhang, Zhenxiao Liang, Lemeng Wu, Xiaowei Zhou, and Qixing Huang. Path-invariant map networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Roland S Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. In *International Conference on Machine Learning*, 2021.